DESENVOLVIMENTO DE MODELOS NEURAIS AUTÔNOMOS PARA PREVISÃO DE CARGA ELÉTRICA

Vitor Hugo Ferreira

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

_____
Prof. Alexandre Pinto Alves da Silva, Ph.D.

_____
Prof. Luiz Pereira Calôba, Dr.Ing.

_____
Prof. José Manoel de Seixas, D.Sc.

_____
Prof. Gerson Zaverucha, Ph.D.

_____
Prof. Reinaldo Castro Souza, Ph.D.

_____
Prof. Marcelo Cunha Medeiros, D.Sc.

RIO DE JANEIRO – BRASIL

MAIO DE 2008

FERREIRA, VITOR HUGO

Desenvolvimento de Modelos Neurais Autônomos para Previsão de Carga Elétrica, [Rio de Janeiro], 2008.

IX, 302 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia Elétrica, 2008)

Tese – Universidade Federal do Rio de Janeiro, COPPE.

1. Previsão de Carga
2. Redes Neurais Artificiais
3. Seleção de Variáveis de Entrada
4. Técnicas de Regularização
5. Treinamento *Bayesiano*
6. Modelos baseados em *kernel*

I. COPPE/UFRJ II. Título (série)

DEDICATÓRIA

Este trabalho é dedicado às pessoas mais importantes da minha vida, que acreditaram e depositaram extrema confiança no meu trabalho. Pessoas como o melhor pai do mundo, também conhecido como Seu Hugo; a melhor mãe do planeta, que também atende pelo nome de Dona Tina; meu avô materno, Seu Alcides, que lá em cima deve estar tomando umas e outras para comemorar mais uma fase ultrapassada na vida do seu neto; minha avó paterna, Dona Filhinha, que conseguiu ver o neto Engenheiro, porém não está presente para celebrar mais esta vitória; minha avó materna Mariana, que acredita muito nesse neto aqui e é o principal ponto de convergência e união da melhor família que um ser humano pode ter; por fim, porém não menos importante dedico à minha futura esposa, Paula, que aceitou a ingrata missão de me aturar pelo resto da vida. Apesar de não ter conhecido em vida, dedico este trabalho ao meu avô paterno, Hugo, sem o qual eu não teria a dádiva divina de ter o exemplo de homem e de pai que tenho ao meu lado.

Dedico também a todas as pessoas que me apoiaram ao longo deste caminho. Não posso me esquecer do grande Wilson Leão, uma das grandes referências da minha vida, que ajudou a forjar o homem que sou hoje. Aos meus amigos, que formam a família que Deus permitiu que eu escolhesse sempre me apoiando nos momentos de necessidade. Ao meu orientador e grande amigo, Alexandre, que sempre orientou, estimulou e apoiou minha vida acadêmica, desde a época da graduação em Itajubá.

Não dedico a Deus esta tese por que sei que este trabalho é ínfimo diante da infinidade da sua bondade. Porém, dedico a Ele todo meu esforço, trabalho e dedicação na busca por um mundo mais unido, solidário e justo, onde o amor, o respeito ao próximo e ao meio ambiente formem os pilares de uma nova civilização.

AGRADECIMENTOS

Primeiramente a Deus, por manter sempre meu caminho iluminado, concedendo sabedoria, confiança, saúde e paz a mim e a todos que estão a minha volta.

Aos meus pais, por terem me dado a vida e me ensinado a vivê-la. Por me aturarem por 27 anos e mesmo assim ainda me amarem. Por ser o porto seguro ao qual recorro nos raros momentos turbulentos. Por rirem comigo nos abundantes momentos de alegria da minha maravilhosa vida. Enfim, por constituírem a base do que sou hoje.

A toda a minha família, pela confiança depositada e pelo carinho enorme que a mantêm unida.

A todos os meus amigos, que sempre me apoiaram nos momentos difíceis, configurando realmente a família que Deus permitiu que escolhêssemos. Colegas de porta de boteco existem vários, mas são raros aqueles que surgem em hospitais na hora do aperto. Ou que ligam no exato momento em que descobrem uma notícia triste. Agradeço todas as noites pela família e pelos amigos que tenho!

À família LASPOT, agora mais distante, pela calorosa acolhida e pelo apoio incondicional durante os dois anos de Mestrado e três de Doutorado, em especial ao bom velhinho Hélio!

Por último, mas não menos importante (muito pelo contrário) agradeço ao meu orientador Alexandre, pelo suporte dado desde os tempos de graduação, estimulando e apoiando minha evolução dentro da área acadêmica. Se no início tinha um orientador, hoje posso afirmar com orgulho que tenho mais um grande amigo.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

DESENVOLVIMENTO DE MODELOS NEURAIS AUTÔNOMOS PARA PREVISÃO DE CARGA ELÉTRICA

Vitor Hugo Ferreira

Maio / 2008

Orientador: Alexandre P. Alves da Silva

Programa: Engenharia Elétrica

O conhecimento do comportamento futuro da carga apresenta importância vital na tomada de decisão em sistemas de potência. Nos últimos 20 anos, modelos neurais de previsão de carga vêm dominando a literatura. A capacidade de aproximação universal destes modelos pode levar ao ajuste excessivo dos dados, comprometendo os erros de previsão. Esta desvantagem, relacionada tanto com a seleção de entradas quanto com a complexidade do modelo, vem sendo enfrentada na literatura de forma heurística e desacoplada. Combinando teoria do caos, inferência *bayesiana* e minimização de limites superiores do erro de generalização, são desenvolvidos métodos autônomos (automáticos) de especificação de modelos neurais (MLP e modelos baseados em *kernel*), incluindo procedimentos analíticos e acoplados de seleção de entradas e controle de complexidade.

Abstract of the Thesis presented to COPPE/UFRJ as a partial fulfillment of the requiremenets for the degree of Doctor of Science (D.Sc.)

DEVELOPMENT OF AUTONOMOUS NEURAL NETWORK ELECTRIC LOAD FORECASTING MODELS

Vitor Hugo Ferreira

May / 2008

Advisors: Alexandre P. Alves da Silva

Department: Electrical Engineering

Anticipation of load's future behavior is very important for decision making in power system operation and planning. In the last 20 years, the literature on load forecasting has been dominated by neural network based proposals. The extent of nonlinearity provided by these models can lead to an excessive training data approximation, which usually increases the forecasting error. This drawback, which depends both on the input representation and the complexity of the model, has been tackled using heuristic procedures and in a decoupled way. Combining chaos theory, bayesian inference and minimization of an upper bound on generalization error, autonomous (automatic) neural models (MLP and kernel-based machines) are developed, including analytic and coupled procedures for input selection and complexity control.

Índice

Índice de Figuras

Índice de Tabelas

1 Introdução

A previsão de carga apresenta importância vital para a operação e o planejamento confiável, seguro e econômico de sistemas de potência. Em função disso, esta área de estudo vem adquirindo maior interesse por parte da comunidade científica ao longo dos anos, principalmente após o advento da competição nos mercados de energia. Neste novo cenário, os agentes integrantes destes mercados devem operar em regime de máxima eficiência, com a minimização dos custos operacionais e a correta avaliação do aporte de recursos financeiros para expansão dos sistemas contribuindo sobremaneira para o alcance desta condição.

De uma maneira geral, segundo os horizontes de interesse e a freqüência das observações, a previsão de carga pode ser classificada em três tipos, a saber: longo, médio e curto prazo. Entretanto, a definição de tais horizontes não é única. Previsões consideradas de médio prazo para algumas empresas podem ser consideradas de longo prazo para outras, dependendo do estudo em questão.

Segundo a literatura, previsões em longo prazo são aquelas realizadas para horizontes variando de alguns meses até trinta anos à frente, com os dados amostrados em base mensal ou anual. Estas previsões são utilizadas em diversas funções relacionadas com o planejamento técnico e financeiro das empresas de energia, tais como planejamento da expansão dos sistemas de transmissão e distribuição e do parque gerador, programação anual da manutenção de unidades geradoras, gerenciamento energético de longo prazo, desenvolvimento de estratégias operacionais, estudos de viabilidade econômica, planejamento dos investimentos e do orçamento, e pesquisa de mercado. Em mercados regulamentados, tais previsões podem ser utilizadas também para o desenvolvimento de políticas tarifárias.

Previsões de carga em médio prazo são aquelas realizadas para horizontes variando de uma semana até cinco anos, com as medições discretizadas em base diária, semanal ou mensal. As previsões em médio prazo fornecem subsídios para diversas atividades relacionadas ao planejamento da expansão e da operação de sistemas de potência, podendo ser citados: programação da compra de combustíveis; planejamento da manutenção de equipamentos, do intercâmbio entre áreas, das transações energéticas e do orçamento; otimização da programação das unidades geradoras; e desenvolvimento de estratégias de gerenciamento energético. Tais previsões também podem ser utilizadas para desenvolvimento de políticas tarifárias.

Para o horizonte de curto prazo, são consideradas previsões realizadas para intervalos variando de alguns minutos a até um mês à frente, utilizando dados em base de minutos, horas, ou dias. As previsões para este horizonte são de suma importância para a operação e o controle em tempo real de sistemas de potência. Dentre as funções inerentes ao planejamento da operação, a previsão de carga em curto prazo fornece subsídios para análise de segurança, incluindo estudo de contingências e elaboração de estratégias de gerenciamento da carga; programação da geração, abrangendo coordenação hidrotérmica, programação da compra e alocação de combustível, comissionamento de unidades térmicas e despacho econômico; estudos de fluxo de potência, como fluxo de potência ótimo e programação do intercâmbio entre áreas; programação da alocação de reserva girante; programação e avaliação das transações de compra e venda de energia; e programação da manutenção. Dentre as atividades relacionadas ao controle em tempo real de sistemas de potência, estas previsões fornecem informações importantes para controle automático da geração e controle do fluxo de potência reativa. Para o caso específico de empresas de distribuição de energia, o conhecimento do comportamento futuro da carga, particularmente do pico de carga,

nas diversas barras do sistema é um dos requisitos mais importantes para o aumento da eficiência da operação. Estas informações constituem a base para a estimação do estado do sistema e para cálculos técnicos e econômicos, possibilitando assim melhorias na manutenção dos equipamentos e no planejamento da operação dos sistemas de distribuição. Tais melhorias podem ser obtidas através de instalações de equipamentos de emergência, desligamento de circuitos, transferências de carga, aumento da refrigeração de equipamentos críticos e ajuste dos *tap's* dos transformadores das subestações.

Além da importância sob o ponto de vista técnico, a previsão em curto prazo também apresenta relevância sob o prisma econômico. Com o advento da competição oriunda da privatização dos mercados de energia em diversos países, os agentes de tais mercados foram obrigados a trabalhar em níveis elevados de eficiência. Visto que a previsão de carga em curto prazo está diretamente associada a diversas atividades relacionadas com a operação de sistemas de potência, a precisão de tais previsões está intimamente ligada à redução dos custos operacionais das empresas de energia. Segundo a estimativa apresentada em [1], para empresas de energia que apresentem gastos com combustível da ordem de centenas de milhões de dólares anuais, melhorias da ordem de 1 % na precisão das previsões em curto prazo podem resultar em reduções nos custos operacionais da ordem de centenas de milhares de dólares por ano. Outros estudos mostrando o impacto da precisão das previsões nos custos das empresas de energia podem ser encontrados em [2], [3] e [4].

Ainda dentro da ótica econômica, a previsão em curto prazo fornece informações essenciais tanto para a formação do preço da energia em mercados desregulamentados, embasando a avaliação do seu preço futuro, quanto em mercados regulamentados, subsidiando o desenvolvimento de políticas tarifárias. Portanto, tais

previsões são relevantes tanto para as empresas fornecedoras de energia quanto para os grandes consumidores industriais, já que estes últimos podem programar seu consumo em função do preço da energia em mercados competitivos, ou em função da tarifa estabelecida em mercados regulamentados.

Tendo em vista o impacto técnico e econômico da previsão de carga, vários modelos vêm sendo propostos para abordagem deste problema ao longo das últimas quatro décadas. Esta vasta experiência deu origem a várias metodologias, tais como regressão múltipla [5], [6], [7], [8], análise de séries temporais [9], [10], [11], [12], [13], redes neurais artificiais (RNAs) [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [33], sistemas de inferência *fuzzy* [35], [36], [37], e modelos híbridos [38], [39], [40], [41], [42], [43], [44], [45]. Entretanto, estas propostas requerem intervenção constante de especialistas na modelagem, no que tange tanto à seleção da estrutura quanto das variáveis de entrada dos modelos.

Neste contexto, o desenvolvimento de sistemas neurais de previsão de carga para níveis inferiores em sistemas de potência, como previsão por barramento, fica comprometido, visto que seria necessária a análise individual, por parte de especialistas, de cada barra do sistema em estudo. Na literatura existem propostas de modelos neurais para previsão por barramento [46], [47], [48]. Entretanto, estas metodologias definem um modelo fixo, incluindo conjunto de entradas e estrutura utilizada, para tratamento de todas as barras. Diante das características específicas de cada barra, que podem atender diferentes tipos de consumidores em diversos níveis de carregamento, esta abordagem não é a mais indicada, visto que dinâmicas não contempladas no processo de definição do modelo podem não ser modeladas. Em outras palavras, entradas significativas para modelagem de uma dada barra podem ser irrelevantes para outras, com o mesmo

valendo para estruturas, visto que dinâmicas complexas necessitam de modelos mais flexíveis, ao contrário de séries com comportamentos mais suaves.

No caso brasileiro, a previsão por barramento é uma necessidade, visto que o Operador Nacional do Sistema Elétrico (ONS) determina que os agentes de distribuição devam fornecer previsões de potência ativa e reativa, por barramento da Rede de Simulação, para um horizonte mínimo de sete meses, podendo chegar a até quatro anos, em base mensal [49], com incidência de multas e penalidades associadas à precisão das previsões fornecidas. O número elevado de barramentos para um dado agente, que pode variar desde dezenas até centenas de unidades, inviabiliza o estudo individualizado de cada barra para fins de previsão. Desta forma, é necessário o desenvolvimento de modelos autônomos para previsão de carga, que abdiquem da intervenção constante de especialistas ao longo do seu desenvolvimento, possibilitando assim a abordagem de diversas séries históricas simultaneamente. Tais modelos devem incluir metodologias automáticas para seleção de variáveis de entrada e controle de complexidade da estrutura estimada, evitando assim o ajuste excessivo dos dados de treinamento e dando origem a modelos com elevada capacidade de generalização, ou seja, desempenho satisfatório para dados ainda não disponibilizados.

Conforme mostra a vasta literatura neste assunto, o comportamento da carga é influenciado de maneira complexa, e muitas vezes não-linear, por uma série de fatores exógenos, como hora do dia, dia da semana, condições climáticas, dentre outras. Esta questão constitui um empecilho à aplicação de técnicas populares, como modelos de regressão linear e análise clássica de séries temporais por meio de modelos ARMA, do inglês *auto regressive moving average*, ao problema de previsão de carga. Além disso, estes métodos dependem de algumas premissas básicas nem sempre verificadas em

casos práticos, tais como tipo de ruído presente na saída, independência entre as variáveis explicativas, dentre outras.

Por outro lado, a literatura tem mostrado o sucesso da aplicação de modelos neurais a complexos problemas multivariados envolvendo bases de dados de cardinalidade considerável, como é o caso do problema de previsão de carga elétrica. Um dos softwares mais utilizados na América do Norte para previsão de carga, popularmente conhecido pela sigla ANNSTLF, *Artificial Neural Network Short Term Load Forecaster* [18], [20], é baseado em redes neurais. Na época da publicação da sua terceira versão [20], este modelo operava em 35 empresas dos EUA e do Canadá, sendo também utilizado pelo Operador Nacional do Sistema Elétrico (ONS) brasileiro. Um dos fatores que explicam este êxito consiste na elevada flexibilidade e capacidade de aproximação deste tipo de modelo, visto que, dado um número suficiente de neurônios, modelos neurais podem aproximar com precisão arbitrária qualquer função contínua [50]. Além disso, ao contrário dos modelos lineares clássicos, as redes neurais apresentam poucas premissas básicas a serem verificadas, aumentando assim a sua flexibilidade e robustez.

Apesar destas vantagens, desde as primeiras propostas de modelos neurais para previsão de carga [14], a utilização prática destas estruturas vem enfrentando alguns empecilhos, a saber: elevado esforço computacional para estimação do modelo, ausência de intervalos de confiança das previsões, baixa interpretabilidade dos resultados, escolha adequada do espaço de entrada e controle de complexidade da estrutura.

O aumento da capacidade de processamento disponível, juntamente com o desenvolvimento de algoritmos de treinamentos mais velozes [50], viabilizou a utilização de modelos neurais para aplicações práticas em sistemas de potência [51].

Diante do elevado grau de não-linearidade presente nestes modelos, o desenvolvimento de intervalos de confiança analíticos, de maneira análoga aos obtidos para modelos lineares, foi deixado de lado em um primeiro momento, com o foco voltado para metodologias baseadas em técnicas de re-amostragem [22]. Atualmente, a aplicação de técnicas de inferência *bayesiana* ao problema de treinamento de modelos neurais, proposta originalmente em [54], fornece uma estimativa analítica para os intervalos de confiança das previsões [50].

A extração de conhecimento de modelos neurais, possibilitando o fornecimento de interpretações qualitativas sobre as previsões, constitui uma das principais vantagens da junção destes modelos com sistemas difusos [38], [39]. Metodologias para interpretabilidade da saída de modelos neurais que abdicam de lógica difusa podem também ser encontradas em [55], [56]. Entretanto, estes trabalhos sinalizam para uma diminuição do grau de interpretabilidade em detrimento de ganhos em termos de precisão das previsões, sendo difícil a obtenção, para modelos neurais, do mesmo nível de interpretabilidade verificado em modelos lineares.

As duas questões restantes, relacionadas à escolha do espaço de entrada e ao controle de complexidade do modelo, são cruciais e ainda não receberam na literatura a devida atenção. Existem propostas tanto de técnicas de seleção de variáveis de entrada quanto de controle de complexidade de modelos neurais para previsão de carga. Entretanto, estas técnicas vêm sendo utilizadas de maneira desacoplada, comprometendo assim a capacidade de generalização do modelo obtido, visto que o nível de não-linearidade disponibilizado pela estrutura neural está diretamente relacionado com o espaço de representação das entradas. Conforme mencionado anteriormente, uma das principais vantagens destes modelos reside na sua capacidade de

aproximação universal. Porém, esta característica pode ser prejudicial se a questão do ajuste excessivo dos dados não for abordada de maneira adequada.

O principal objetivo das técnicas de seleção de entradas reside na escolha dos sinais que apresentem um nível mínimo de interdependência com a saída, retirando do modelo sinais desassociados com a última e que podem comprometer a precisão das previsões. Esta é uma das principais tarefas ao longo do desenvolvimento de modelos de previsão de carga. Diante da característica não-linear dos modelos neurais, técnicas de seleção de variáveis de entrada baseadas em índices lineares de autocorrelação e correlação cruzada não são recomendáveis [11]. Técnicas de extração de características através de análise de multi-resolução baseadas em *wavelets* são mais adequadas para este tipo de modelo, como mostrado em [30]. Entretanto, esta técnica observa somente a série em estudo, não considerando na análise o modelo que será utilizado. Técnicas que utilizam somente informações das séries em estudo para seleção de características são incluídas na categoria de procedimentos de filtragem [60]. Ao abdicarem do modelo de previsão, estas técnicas não garantem que o espaço de representação selecionado é o mais adequado em termos de desempenho de previsão. Sendo assim, uma metodologia mais orientada a modelos neurais, integrante do conjunto de técnicas encapsuladas [59], [60] é necessária, visando obter um espaço de representação adequado ao modelo em questão.

O controle de complexidade ou regularização de modelos neurais visa adequar o nível de não-linearidade disponibilizado pela estrutura à regularidade apresentada pelos dados, evitando a modelagem desnecessária do ruído e a conseqüente redução da capacidade de generalização do modelo. Um dos procedimentos mais populares de regularização de modelos neurais, e um dos mais utilizados em aplicações de previsão de carga, têm como base técnicas de validação cruzada, sendo conhecido como parada

antecipada do treinamento, do inglês *early stopping*. Esta técnica é deveras heurística, visto que deve ser selecionado adequadamente tanto o ponto a partir do qual o erro para o conjunto de validação é degradado, quanto o próprio conjunto de validação, que deve representar de maneira fidedigna a função a ser aproximada. Avaliações teóricas sobre as desvantagens deste tipo de técnica podem ser encontradas em [57], [58].

Nesta tese são desenvolvidos métodos automáticos, analíticos e acoplados de seleção de entradas e controle de complexidade de modelos neurais aplicados à previsão de carga. Da vasta gama de modelos neurais existentes na literatura, são considerados *perceptrons* de múltiplas camadas, do inglês *multi-layered perceptrons* (MLPs), e modelos baseados em *kernel* (*kernel-based machines*) [62]. Resultados preliminares indicando a aplicabilidade destas metodologias ao problema de previsão de carga podem ser encontrados em [32], [33] e [34].

Apesar de existirem trabalhos utilizando inferência *bayesiana* em previsão de carga [52], [53], nesta tese são percorridos de forma inédita todos os níveis hierárquicos da inferência *bayesiana* baseada na maximização da evidência [54] para MLPs, desde a estimação do conjunto de parâmetros até a seleção do modelo. A escolha de diferentes distribuições de probabilidade *a priori* para distintas variáveis de entrada dá origem a um procedimento de estimação da relevância de cada entrada conhecido como determinação automática de relevância (*automatic relevance determination* – ARD).

Para os modelos baseados em *kernel*, são desenvolvidas metodologias independentes para máquinas de vetor suporte (*support vector machines* – SVMs) [61] e para as máquinas de vetores relevantes (*relevance vector machines* – RVMs) [63]. A otimização de estimativas analíticas para o limite superior do erro estimado por validação cruzada única (*leave-one-out*) é utilizada tanto para seleção das variáveis de entrada, por meio da análise dos parâmetros do *kernel*, quanto para controle de

complexidade de SVMs, através da escolha dos parâmetros C e ε. Para as RVMs, o método automático é divisado através da aplicação de inferência *bayesiana* utilizando a maximização da evidência de forma análoga à proposta para MLPs, através da combinação original de um método analítico para seleção de funções de base [64] e subida em gradiente para estimação da relevância de cada entrada.

As metodologias desenvolvidas necessitam da definição de um conjunto inicial de entradas e de limiares de relevância a partir dos quais as entradas podem ser descartadas. Para tratar estas questões, técnicas baseadas na teoria do caos são aplicadas para definição do espaço de entrada inicial. Em conjunto com o teorema de *Takens* [65], aplicado para definição dos atrasos da própria série a serem utilizados como entradas, o conceito de sincronismo generalizado [66], [67] é utilizado para detecção de relacionamentos entre sistemas e definição dos respectivos atrasos. Para definição dos limiares de relevância, é desenvolvido um método original baseado na inserção artificial de variáveis aleatórias de prova [68], abdicando assim da intervenção de especialistas para a escolha dos sinais mais relevantes. A Figura 1.1 apresenta um diagrama ilustrando as diversas contribuições do trabalho e a inserção de cada uma delas ao longo do processo de especificação e treinamento de modelos neurais para previsão de carga. Nesta Figura, as siglas BMLP, A-L2-SVM e A-RVM fazem menção às três metodologias desenvolvidas nesta tese, a saber: inferência bayesiana aplicada à especificação de MLPs (BMLP), método automático para especificação de SVMs (A-L2-SVM) e método automático de especificação de RVMs (A-RVM).

Para avaliação das técnicas, são utilizadas três bases de dados públicas. A primeira corresponde a dados horários de carga e temperatura de uma concessionária de energia norte-americana [8], [14], os quais vêm sendo utilizados em competições entre modelos de previsão de carga. A segunda base de dados apresenta informações de carga

e temperatura máximas diárias da *Eastern Slovakian Electricity Corporation*, as quais foram utilizadas na competição promovida em 2001 pelo *European Network on Intelligent Technologies for Smart Adaptive Systems* (EUNITE) [29]. O último conjunto possui dados de carga, temperatura e preço da energia, verificados a cada meia-hora e disponibilizados pela *National Electricity Market Management Company Limited* (NEMMCO), empresa responsável pela operação do sistema elétrico e gerenciamento do mercado de energia australiano [31], [69], [70]. A utilização destas bases de dados tem por objetivo a comparação entre as metodologias propostas e as técnicas correntemente utilizadas na literatura.

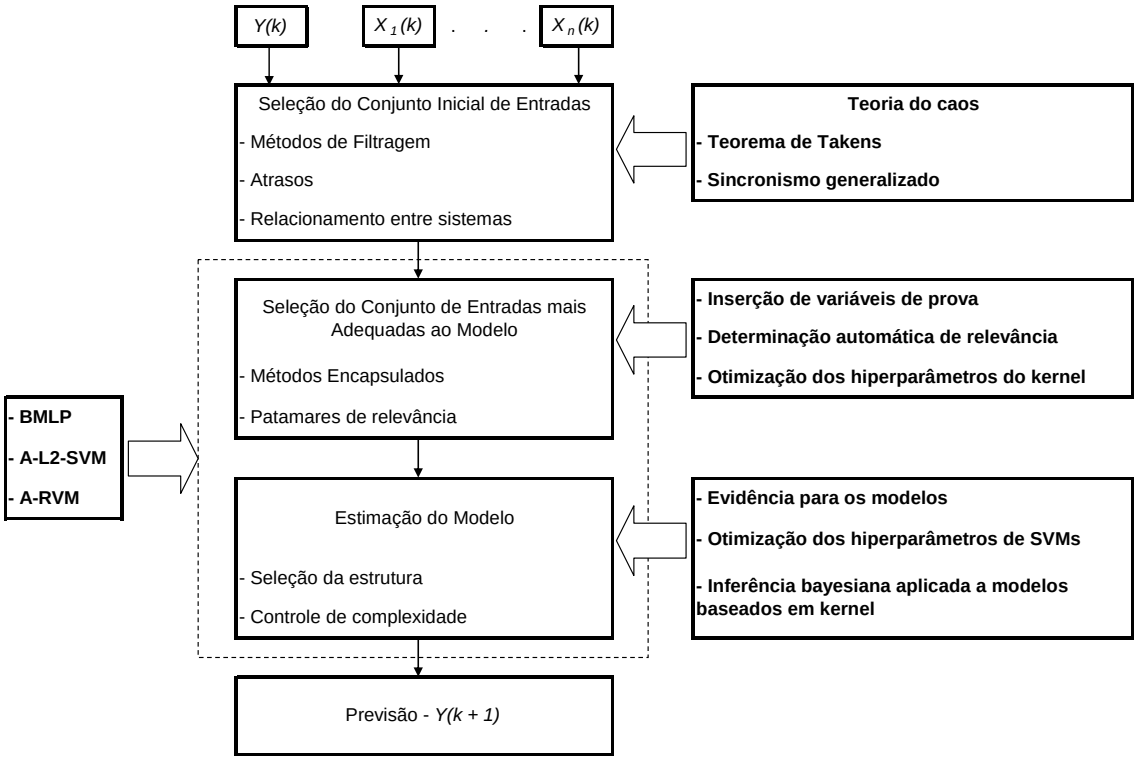

Figura 1.1 – Fluxograma do processo de desenvolvimento de modelos neurais e a inserção das técnicas desenvolvidas nesta tese

Esta tese está organizada da seguinte maneira. O capítulo 2 discute de forma geral os problemas de seleção de entrada e controle de complexidade, ou regularização, de modelos neurais, destacando o estado da arte no que tange a estes assuntos em

previsão de carga. As técnicas de seleção do conjunto inicial baseadas na teoria do caos, juntamente com o método de definição empírica de limiares de relevância, são apresentadas no capítulo 3. Os modelos automáticos desenvolvidos nesta tese são apresentados nos capítulos 4 e 5, que respectivamente discutem os métodos automáticos desenvolvidos para os MLPs e para as máquinas baseadas em *kernel*. O sexto capítulo é dedicado à apresentação dos resultados, incluindo uma descrição das bases de dados envolvidas e dos modelos utilizados. Por fim, são apresentadas as conclusões e sugestões de trabalhos futuros.

2 Redes neurais artificiais

As redes neurais artificiais (RNAs) podem ser vistas como um processador de sinais paralelamente distribuído, constituído de unidades de processamento simples, conhecidas como neurônios, que adquirem conhecimento acerca de uma determinada tarefa através da integração com o ambiente via um algoritmo de aprendizagem. Tal conhecimento é armazenado nos pesos sinápticos que interligam os diversos neurônios. A Figura 2.1 apresenta um diagrama esquemático de um neurônio, cuja saída c é dada pela equação:

$$c = \phi\left(\sum_{i=1}^{n} \omega_i x_i + b\right) \tag{2.1}$$

Nesta equação, c representa a saída do neurônio, $\underline{\omega} \in \mathbb{R}^n$, $\underline{\omega} = \begin{bmatrix} \omega_1 & \dots & \omega_n \end{bmatrix}^t$, o conjunto de pesos sinápticos que ligam as entradas $\underline{x} \in \mathbb{R}^n$, $\underline{x} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}^t$, deste neurônio, que podem ser oriundas da saída de outro neurônio ou da própria camada de entrada, $b \in \mathbb{R}$ o *bias* associado e $\phi(\cdot) : \mathbb{R} \to \mathbb{R}$ a sua função de ativação.

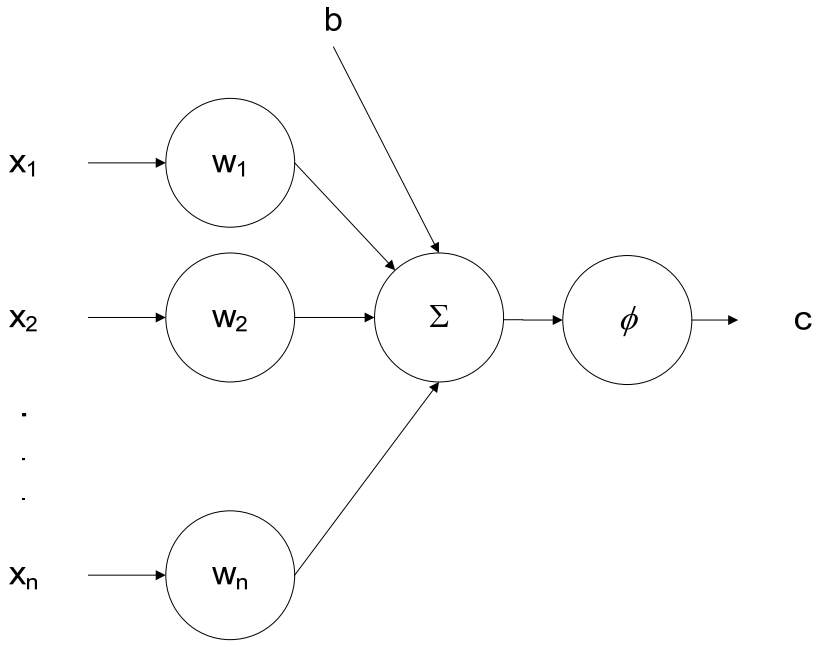

Figura 2.1 – Diagrama esquemático de um neurônio

Mantendo a analogia biológica, os neurônios constituintes das RNAs são dispostos em camadas, e a forma com que estas camadas são interligadas define a arquitetura do modelo. Basicamente, existem duas estruturas, as redes alimentadas adiante, *feedforward*, e as redes recorrentes. Nas redes *feedforward* as camadas são conectadas de forma consecutiva e adjacente, com o sinal fluindo da entrada para a saída em sentido único, conforme ilustrado na Figura 2.2. As redes recorrentes apresentam um ou mais laços de realimentação na estrutura apresentada nesta Figura. Visto que a maioria das propostas de modelos neurais para previsão de carga utiliza modelos *feedforward*, este trabalho focará apenas neste tipo de estrutura.

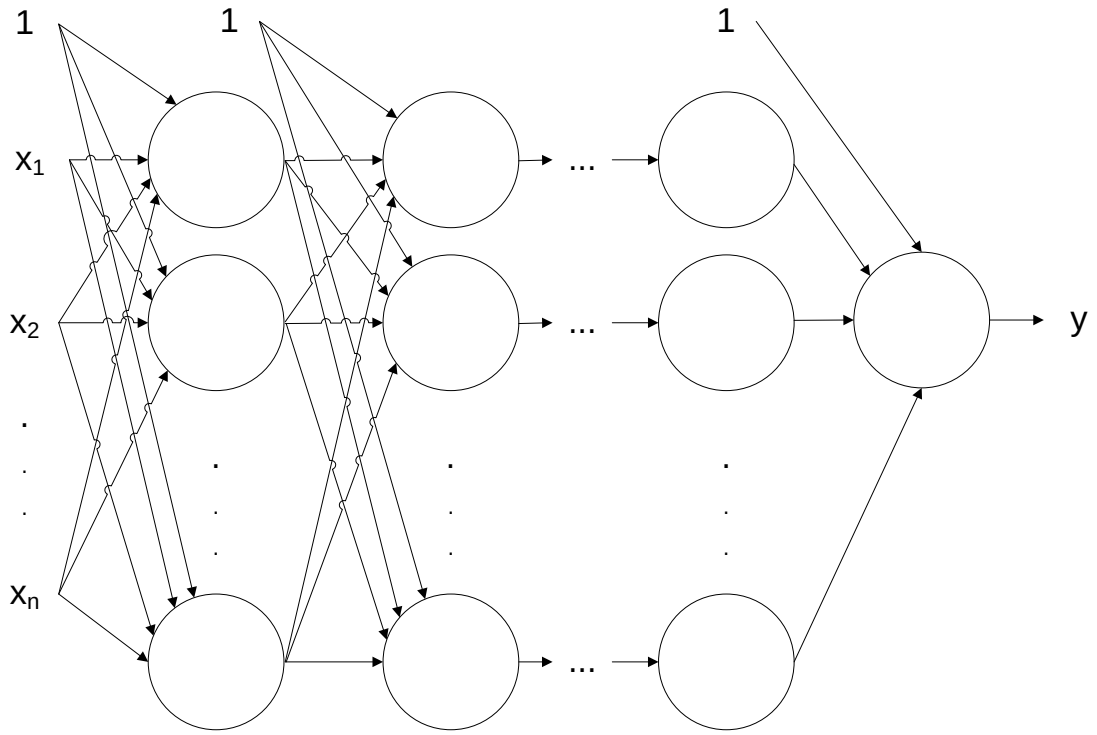

Figura 2.2 – Rede neural *feedforward* com múltiplas camadas e saída única

Dentre as principais vantagens dos modelos neurais, podem ser citadas as seguintes:

- Não-linearidade: para o caso mais comumente utilizado, em que a função de ativação dos neurônios da camada oculta é não-linear, o modelo neural

resultante da interconexão destas unidades mais simples apresenta considerável grau de não-linearidade. Entretanto, esta vantajosa característica pode ser prejudicial na presença de dados ruidosos, problema que será abordado ao longo deste capítulo.

- Mapeamento entrada-saída: a partir de um conjunto de pares entrada-saída, as redes neurais realizam um mapeamento destes dados, sem a necessidade de desenvolvimento de modelos matemáticos abordando a dinâmica do processo.
- Adaptabilidade: estes modelos apresentam elevada capacidade de adaptação em virtude de mudanças nas condições do ambiente para o qual a rede foi treinada para operar. Para tal, basta treinar a rede novamente, incluindo no conjunto de treinamento os padrões referentes às novas condições operacionais.
- Implementação simples: já que os modelos *feedforward* podem ser vistos como grafos orientados, a implementação destas estruturas é extremamente simples quando comparada com o grau de complexidade dos modelos que podem ser gerados.

Além destas vantagens, o teorema da aproximação universal [50] afirma que modelos *feedforward* podem aproximar com precisão arbitrária qualquer função contínua $F(\underline{x}): \mathbb{R}^n \to \mathbb{R}$. Para tal, a estrutura deve apresentar ao menos uma camada oculta contendo neurônios com função de ativação contínua, não-constante, limitada, e uma saída linear, representando a aproximação de $F(\underline{x})$ gerada pelo modelo. Portanto, modelos *feedforward* com uma única camada escondida contendo um número suficiente de neurônios com função de ativação com as características anteriormente citadas podem aproximar qualquer função contínua. Esta característica constitui a principal motivação para utilização ao longo deste trabalho de modelos com uma única camada

escondida. A saída y deste tipo de modelo, contendo m neurônios na camada oculta, é dada por:

$$y = \sum_{j=1}^{m} \omega_j \phi\left(\sum_{i=1}^{n} \omega_{ji} x_i + b_j \right) + b \tag{2.2}$$

Na expressão acima, $\underline{\omega}_s \in \mathbb{R}^m$, $\underline{\omega}_s = \begin{bmatrix} \omega_1 & \omega_1 & \dots & \omega_m \end{bmatrix}^t$, representa os pesos que ligam os neurônios da camada oculta ao neurônio linear de saída, $\underline{\omega}_j \in \mathbb{R}^n$, $\underline{\omega}_j = \begin{bmatrix} \omega_{j1} & \omega_{j2} & \dots & \omega_{jn} \end{bmatrix}^t$, constituído pelos pesos que ligam as entradas ao j-ésimo neurônio da camada escondida, $b_j \in \mathbb{R}$ o bias deste neurônio e $b \in \mathbb{R}$ o bias do neurônio de saída. Assim, o vetor $\underline{w} \in \mathbb{R}^M$, $\underline{w} = \begin{bmatrix} \underline{\omega}_s^t & \underline{\omega}_1^t & \dots & \underline{\omega}_j^t & b & b_1 & \dots & b_j \end{bmatrix}^t$ apresenta um total de $M = mn + 2m + 1$ parâmetros livres.

Várias propostas de modelos de previsão de carga baseados em redes *feedforward* com uma única camada escondida podem ser encontradas na literatura, podendo ser citados o MLP [14], [15], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [33], as redes de função de base radial, do inglês *radial basis function networks* (RBFN) [16], [71], [72], [73], *functional link network* (FLN), [74], [75], SVM [76], [77], [78], [79], [80], [81], dentre outros. Conforme mencionado anteriormente, neste trabalho são utilizados apenas modelos *feedforward* com uma única camada oculta, mais especificamente, o MLP e as máquinas baseadas em *kernel*.

Apesar da vantajosa característica de aproximação universal, o objetivo do desenvolvimento de uma máquina de aprendizagem não reside na representação exata do conjunto de dados disponíveis, mas sim na obtenção de um modelo estatístico do processo gerador de tais dados [50]. Logo, é desejado que o modelo apresente resultados satisfatórios tanto para os dados disponíveis quanto para novos dados a serem

apresentados. Em outras palavras, a estrutura desenvolvida deve apresentar boa capacidade de generalização.

Na presença de dados ruidosos, o elevado grau de não-linearidade disponibilizado pelas RNAs pode modelar além da função geradora dos dados traços específicos do conjunto de dados disponível, comprometendo o desempenho do modelo. Evitar o ajuste excessivo dos dados de treinamento, popularmente conhecido como *overfitting*, constitui uma das principais tarefas no desenvolvimento de modelos neurais.

O nível de não-linearidade disponibilizado pelo modelo neural está diretamente relacionado com as suas entradas. Além disso, a utilização direta de variáveis irrelevantes que não possuem nenhum grau de interdependência com a saída pode comprometer a precisão das previsões. Assim, em conjunto com o controle de complexidade, é necessária a seleção adequada do espaço de entrada no intuito de obter estruturas com elevada capacidade de generalização.

Estas questões evidenciam a necessidade do controle de complexidade, ou regularização, de modelos neurais em conjunto com a seleção do espaço de entrada. Apesar da importância, a grande maioria das propostas de RNAs para previsão de carga aborda de maneira inadequada esta questão, de um lado tratando somente da regularização, e de outro abordando somente a seleção de variáveis de entrada. Desta forma, em conjunto com a definição dos problemas de seleção de entradas e controle de complexidade, serão discutidas as propostas encontradas na literatura em previsão de carga que abordam estas questões. As raras propostas de modelos autônomos encontradas serão apresentadas na seqüência, antes da discussão que encerra o capítulo.

2.1 Seleção do espaço de entrada

Além de estar relacionada com a complexidade do modelo, a seleção do espaço de entrada possui importância sob diversos aspectos, tais como facilitação da

visualização e entendimento dos dados, redução do número de sinais a serem medidos e conseqüentemente armazenados, diminuição do esforço computacional necessário para treinamento e melhoria do desempenho de previsão [60]. Apesar da importância, este estágio do desenvolvimento de modelos neurais em previsão de carga ainda não mereceu a devida atenção na literatura. De outra forma, a utilização de procedimentos analíticos adequados a modelos não-lineares para escolha das variáveis de entrada ainda não é praxe no desenvolvimento de previsores neurais de carga. A grande maioria das aplicações define de forma heurística o espaço de entrada, utilizando conhecimento de operadores do sistema e de especialistas [25], [26], [28], [42], [44], [46], [71], [72], [73], [82], [83], [84], [85], [86], [87], [88], [89], [90], [91], [92].

Existem na literatura duas abordagens gerais para este problema. A primeira metodologia, conhecida como filtragem, utiliza informações somente das séries em estudo para escolha das entradas, buscando determinar aquelas mais relacionadas com a saída. Neste contexto, são descartadas variáveis ditas irrelevantes, apresentando pequena ou nenhuma relação com a saída, e redundantes, possuindo forte interdependência com algum outro sinal de entrada mais relacionado com a saída desejada.

Os métodos encapsulados de seleção de entradas buscam melhorias no desempenho do modelo ao longo do processo de seleção do espaço de entrada. Em outras palavras, visam determinar o conjunto de entradas mais adequado para o modelo em questão. Ao fazerem uso direto do modelo, estes métodos são mais recomendáveis sob o ponto de vista de previsão. Maiores detalhes sobre estes dois grupos de métodos de seleção de entrada serão apresentados a seguir.

2.1.1 Métodos de filtragem

Os métodos de filtragem fazem uso de estatísticas para avaliação de relacionamentos entre variáveis. Dentre os diversos indicadores encontrados na literatura, o índice de correlação linear é o mais simples e popular para mensuração de relacionamentos entre variáveis. Dadas duas variáveis aleatórias X e Y, o coeficiente de correlação τ_{XY} é definido pela expressão:

$$\tau_{XY} = \frac{E\left\{\left[X - E(X)\right]\left[Y - E(Y)\right]\right\}}{\sqrt{E\left\{\left[X - E(X)\right]^2\right\} E\left\{\left[Y - E(Y)\right]^2\right\}}} \tag{2.3}$$

Na equação acima, $E(\cdot)$ representa o operador de valor esperado, para variáveis contínuas dado por:

$$E(X) = \int_{-\infty}^{\infty} x f(x)\, dx \tag{2.4}$$

Em (2.4), $f(x)$ representa a função de densidade de probabilidade de X. O valor esperado $E(X)$ de uma variável discreta apresentando p possibilidades de ocorrência e com distribuição de probabilidade $P(X = x_i) : \aleph \to \mathbb{R}$, $x \in \aleph = \{x_1, x_2, ..., x_p\}$, é definido como:

$$E(X) = \sum_{i=1}^{p} x_i P(X = x_i) \tag{2.5}$$

O coeficiente τ_{XY}, definido no intervalo $[-1,1]$, mensura relacionamentos entre variáveis aleatórias. Informalmente, correlações positivas indicam que aumentos em X e Y ocorrem simultaneamente. Por outro lado, variações positivas em X ocorrendo em conjunto com decréscimos em Y denotam a existência de correlação negativa. Por último, se X e Y forem estatisticamente independentes, a correlação τ_{XY} entre estas variáveis é nula. Entretanto, a recíproca não é verdadeira, ou seja, correlação nula não

implica independência estatística entre variáveis. Este fato encontra fundamento na capacidade de τ_{XY} avaliar somente relacionamentos lineares. Por exemplo, a correlação entre as variáveis aleatórias X e $Y = X^2$ é nula, apesar destas variáveis por construção serem dependentes entre si.

A definição de τ_{XY} pela equação (2.3) não é utilizada na prática, sendo substituída pelo índice de correlação amostral r_{XY}. De posse de um conjunto $D = \left\{ \left(x_i, y_i \right) \in \mathbb{R}^2 : X = x_i,\ Y = y_i,\ i = 1, 2, ..., N \right\}$ contendo N realizações de X e Y, r_{XY} é obtido através da seguinte expressão:

$$r_{XY} = \frac{\sum_{i=1}^{N} \left(x_i - \bar{x} \right) \left(y_i - \bar{y} \right)}{\sqrt{\sum_{i=1}^{N} \left(x_i - \bar{x} \right)^2 \sum_{j=1}^{N} \left(y_j - \bar{y} \right)^2}} \tag{2.6}$$

Em (2.6), $\bar{x}$ e $\bar{y}$ respondem pelas médias amostrais das variáveis aleatórias X e Y, respectivamente estimadas por:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{2.7}$$
$$\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$$

Além de relações entre diferentes tipos de variáveis, a expressão (2.6) pode ser utilizada para avaliação de possíveis interdependências entre valores consecutivos de um processo estocástico. Considerando $X = \begin{bmatrix} x_1 & x_2 & ... & x_N \end{bmatrix}^t$ como uma seqüência de número reais aleatórios, o coeficiente de autocorrelação amostral $r_{XX}\left(k \right)$ do processo estocástico X pode ser escrito da seguinte forma:

$$r_{XX}(k) = \frac{\sum_{i=1}^{N}\left(x_i - \bar{x}\right)\left(x_{i+k} - \bar{x}\right)}{\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2} \tag{2.8}$$

O índice $r_{XX}(k)$ mede o nível de interdependência entre a realização x_t do processo estocástico X no instante t e a observação x_{t+k} deste mesmo processo realizada k instantes à frente. Este conceito pode ser aplicado para análise de pares de processos estocásticos, ampliando a informação fornecida por r_{XY} em virtude da inserção do aspecto temporal. Desta forma, o índice de correlação cruzada amostral $r_{XY}(k)$ passa a ser dado por:

$$r_{XY}(k) = \frac{\sum_{i=1}^{N}\left(x_i - \bar{x}\right)\left(y_{i+k} - \bar{y}\right)}{\sqrt{\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2 \sum_{j=1}^{N}\left(y_j - \bar{y}\right)^2}} \tag{2.9}$$

Em (2.9), o nível de interdependência entre os processos X e Y é quantificado para realizações verificadas em diferentes instantes de tempo, ou seja, x_t e y_{t+k}, respectivamente. Além disso, vale destacar que $r_{XY}(0)$ representa o índice de correlação cruzada r_{XY} dado por (2.6).

Os índices de correlação dados nas equações (2.8) e (2.9) são comumente utilizados para identificação de sistemas lineares. Métodos clássicos de análise de séries temporais avaliam o comportamento destes índices para diferentes atrasos de tempo visando à determinação da ordem de modelos ARMAX, do inglês *auto regressive moving average with exogenous input*. Especificamente, o estudo da função de autocorrelação da série Y a ser modelada (por exemplo um histórico de carga horária) obtida através do cálculo da equação (2.8) para diferentes atrasos k, permite estimar a ordem da parcela média móvel. A ordem da parcela associada à entrada exógena X,

relacionada por exemplo com medidas horárias de temperatura, é obtida através da análise da função de correlação cruzada, gerada a partir da avaliação da expressão (2.9) para diversos valores de k. Para estimação da ordem da parcela autoregressiva, é necessário o estudo da função de autocorrelação parcial. Relacionada com os índices de autocorrelação, esta função é obtida através da solução do sistema de equações de *Yule-Walker* [93], dado por:

$$\begin{bmatrix} 1 & r_{XX}(1) & \cdots & r_{XX}(k-1) \\ r_{XX}(1) & 1 & \cdots & r_{XX}(k-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{XX}(k-1) & r_{XX}(k-2) & \cdots & 1 \end{bmatrix} \begin{bmatrix} \alpha_{XX}(1) \\ \alpha_{XX}(2) \\ \vdots \\ \alpha_{XX}(k) \end{bmatrix} = \begin{bmatrix} r_{XX}(1) \\ r_{XX}(2) \\ \vdots \\ r_{XX}(k) \end{bmatrix} \tag{2.10}$$

O índice de autocorrelação parcial $\alpha_{XX}(k)$ busca avaliar isoladamente a relação entre x_t e x_{t+k}. Enquanto $r_{XX}(k)$ mensura a relação existente entre x_t e x_{t+k} considerando a dependência de x_t em relação à x_{t+1}, x_{t+2} até x_{t+k-1}, $\alpha_{XX}(k)$ quantifica somente o relacionamento entre x_t e x_{t+k}, não incorporando as interdependências entre instantes de tempo anteriores. A análise de índices lineares de autocorrelação parcial e correlação cruzada é uma técnica comumente utilizada para seleção de variáveis de entrada em previsão de carga [94], [95], [96], [97], [98], [99], [100].

Além do procedimento clássico para identificação de modelos lineares de séries temporais, apresentado com maiores detalhes em [93] e [101], os índices de correlação podem ser utilizados de forma alternativa para seleção de entradas. Dado um conjunto $D = \left\{ (\underline{x}_k, y_k) \in \mathbb{R}^n \times \mathbb{R},\ k = 1, 2, ..., N \right\}$ contendo N pares entrada-saída, o índice r_{XY} dado por (2.6) pode ser utilizado para determinação das variáveis mais relacionadas com a saída, juntamente com a detecção de possíveis redundâncias entre os sinais de entrada. Especificamente, é calculado o índice de correlação cruzada r_{XY} entre cada uma das n entradas e a saída, sendo selecionados somente os sinais que apresentarem

nível mínimo de interdependência com a última. Feita a primeira filtragem, a existência de redundância entre os sinais selecionados é determinada através da estimativa de r_{XY} entre todas as entradas escolhidas. Existindo variáveis redundantes, é mantida no modelo final a entrada mais relacionada com a saída, utilizando como medida de avaliação o índice de correlação cruzada calculado no primeiro estágio.

O método descrito anteriormente necessita da definição de limiares a partir dos quais os sinais podem ser considerados descorrelacionados. Segundo [93], a correlação cruzada r_{XY} entre duas séries descorrelacionadas apresenta assintoticamente distribuição *gaussiana* com valor esperado e variância dados por:

$$E\left[r_{XY}\right] = 0$$
$$E\left[r_{XY}^2\right] = \frac{1}{N} \tag{2.11}$$

Estas estatísticas podem ser utilizadas em um teste de hipótese para r_{XY}, ou seja, para identificação da existência de correlação entre X e Y. Assim, para r_{XY} ser diferente de zero com nível de confiança α, esta estatística deve pertencer a um dos seguintes intervalos:

$$r_{XY} < -\frac{s}{\sqrt{N}} \text{ ou } r_{XY} > \frac{s}{\sqrt{N}} \tag{2.12}$$

Na equação (2.12), s representa uma constante relacionada com o grau de confiança do teste. Para $s = 2$, o nível de confiança α é da ordem de 95 %. Utilizando este teste, é possível determinar a existência de correlação entre variáveis, fornecendo o limiar necessário para o método de seleção de entradas descrito anteriormente.

TSEKOURAS *et. al.* [99] utilizam o método linear descrito acima, com os limiares de relevância definidos pelo usuário, em conjunto com uma técnica encapsulada para avaliação do modelo. Além dos problemas relacionados com a utilização de índices lineares para seleção de entradas, o elevado esforço computacional

requerido pelo estágio encapsulado, discutido na próxima seção, compromete a aplicabilidade da proposta.

Outra técnica linear de seleção de entradas comumente utilizada é a análise de componentes principais, do inglês *principal component analysis* (PCA). Diferentemente da análise de correlação, esta metodologia busca determinar um novo espaço de entrada, de dimensão reduzida em relação ao espaço original, através de uma transformação linear do primeiro, minimizando a perda média de capacidade de reprodução do espaço original em virtude da redução de dimensionalidade. Visando minimizar a perda da capacidade de representação das entradas originais não levando em conta as saídas desejadas, componentes vitais para discriminação entre classes ou para previsão podem ser desconsideradas, se estas não contribuírem para caracterização do vetor de entradas original [50]. Exemplos da aplicação desta técnica em previsão de carga podem ser encontrados em [102], [103].

Métodos lineares como análise de índices de correlação e PCA, apesar de populares, de simples implementação e entendimento, quantificam apenas relacionamentos lineares entre as variáveis, não identificando possíveis interdependências de outra natureza. Diante da característica não-linear do modelo neural, estas metodologias não são as mais adequadas para seleção de variáveis de entrada deste tipo de estrutura, visto que sinais relacionados de forma não-linear com a saída podem eventualmente ser descartados do modelo final.

A teoria da informação fornece índices capazes de quantificar relacionamentos de quaisquer naturezas entre variáveis. Seja X uma variável aleatória discreta apresentando p possibilidades de ocorrência e com distribuição de probabilidade $P\left(X = x_i\right) : \aleph \to \mathbb{R}$, $x \in \aleph = \left\{x_1, x_2, ..., x_p\right\}$. A entropia de *Shannon* para a variável aleatória X é definida como [104]:

$$H(X) = -E\left[\log P(X)\right] = -\sum_{i=1}^{p} P(X = x_i) \log P(X = x_i) \tag{2.13}$$

De maneira análoga, a entropia conjunta de duas variáveis aleatórias discretas X e Y, $x \in \aleph = \{x_1, x_2, x_3, ..., x_p\}$, $y \in \Im = \{y_1, y_2, y_3, ..., y_q\}$, com probabilidade conjunta $P(X = x_i, Y = y_j) : \aleph \times \Im \to \mathbb{R}$ pode se definida da forma que segue:

$$H(X,Y) = -E\left[\log P(X,Y)\right] = -\sum_{i=1}^{p}\sum_{j=1}^{q} P(X = x_i, Y = y_j) \log P(X = x_i, Y = y_j) \tag{2.14}$$

Os índices de entropia definidos acima representam medidas do nível de incerteza existente em uma dada variável aleatória discreta ou em pares de variáveis. De outra forma, o grau de informação que uma dada variável carrega sobre outra, ou seja, a redução da incerteza em Y em virtude do conhecimento de X, é mensurado no índice de informação mútua, dado por:

$$I(X,Y) = \sum_{i=1}^{p}\sum_{j=1}^{q} P(X = x_i, Y = y_j) \log \frac{P(X = x_i, Y = y_j)}{P(X = x_i) P(Y = y_j)} \tag{2.15}$$

A informação mútua $I(X,Y)$ mede a divergência entre a probabilidade conjunta de X e Y, $P(X,Y)$, e o produto das probabilidades marginais. Desta forma, $I(X,Y)$ pode ser entendido como uma medida de distância entre a existência de relacionamento entre X e Y, representado por $P(X,Y)$, e a independência entre as variáveis, visto que nesse caso $P(X,Y) = P(X)P(Y)$ com $I(X,Y)$ igual a zero.

A relação entre $I(X,Y)$ e as respectivas entropias é obtida de forma direta, manipulando as equações (2.13) a (2.15) e dando origem à seguinte expressão:

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \tag{2.16}$$

Os conceitos de entropia e informação mútua apresentados acima podem ser estendidos para variáveis contínuas. Seja $f(x):\mathbb{R}\to\mathbb{R}$ a função de densidade de probabilidade da variável aleatória contínua $X\in\mathbb{R}$, $g(y):\mathbb{R}\to\mathbb{R}$ a densidade de probabilidade de $Y\in\mathbb{R}$ e $q(x,y):\mathbb{R}^2\to\mathbb{R}$ a densidade de probabilidade conjunta de X e Y. Assim, a entropia diferencial de X e a entropia diferencial conjunta são definidas como:

$$h(X)=-E\left[\log f(x)\right]=-\int_{-\infty}^{\infty} f(x)\log f(x)\,dx \tag{2.17}$$

$$h(X,Y)=-E\left[\log q(x,y)\right]=-\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} q(x,y)\log q(x,y)\,dxdy \tag{2.18}$$

Neste contexto, a informação mútua entre X e Y é dada por:

$$I(X,Y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} q(x,y)\log\frac{q(x,y)}{f(x)g(y)}\,dxdy \tag{2.19}$$

A relação entre informação mútua e entropia para variáveis aleatórias discretas, dada em (2.16), permanece válida para sinais contínuos, com a entropia sendo substituída pelo seu equivalente diferencial. Contudo, vale destacar que, enquanto as medidas de entropia para variáveis discretas são invariantes a transformações aplicadas aos sinais, o mesmo não é verificado para os análogos contínuos.

Conforme mencionado anteriormente, o índice de informação mútua está relacionado com a diminuição da incerteza sobre uma dada variável aleatória em função da verificação de outro sinal aleatório. Pode ser mostrado [104] que este índice é necessariamente não-negativo, assumindo seu valor mínimo igual a zero somente para o caso onde as variáveis aleatórias envolvidas são independentes. Porém, ao contrário do índice de correlação cruzada r_{XY}, $I(X,Y)$ não é limitado superiormente, sendo dependente da forma na qual as variáveis são distribuídas.

O cálculo da informação mútua $I(X,Y)$ utilizando as expressões (2.15), (2.16) ou (2.19) exige a estimação das distribuições de probabilidade marginais de X e Y, além da distribuição conjunta, a partir de um conjunto de dados D. Para variáveis discretas, estas funções podem ser estimadas diretamente por meio de histogramas. No caso de variáveis contínuas, a estimação das respectivas funções de densidade de probabilidade não é trivial. Técnicas para estimação de tais probabilidades, como histogramas e métodos baseados em *kernel*, são apresentados no APÊNDICE A.

Diante da capacidade de capturar relacionamentos de quaisquer naturezas, a informação mútua $I(X,Y)$ pode ser utilizada de forma análoga ao índice de correlação r_{XY} para seleção de variáveis, selecionando as mais relacionadas com a saída e descartando eventuais sinais redundantes. Porém, para $I(X,Y)$ não existem limiares de relevância analíticos similares aos obtidos para r_{XY}, exigindo a definição de limites para este índice. De outra forma, o número de variáveis a serem selecionadas também pode ser especificado. Tais definições são dependentes do problema em questão, requisitando a intervenção de especialistas na modelagem. Outro empecilho na análise de índices de informação mútua em previsão de carga reside na dificuldade da estimação das probabilidades ou funções de densidade de probabilidade necessárias no cálculo de $I(X,Y)$ para variáveis contínuas. Diante destas questões, apesar de atrativo sob o ponto de vista teórico, este método ainda encontra poucas aplicações em previsão de carga [11].

A teoria do caos também fornece ferramentas para detecção de interdependência de qualquer natureza entre variáveis. Segundo esta teoria, séries temporais caóticas podem ser expressas por modelos determinísticos aplicados a espaços de elevada dimensionalidade, conhecidos como espaços de fase. Conforme será apresentado na

seção 3.1, no contexto do teorema de *Takens* [65], este espaço pode ser definido por valores atrasados da própria série.

Em previsão de carga, esta teoria é aplicada em conjunto com modelos neurais, que desempenham a função do modelo determinístico responsável pela reconstrução da série no espaço de fase [19]. Apesar da capacidade de capturar dependências não-lineares, no contexto do teorema de *Takens* a teoria do caos permite estudo somente de séries univariadas, não contribuindo para identificação de relacionamentos entre séries distintas, visto que o espaço de fase é constituído somente de valores atrasados da série em estudo. Para problemas multivariados, ou seja, tarefas de modelagem considerando múltiplas entradas e/ou múltiplas saídas, como previsão de carga, onde a influência de variáveis exógenas como condições climáticas e informações de calendário é reconhecida, a identificação de correlações entre as séries disponíveis é de suma importância. Reconhecendo que padrões associados com o calendário podem ser identificados na própria série através de representações sazonais, a utilização de informações climáticas é vital para fins de previsão, visto que o conhecimento prévio de entradas de frentes frias ou quentes pode contribuir decisivamente para a melhoria do desempenho de previsão. Conforme será apresentado no capítulo 3 em conjunto com métodos para determinação das constantes τ e d que definem o espaço de fase, técnicas baseadas em sincronismo generalizado podem ser aplicadas para detecção de relacionamentos entre sistemas caóticos, ampliando a aplicabilidade da teoria do caos no contexto de seleção de entradas.

Seguindo a linha de mensurar relacionamentos de qualquer natureza entre variáveis, técnicas de extração de características através de análise de multi-resolução também podem ser aplicadas. Estes métodos buscam decompor a série histórica em diversas componentes, ou escalas de resolução, que quando combinadas reproduzem o

sinal em estudo. REIS e ALVES DA SILVA [30] utilizam este método para definição de entradas de modelos neurais para previsão de carga. Além de outras variáveis definidas pelos autores, como codificações da hora do dia e informações de temperatura, componentes obtidas através da decomposição da série histórica em *wavelets* são utilizadas como entradas. Mesmo sendo capaz de extrair informações imperceptíveis na escala temporal, a principal restrição à utilização de análise de multiresolução reside na sua limitação a problemas univariados. Relações entre a série de carga e as diversas variáveis exógenas que a influenciam não podem ser identificadas segundo esta teoria.

Ainda na linha de mensurar relacionamentos de qualquer natureza, YANG e STENZEL [76] combinam árvores de regressão e SVMs para previsão de carga. Árvores de regressão são modelos não-paramétricos que geram as previsões através de uma série de regras determinadas automaticamente da própria base de dados. A árvore é gerada através da divisão do conjunto de dados em diversos nós, determinados através da maximização de um dado índice de dispersão. Os nós-folha, situados na base da árvore, são obtidos através do atendimento de um dos critérios: a dispersão das saídas associadas aos padrões constituintes do nó ser menor que um valor máximo especificado, ou o número de padrões integrantes do nó ser menor que um valor mínimo definido pelo usuário. Para nós que atendam somente o critério de dispersão, ou seja, apresentem pequena dispersão em um conjunto com um número razoável de dados, a previsão é dada pela média das saídas dos padrões associados a este nó. Para nós-folha que não atendam o requisito de dispersão, porém apresentem poucos dados, é utilizada uma SVM para estimação da previsão. Visto que para este tipo de nó são disponibilizados poucos dados, uma heurística é utilizada para determinação de quais nós imediatamente anteriores na árvore devam ser incorporados ao conjunto de treinamento. Desta forma, utilizando uma característica intrínseca das árvores de

regressão, a seleção de quais nós utilizar para treinamento dá origem a uma técnica automática de seleção de entradas de SVMs. Resumidamente, dado um conjunto inicial de entradas, a SVM é alimentada somente por aquelas que geram divisão subseqüente nos nós selecionados da árvore original.

A idéia da árvore de regressão é interessante, intuitiva e de simples entendimento, podendo inclusive ser estendida para MLPs. Porém, requer a definição de uma série de parâmetros por parte de especialistas, como níveis máximos de dispersão, número mínimo de padrões por nó, dentre outros, tornando a sua aplicação excessivamente heurística e dependente do problema.

Ao utilizar somente as séries em estudo, as técnicas de filtragem perdem competitividade quando comparadas a métodos mais focados nos modelos. Mesmo sendo capazes de determinar as variáveis mais relacionadas com a saída, estas técnicas não fornecem o melhor conjunto de sinais sob o ponto de vista de previsão, visto que a análise prescinde do modelo, ao contrário dos métodos encapsulados apresentados a seguir.

2.1.2 Métodos encapsulados

Os métodos encapsulados de seleção de entradas buscam melhorias no desempenho do modelo de previsão ao longo do processo de seleção do espaço de entrada. De uma maneira geral, o problema de seleção do espaço de entrada pode ser formulado da seguinte forma. Dado um conjunto de funções $f\left(\underline{x},\underline{w}\right):\mathbb{R}^n \to \mathbb{R}$ e um grupo de sinais de entrada $\underline{x} \in \mathbb{R}^n$, o objetivo da seleção de variáveis reside na determinação do vetor $\underline{\nu} \in \{0,1\}^n$ que solucione o problema dado por [105]:

$$\min_{\underline{\nu}} \tau(\underline{\nu}) = \int V\left[d, f\left(\underline{x}', \underline{w}\right)\right] dP(\underline{x}, d) \tag{2.20}$$
$$s.a$$
$$\underline{\nu} \in \{0,1\}^n$$

onde $\underline{x}' = \left[x_1\nu_1 \quad x_2\nu_2 \quad \ldots \quad x_n\nu_n\right]^t$, $P(\underline{x}, d)$ é a distribuição desconhecida de probabilidade conjunta geradora dos dados e $V\left[d, f\left(\underline{x}', \underline{w}\right)\right]$ uma função de perda. Em outras palavras, dado um conjunto inicial de variáveis de entrada, devem ser selecionadas aquelas que minimizem uma dada medida de desempenho do modelo.

A solução do problema dado pela equação (2.20) exige a avaliação de todos os subconjuntos possíveis de variáveis gerados por $\underline{\nu}$, um problema combinatorial de solução impraticável em tempo finito para casos contendo algumas dezenas de variáveis. Em [99], após a filtragem inicial do espaço de entrada através de análise de correlação, são testados todos os possíveis subconjuntos de entradas gerados por combinações entre as variáveis selecionadas, sendo avaliado um total de $2^n - 1$ modelos e escolhido aquele que apresentar o menor erro para um conjunto de validação. Além de restrições à utilização de índices lineares para seleção de entradas, a avaliação de $2^n - 1$ modelos compromete a aplicação prática da técnica em virtude do elevado esforço computacional requerido para treinamento e avaliação de todas estas estruturas.

O problema combinatorial em (2.20) pode ser abordado através do relaxamento da restrição em $\underline{\nu}$, aproximando esta variável binária por um vetor de variáveis reais $\underline{\sigma} \in \mathbb{R}^n$. Esta aproximação resolve o problema combinatorial, porém traz consigo a questão do ponto a partir do qual uma variável pode ser considerada irrelevante. Em outras palavras, é necessária a determinação do valor $\sigma_0 \in \mathbb{R}$ tal que

$$\nu_i = \begin{cases} 0, \text{ se } \sigma_i \leq \sigma_0 \\ 1, \text{ se } \sigma_i > \sigma_0 \end{cases} \tag{2.21}$$

De forma análoga aos métodos de filtragem, a especificação do limiar σ_0 pode ser substituída pela definição do número de variáveis a serem selecionadas. Tal escolha continua dependente do problema, requisitando a definição heurística por parte de especialistas em aplicações práticas.

Além do relaxamento das restrições, a escolha adequada da medida de desempenho $V\left[d, f\left(\underline{x}', \underline{w}\right)\right]$ é vital para tornar o problema (2.20) tratável em tempo prático. ZHANG e DONG [106] utilizam determinação automática de relevância (ARD) para seleção de variáveis de entrada. Conforme será detalhado no capítulo 4, esta técnica utiliza a evidência para os modelos como medida de desempenho, permitindo a obtenção de um algoritmo iterativo para estimação dos ponderadores $\underline{\sigma}$. Porém, nesta proposta é necessária a especificação de um limiar de relevância por parte de especialistas. Tal definição não é trivial, sendo extremamente dependente da dinâmica da série modelada e do nível de relacionamento desta com as demais variáveis explicativas disponíveis.

Ao considerarem o desempenho do modelo ao longo do processo de seleção do espaço de entrada, os métodos encapsulados são mais atraentes sob o ponto de vista de previsão [60], [105]. Diante do elevado esforço computacional requerido para solução de (2.20), visto que além do número elevado de subconjuntos, para avaliação de cada um deles é necessária a estimativa de um novo modelo, este conjunto de técnicas ainda não encontra muitas aplicações em previsão de carga. Conforme mencionado anteriormente, este empecilho pode ser superado através do relaxamento das restrições e da escolha adequada de uma medida de desempenho, trazendo consigo o problema da definição de limiares de relevância.

Um método baseado na inserção de variáveis aleatórias de prova para definição empírica do limiar de relevância σ_0 será apresentado no capítulo 3. Para cada modelo

proposto nesta tese, medidas de desempenho oriundas da aplicação de inferência *bayesiana* e da obtenção de limites superiores para o erro de generalização são utilizadas permitindo resolver (2.20) em tempo prático, conforme será apresentado nos próximos capítulos.

A seleção do espaço de entrada está diretamente relacionada com o nível de não-linearidade fornecido na saída do modelo neural. Desta forma, determinados os sinais de entrada, é necessário o controle de complexidade, ou regularização, das estruturas utilizadas visando o desenvolvimento de modelos com considerável desempenho de previsão, motivando assim as discussões levantadas na próxima seção.

2.2 Controle de complexidade de RNAs

Conforme mencionado anteriormente, abundam na literatura propostas de modelos neurais para previsão de carga. Porém, analogamente ao problema de seleção de entradas, a preocupação com o controle de complexidade de tais estruturas visando à obtenção de modelos com elevada capacidade de generalização ainda não é uma prática comum. Grande parte das propostas de modelos neurais para previsão de carga, principalmente nos primórdios da aplicação destas técnicas a este tipo de problema, ignora solenemente a questão do ajuste excessivo [14], [15], [17], [26], [28], [39], [42], [44], [46], [48], [69], [70], [71], [72], [73], [74], [83], [84], [86], [87], [89], [90], [91], [92], [97], [100], [102], [107], [108], [109], [110], [111], [112], [113], [114], [115], [116], [117], [118], [119], [120], [121], [122], [123], [124].

O desenvolvimento de uma máquina de aprendizagem visa à estimação de um modelo do processo gerador dos dados e não ao ajuste exato destes, dando origem a estruturas que apresentem desempenho satisfatório para novos conjuntos obtidos segundo o mesmo mecanismo. Esta tarefa pode ser vista como a reconstrução ou

estimação de uma superfície a partir de uma série de exemplos de entrada-saída, ou seja, aproximação de um mapeamento multivariado.

Problemas de aproximação funcional buscam a aproximação, ou interpolação, de uma função contínua $F(\underline{x}):\mathbb{R}^n \to \mathbb{R}$, por uma função $f(\underline{x},\underline{w}):\mathbb{R}^n \to \mathbb{R}$, definida por um vetor de parâmetros $\underline{w} \in \mathbb{R}^M$, utilizando para tal um conjunto de exemplos $D = \{\underline{x}_k, d_k\}$, $k = 1,2,...,N$, $\underline{x}_k \in \mathbb{R}^n$, e $d_k = F(\underline{x}_k)$, para problemas de interpolação, ou $d_k = F(\underline{x}_k) + \zeta_k$, para problemas de aproximação. Neste contexto, existem duas questões vitais. A primeira, relacionada com a escolha da função aproximativa $f(\underline{x},\underline{w})$, que deve ser capaz de aproximar minimamente a função desejada $F(\underline{x})$; e a segunda, associada ao algoritmo a ser utilizado para estimação do conjunto de parâmetros $\underline{w}$.

O teorema da aproximação universal demonstra que modelos neurais com uma única camada escondida contendo número suficiente de neurônios podem aproximar com precisão arbitrária qualquer função contínua não-linear [50]. Assim, para $f(\underline{x},\underline{w})$ representando um modelo neural, um dos principais desafios reside na determinação do número suficiente de neurônios na camada oculta. Este processo é conhecido como estabilização de estrutura, podendo ser aplicado de três formas.

A primeira forma de estabilização de estrutura consiste na comparação entre diversos modelos, com quantidades diferentes de neurônios na camada intermediária, escolhendo a estrutura através da análise do desempenho para um conjunto independente de dados, utilizando técnicas de re-amostragem como validação cruzada, ou índices analíticos de qualificação de modelos, como MDL, do inglês *minimal description lenght* [50], [61], NIC, do inglês *network information criterion* [125], dentre outros [126], [127], [128], [129] e [130].

Comumente em previsão de carga, o desempenho para um conjunto de validação é utilizado como critério para avaliação das estruturas. A diversidade de modelos abrange estruturas com uma ou mais camadas ocultas, com o número de neurônios em cada camada sendo também variado, sendo selecionada aquela que apresentar o menor erro para o conjunto de validação. Aplicações desta técnica de estabilização de estrutura no desenvolvimento de previsores neurais de carga podem ser encontradas em [16], [18], [20], [47], [82], [85], [88], [96], [131], [132], [133], [134], [135], [136], [137], [138], [139], [140], [141]. Este procedimento padece dos problemas inerentes a técnicas baseadas em re-amostragem, tais como: aumento do requisito de dados, visto que deve ser dedicado um conjunto específico para avaliação das estruturas; escolha adequada do conjunto de validação, que deve representar de maneira fidedigna a função a ser aproximada; e elevado custo computacional, já que diversas estruturas devem ser treinadas e avaliadas.

A segunda metodologia parte de um modelo demasiadamente complexo, ou seja, contendo um número elevado de neurônios na camada oculta, sendo aplicados a este modelo alguns algoritmos de poda de rede, com o intuito de extirpar os neurônios em excesso, chegando à dimensão suficiente da camada escondida.

OSOWSKI e SIWEK [24] utilizam um algoritmo de poda de rede conhecido como dano cerebral ótimo, do inglês *optimal brain damage* (OBD), para determinação da estrutura neural. A análise da matriz *hessiana* da função de erro permite determinar a sensibilidade do modelo a um dado peso, determinando assim um procedimento para eliminação de pesos. Apesar de atraente, esta técnica necessita da determinação de níveis de sensibilidade a partir dos quais os pesos podem ser considerados desnecessários ao modelo. Esta definição não é simples, sendo extremamente dependente do problema e do conhecimento de especialistas.

O último procedimento de estabilização de estrutura pode ser considerado como o antípoda da segunda metodologia. Em outras palavras, a idéia consiste em começar com um modelo extremamente simples, por exemplo, com a saída sendo obtida através da combinação linear das entradas, sendo adicionados neurônios à camada oculta ao longo do processo, objetivando a obtenção do número suficiente de neurônios na camada intermediária. Procedimentos baseados nesta metodologia são conhecidos como métodos construtivos [142] e [143].

Na linha dos métodos construtivos, CHOUEIKI *et. al.* [144] utilizam um algoritmo conhecido como correlação em cascata para determinação do número de neurônios na camada oculta. Além disso, utilizam inserção de ruído nas entradas do conjunto de treinamento visando, segundo os autores, evitar que o algoritmo de retropropagação do erro convirja para um mínimo local. Na realidade, a inserção de ruído ao conjunto de treinamento tem analogia com pressupostos de suavidade da função a ser aproximada, utilizados na teoria da regularização. Esta heurística visa o desenvolvimento de modelos que apresentem saídas semelhantes para entradas similares, ou seja, a obtenção de mapeamentos suaves. Esta técnica também é utilizada em [145]. Maiores detalhes sobre a analogia entre inserção de ruído e teoria da regularização serão apresentados ainda nesta seção.

Escolhida a estrutura neural associada à função $f\left(\underline{x},\underline{w}\right)$, resta definir um algoritmo para estimação dos parâmetros $\underline{w}$. Originalmente, os modelos neurais foram desenvolvidos segundo o paradigma da minimização do risco empírico, onde a estimativa do vetor $\underline{w}$ deve ser obtida através da otimização do erro para o conjunto de treinamento dado por:

$$\min_{\underline{w}} E_S\left[f\left(\underline{x},\underline{w}\right)\right] = \frac{1}{N}\sum_{i=1}^{N}\left[d_k - f\left(\underline{x}_k,\underline{w}\right)\right]^2 \tag{2.22}$$

Para minimização deste funcional, foi desenvolvido o algoritmo de retropropagação do erro. Utilizando a regra da cadeia, este algoritmo propaga inversamente ao longo da rede o erro verificado na camada de saída visando o cálculo do gradiente de $E_s\left[f\left(\underline{x},\underline{w}\right)\right]$, dando origem a um procedimento iterativo para solução de (2.22). Existem também algoritmos de segunda ordem, como *Levenberg-Marquardt* [50], para solução do problema dado por (2.22). Um resumo dos algoritmos encontrados na literatura para minimização de $E_s\left[f\left(\underline{x},\underline{w}\right)\right]$ é apresentado no APÊNDICE B.

A concepção original do algoritmo de retropropagação de erro, baseada no princípio da minimização do risco empírico, demonstra preocupação única e exclusiva com o ajuste dos dados disponíveis. Na presença de dados ruidosos, esta abordagem pode conduzir a modelos com reduzida capacidade de generalização em virtude do ajuste excessivo dos dados de treinamento. Além da função geradora dos dados, modelos demasiadamente complexos podem ajustar traços específicos dos dados disponíveis, comprometendo o desempenho para novos dados e degradando a capacidade de generalização do modelo.

Existem algumas heurísticas para evitar a redução da capacidade de generalização devido ao ajuste excessivo dos dados. Uma das mais utilizadas é conhecida como parada antecipada do treinamento, do inglês *early stopping*. Baseada em procedimentos de re-amostragem, esta técnica busca monitorar, ao longo do treinamento, a capacidade de generalização do modelo, com o processo de estimação sendo interrompido em virtude da degradação desta capacidade. Como estimativa da capacidade de generalização, é utilizado o erro obtido para um conjunto de validação selecionado previamente. Desta forma, deve ser escolhido adequadamente tanto o ponto a partir do qual o erro para o conjunto de validação é degradado, quanto o próprio conjunto de validação, que deve representar de maneira fidedigna a função a ser

aproximada. Estas questões evidenciam o caráter heurístico da metodologia. Além disso, para séries temporais como as tratadas em previsão de carga, o uso de um conjunto de validação pode comprometer a modelagem de correlações temporais, visto que dados sequencialmente dispostos no tempo podem ser separados em virtude da seleção do conjunto de validação.

Além de questões de ordem prática, este procedimento apresenta também restrições teóricas. Considerando um conjunto de dados assintoticamente grande, a referência [57] mostra que esta técnica dá origem a modelos com capacidade de generalização inferior em relação àqueles treinados utilizando todo o conjunto de treinamento. Além disso, para bases de dados finitas, os autores mostram que a parada antecipada produz pouco ganho em termos de erro de generalização quando comparada a métodos que utilizam todos os dados, mesmo de posse da partição ótima do conjunto de dados e do ponto ótimo para interrupção do treinamento. Em [58], supondo que modelos com mesmo erro para o conjunto de treinamento são equiprováveis, os autores mostram que esta técnica de controle de complexidade produz modelos com capacidade de generalização esperada inferior em relação aos treinados até a minimização do erro para o conjunto de treinamento. Portanto, além do caráter excessivamente heurístico, análises teóricas aprofundadas restringem a aplicação desta técnica visando à determinação de modelos com boa capacidade de generalização.

Apesar das restrições práticas e teóricas, a parada antecipada do treinamento é uma das técnicas mais utilizadas para controle de complexidade de previsores neurais de carga [18], [20], [21], [47], [94], [98], [103], [136], [138], [146], [147], [148].

Outra heurística desenvolvida para abordagem do problema de ajuste excessivo dos dados tem origem na inserção de ruído aditivo aos padrões de entrada do conjunto de treinamento. Para um dado padrão $\underline{x}_k$, esta técnica está baseada na criação de

versões corrompidas de $\underline{x}_k$, ou seja, $\underline{x}_k^{'} = \underline{x}_k + \underline{\upsilon}$, com $\underline{\upsilon} \in \mathbb{R}^n$ sendo um vetor constituindo de variáveis aleatórias geradas artificialmente segundo uma dada distribuição. A saída desejada associada a $\underline{x}_k^{'}$ é igual à saída associada ao padrão $\underline{x}_k$, ou seja, para um dado par entrada-saída $\left(\underline{x}_k, d_k\right)$, são gerados diversos pares $\left(\underline{x}_k^{'}, d_k\right)$, ampliando assim o conjunto de treinamento. Através da geração de protótipos do sinal de entrada com a saída associada fixa, intuitivamente é esperado que a tarefa de ajustar um específico par $\left(\underline{x}_k, d_k\right)$ seja dificultada, diminuindo a possibilidade de *overfitting*.

Além de não possuir nenhuma justificativa teórica, a inserção de ruído eleva os requisitos computacionais exigidos pelo algoritmo de treinamento, visto que a geração de versões corrompidas de cada padrão aumenta a cardinalidade do conjunto de dados. Entretanto, modelos com capacidade de generalização similar à obtida através desta heurística podem ser obtidos através de um método analítico simples, conhecido como escalonamento do ganho da função de ativação [149].

As funções de ativação sigmoidais utilizadas nos neurônios da camada oculta de MLPs apresentam um parâmetro $a \in \mathbb{R}^+$ conhecido como ganho. Para os MLPs utilizados neste trabalho, essas funções $\phi(x) : \mathbb{R} \to \mathbb{R}$ são dadas por:

$$\phi(x) = \tanh(ax) \tag{2.23}$$

De forma qualitativa, a variação de a implica no aumento, ou diminuição, da região linear de operação destas funções, conforme evidenciado na Figura 2.3. Assim, para MLPs com uma camada escondida e saída linear única, quanto maior a região linear de operação das funções de ativação dos neurônios ocultos, menor será a não-linearidade modelada pela saída. No caso extremo em que as funções sigmoidais estejam definidas apenas em suas respectivas regiões lineares, a saída será dada pela soma ponderada de transformações desta natureza do espaço de entrada. Neste caso, o MLP representa uma

máquina de aprendizagem linear. Portanto, o ajuste dos ganhos a das funções de ativação dos neurônios da camada oculta de um MLP parece ser uma heurística razoável para controle de complexidade destes modelos, visto que tais ganhos estão diretamente relacionados com o nível de não-linearidade modelado pela saída.

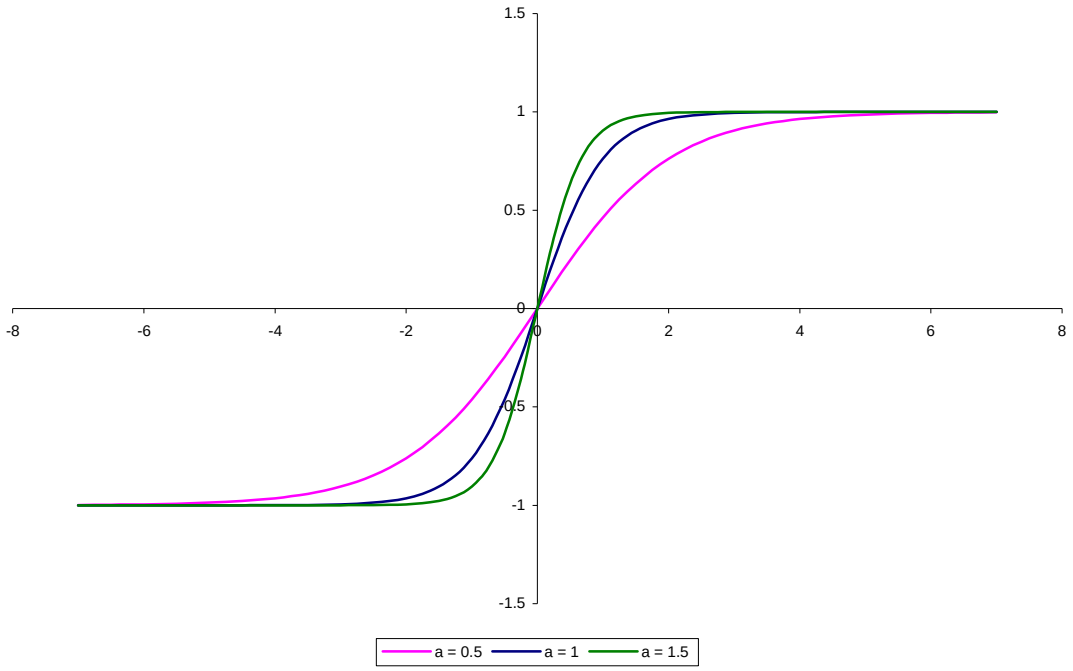

Figura 2.3 – Função tangente hiperbólica utilizando diferentes ganhos a

Neste contexto, REED *et. al.* [149] propuseram uma metodologia de ajuste dos ganhos a das funções de ativação de MLPs a ser realizado após o treinamento, mostrando as semelhanças entre esta metodologia e a heurística de inserção de ruído nas entradas. Desenvolvida para MLPs com uma única camada oculta contendo neurônios não-lineares e uma única saída linear, esta técnica parte de premissas relacionadas com o mecanismo de geração dos dados e com o ruído inserido. Especificamente, é suposto que as amostras do conjunto de treinamento são obtidas segundo uma distribuição uniforme e que o ruído $\underline{\upsilon} \in \mathbb{R}^n$ adicionado às entradas apresenta distribuição *gaussiana*, com vetor média nulo e matriz de covariância $\sigma^2_{ruido} \underline{\underline{I}}$, $\underline{\underline{I}} \in \mathbb{R}^n \times \mathbb{R}^n$. Sob este conjunto

de hipóteses, um MLP treinado através da minimização do risco empírico para o conjunto original de dados irá apresentar capacidade de generalização similar aos modelos estimados através da heurística de inserção de ruído se os ganhos das funções de ativação dos neurônios da camada oculta deste MLP forem multiplicados pelo fator a_j, dado por:

$$a_j = \frac{1}{\sqrt{\left\|\underline{\omega}_j\right\|^2 \sigma_{ruido}^2 + 1}} \qquad (2.24)$$
$$j = 1, 2, ..., m$$

Na equação (2.24) a_j representa o ganho da função de ativação do *j*-ésimo neurônio da camada oculta, $\underline{\omega}_j \in \mathbb{R}^n$ o vetor contendo os pesos que ligam as entradas ao *j*-ésimo neurônio, excluído o respectivo *bias*, e m representa o número de neurônios na camada escondida. O procedimento de ajuste dos ganhos a_j, que no algoritmo de retropropagação do erro são feitos constantes e iguais a um, ou seja, $a_j = 1$ para todos os neurônios, é conhecido como escalonamento do ganho da função de ativação.

Apesar deste procedimento ter sido desenvolvido para MLPs, a equação (2.24) pode ser aplicada a quaisquer modelos *feedforward* com uma única camada oculta contendo neurônios não-lineares e uma única saída linear, sugerindo um procedimento pós-treinamento de ajuste dos ganhos das funções de ativação. Além disso, estes resultados sugerem que o aumento do custo computacional requerido pela heurística de inserção do ruído pode ser evitado através do simples escalonamento dos ganhos a_j de modelos estimados através da minimização do erro para o conjunto de treinamento original. Entretanto, a principal desvantagem desta técnica reside na especificação da variância σ_{ruido}^2 utilizada na equação (2.24), usualmente estimada através de técnicas de re-amostragem, como validação cruzada. Métodos analíticos de qualificação de modelos

utilizados para estabilização de estrutura, como MDL e NIC, dentre outros, também podem ser utilizados para estimação de $\sigma^2_{ruído}$.

Além de heurísticas, existem técnicas analíticas para abordagem do problema de ajuste excessivo dos dados de treinamento. Estas técnicas encontram fundamento na formulação de problemas de reconstrução de superfície, categoria na qual a aproximação funcional pode ser inserida. Especificamente, diante da limitação de dados disponíveis de forma a reconstruir de forma única o mapeamento $F(\underline{x}):\mathbb{R}^n \to \mathbb{R}$ para todo o espaço, o problema de aproximação funcional definido anteriormente é classificado como mal-formulado. Em outras palavras, visto que o conjunto de dados não cobre todo o domínio de $F(\underline{x})$, não é possível reproduzir de forma única este mapeamento para regiões do domínio não contempladas. Em virtude disto, a aplicação direta do princípio da minimização do risco empírico pode resultar em modelos com instabilidade numérica e fraco desempenho de generalização, ou seja, modelos apresentando resultados insatisfatórios para novos padrões, diferentes daqueles utilizados para treinamento, porém provenientes da mesma população [62].

A teoria da regularização fornece subsídios para solução de problemas de reconstrução de superfície mal-formulados como o descrito acima. Nesta teoria, a inserção de conhecimento prévio acerca do problema é necessária para tornar o problema bem formulado, e, na ausência de tal conhecimento, a única informação que pode ser inserida *a priori* diz respeito ao elevado grau de suavidade da função a ser aproximada [50]. Em linhas gerais, a suavidade de uma função está relacionada com as suas características locais, ou seja, o valor da função em um ponto específico depende do valor da mesma nas vizinhanças do último. Esta definição de suavidade está relacionada com a motivação intuitiva do treinamento com inserção de ruído. Qualitativamente, a geração de versões corrompidas das entradas originais, sem

alteração das saídas desejadas associadas, significa que, para padrões de entrada similares, a saída sofrerá pouca ou nenhuma alteração, o que é equivalente a supor que a função a ser aproximada $F(\underline{x})$ apresenta um determinado grau de suavidade. A relação entre o treinamento com inserção de ruído e a teoria da regularização é discutida em [149], onde é mostrado que modelos estimados através da primeira técnica irão apresentar capacidade de generalização similar à obtida por modelos treinados através da aplicação da teoria da regularização.

Assim, esta teoria afirma que os parâmetros $\underline{w}$ da função $f(\underline{x},\underline{w})$ devem ser estimados através da minimização do funcional de *Tikhonov*, dado por:

$$\begin{aligned} &\min_{\underline{w}} R\left[f(\underline{x},\underline{w})\right] = E_S\left[f(\underline{x},\underline{w})\right] + \lambda E_C\left[f(\underline{x},\underline{w})\right] \\ &E_C\left[f(\underline{x},\underline{w})\right] = \left\|Pf(\underline{x},\underline{w})\right\|^2 \end{aligned} \tag{2.25}$$

Na equação (2.25), P é um operador diferencial responsável pela inserção da restrição de suavidade, $\|\cdot\|$ é uma norma definida no espaço ao qual $Pf(\underline{x},\underline{w})$ pertence e λ é o chamado parâmetro de regularização, responsável pelo equilíbrio entre o ajuste dos dados de treinamento e o controle de complexidade, ou regularização, do modelo.

Um dos principais empecilhos na aplicação da teoria da regularização reside na estimativa do parâmetro de regularização λ, responsável pelo equilíbrio entre ajuste dos dados de treinamento e controle de complexidade do modelo. Em previsão de carga, este parâmetro vem sendo estimado por validação cruzada, trazendo consigo todos os problemas inerentes a este tipo de técnica. CHAN *et. al.* [141] utilizam algoritmos genéticos para estimação dos diversos parâmetros de regularização em conjunto com os pesos que definem o MLP, utilizando como função adequabilidade o erro para um conjunto de validação. Além da necessidade de um conjunto dedicado de dados, o esforço computacional requerido pelo algoritmo genético, em virtude do elevado

número de parâmetros a serem otimizados, constitui uma restrição à aplicação prática deste tipo de abordagem.

Além da estimação do parâmetro de regularização λ, a definição do funcional regularizador $E_C\left[f\left(\underline{x},\underline{w}\right)\right]$ responsável pelo controle de complexidade do modelo constitui outro empecilho na aplicação direta da teoria da regularização no desenvolvimento de previsores neurais de carga. Esta escolha é extremamente dependente da dinâmica da série a ser modelada, visto que o operador diferencial está relacionado com premissas de suavidade da função a ser aproximada, requisitando a intervenção de especialistas para seleção do funcional e comprometendo a automatização do processo.

TAYLOR e BUIZZA [25] utilizam o funcional regularizador conhecido como decaimento dos pesos, do inglês *weight decay*, que favorece modelos apresentando pequena magnitude do vetor de pesos. Nesta aplicação, os autores separam os pesos em dois grupos, o primeiro contendo os pesos que ligam às entradas aos neurônios da camada oculta, e o segundo com os pesos que ligam a camada oculta à saída.

DOVEH *et. al.* [103] testam duas técnicas de controle de complexidade baseadas na teoria da regularização, especificamente decaimento dos pesos e eliminação dos pesos, do inglês *weight elimination*. Nesta última técnica, o funcional regularizador busca eliminar pesos excedentes do modelo, através da redução das respectivas magnitudes.

O desenvolvimento da teoria da regularização deu origem às chamadas redes de regularização, fornecendo embasamento teórico para o desenvolvimento das redes de função de base radial (RBFNs) [150]. As redes de regularização podem ser vistas como estruturas *feedforward* com uma única camada oculta contendo N neurônios e uma saída linear, ou seja, um neurônio para cada dado do conjunto de treinamento utilizando

funções de *Green* centradas nos respectivos padrões de entrada. Ao utilizarem um número reduzido de funções de *Green* na camada oculta, as RBFNs podem ser entendidas como aproximações deste tipo de modelo. Existem diversos métodos para determinação dos padrões do conjunto de treinamento a serem utilizados como centros destas funções, utilizando basicamente técnicas não-supervisionadas de agrupamento da base de dados. Maiores detalhes podem ser encontrados em [50].

Os modelos baseados em *kernel*, como as máquinas de vetor suporte (SVM) e as máquinas de vetores relevantes (RVM), guardam analogia com as redes de regularização. Tendo por base o princípio da minimização do risco estrutural, estes modelos buscam a otimização de um funcional similar ao desenvolvido pela teoria da regularização, dado pela Equação (2.25). Além disso, o teorema de *Mercer* garante que os *kernels* utilizados na camada oculta destas estruturas são definidos positivamente, fazendo com que estes núcleos do produto interno também sejam funções de *Green* [62]. Assim, para casos onde todos os padrões de treinamento são considerados vetores suporte ou vetores relevantes, as máquinas baseadas em *kernel* podem ser vistas como uma rede de regularização. Em situações práticas onde somente alguns dados são selecionados como vetores suporte/relevantes, estes modelos fornecem um mecanismo automático para seleção das funções de base de RBFNs. Maiores detalhes sobre estes modelos serão apresentados no capítulo 5.

Apesar de promissores, os modelos baseados em *kernel* ainda não encontram muitas aplicações em previsão de carga, com todos os trabalhos relevantes focando no desenvolvimento de SVMs [76], [77], [78], [79], [80], [81]. Nestas aplicações, os parâmetros que definem este modelo são estimados por validação cruzada, elevando os requisitos computacionais e de dados, comprometendo a utilização prática da metodologia.

Conforme observado acima, o controle de complexidade de modelos neurais de previsão de carga ainda não vem sendo tratado de maneira adequada na literatura. A maioria das propostas sequer menciona o problema, podendo dar origem a modelos com reduzida capacidade de generalização em função do ajuste excessivo dos dados de treinamento. Das propostas que abordam esta questão, grande parte utiliza procedimentos baseados em validação cruzada, como estabilização de estrutura através da seleção do número de neurônios na camada oculta e parada antecipada do treinamento. Além de empecilhos de ordem prática, como elevado requisito de dados e de esforço computacional, estas técnicas apresentam restrições sob o ponto de vista teórico, conforme mostram [57], [58]. Este fato evidencia a necessidade de procedimentos analíticos para regularização de modelos neurais, que possibilitem a utilização de todo o conjunto de dados para treinamento e produzam modelos com considerável capacidade de generalização.

2.3 Modelos neurais autônomos

As seções anteriores mostram que a maioria das propostas de modelos neurais para previsão de carga encontrados na literatura sequer aborda duas questões vitais no desenvolvimento deste tipo de estrutura, a saber: seleção de entradas e controle de complexidade. Existem propostas para tratamento independente destas questões, porém é sabido que as mesmas devem ser tratadas de forma acoplada e conjunta, visto que o nível de não-linearidade disponibilizado pela estrutura neural está diretamente relacionado com o conjunto de entradas utilizado, principalmente para situações práticas onde poucos dados para treinamento são disponíveis [151]. Este fato motiva o desenvolvimento e evidencia a necessidade de modelos neurais autônomos para previsão de carga, incluindo métodos automáticos e acoplados tanto para seleção de entradas quanto para regularização do modelo.

Algumas propostas de métodos automáticos encontradas na literatura utilizam análise de índices de autocorrelação parcial e correlação cruzada para determinação do conjunto de entradas, com a complexidade do modelo sendo controlada através de estabilização de estrutura via seleção do número de neurônios na camada oculta utilizando validação cruzada [96], [133], [134], [139], parada antecipada do treinamento [94], [98], ou ambas [47]. Conforme discutido nas seções 2.1 e 2.2, estas técnicas apresentam uma série de restrições teóricas e práticas.

Dentro deste conjunto geral de métodos automáticos, TSEKOURAS *et. al.* [99] utilizam análise de índices lineares de correlação cruzada da forma apresentada na seção 2.1.1 para seleção do espaço de entrada. Posteriormente, definidas as n entradas a serem utilizadas, são testados todos os possíveis subconjuntos de entradas gerados por combinações entre estas variáveis, sendo avaliado um total de $2^n - 1$ modelos. Estes modelos são regularizados através de estabilização de estrutura, realizada via seleção do número de neurônios na camada oculta por validação cruzada. Parâmetros que definem o algoritmo de retropropagração do erro, como taxas de aprendizado e de momento, também são otimizados via minimização do erro para o conjunto de validação. Além dos problemas relacionados com a utilização de índices lineares para seleção de entradas e com técnicas baseadas em validação cruzada para estabilização de estrutura, a avaliação de $2^n - 1$ modelos gera um problema grave de ordem prática. Problemas de reduzida cardinalidade do espaço de entrada, por exemplo, com 10 entradas, geram 1023 possíveis subconjuntos de variáveis de entrada, sendo necessário o treinamento e avaliação de todas estas estruturas. Além disso, para cada modelo, os demais parâmetros (número de neurônios na camada oculta, taxas de aprendizagem e de momento, dentre outras), também são otimizados visando à escolha do modelo que apresente menor erro para o conjunto de validação. Diante do elevado número de possibilidades a serem

testadas, aplicações práticas da proposta ficam inviabilizadas, em virtude do elevado custo computacional requerido.

CHARYTONIUK e CHEN [95] utilizam uma técnica linear de detecção de variáveis de entrada redundantes para redução do espaço de entrada. Para determinação do espaço de entrada inicial, os autores utilizam análise de índices de autocorrelação linear. Definido o espaço inicial, o estudo dos valores singulares da matriz $\underline{\underline{\Omega}} \in \mathbb{R}^N \times \mathbb{R}^n$, formada pelos n sinais de entradas verificados nos N padrões, permite a detecção de redundância entre as variáveis de entrada. O número de entradas selecionadas é determinado pelo número de valores singulares dominantes, determinados através de um algoritmo de fatoração triangular. Para definição do número de neurônios na camada oculta, é feita uma análise linear das saídas dos neurônios desta camada, de forma análoga à técnica utilizada para identificação de redundância entre os sinais de entrada. Segundo a técnica proposta, modelos excessivamente parametrizados apresentam neurônios redundantes na camada oculta, ou seja, que apresentam saídas colineares. Assim, a análise dos valores singulares da matriz $\underline{\underline{\Sigma}} \in \mathbb{R}^N \times \mathbb{R}^m$, formada pelas saídas dos m neurônios da camada oculta geradas pela propagação dos N padrões de entrada pela rede, permite determinar a existência de neurônios redundantes, sendo estes retirados do modelo final. O número de valores singulares dominantes determina o número de neurônios a ser utilizado, sendo necessária a definição de um patamar para identificação de tais valores dominantes.

A metodologia proposta em [95] é interessante, principalmente no que tange ao procedimento para determinação do número de neurônios na camada escondida. Porém, para seleção de entradas, a técnica é baseada em análise de índices lineares, sendo capaz de detectar somente níveis de redundância desta natureza entre as variáveis, comprometendo a sua aplicação em modelos não-lineares. Além disso, ambos os

procedimentos necessitam da definição de limites inferiores para os valores singulares, visando à determinação das variáveis ou neurônios a serem utilizados. A definição deste limiar não é trivial, sendo extremamente dependente do problema em questão.

Existem propostas de modelos neurais autônomos de previsão de carga que utilizam índices que buscam identificar relacionamentos de quaisquer naturezas entre as variáveis. DREZGA e RAHMAN [21] utilizam uma técnica baseada na teoria do caos para seleção de entradas. Este método, proposto em [19] e descrito sucintamente na seção 2.1, utiliza o método da deformação integral local para determinação do espaço de fase, ou seja, estimação do atraso τ e da dimensão de imersão d. A complexidade do modelo é controlada através de estabilização de estrutura, com o número de neurônios na camada oculta sendo determinado por validação cruzada e os modelos sendo treinados com parada antecipada do treinamento. Apesar da capacidade de identificar relações não-lineares entre as variáveis, técnicas de seleção de entradas baseadas no teorema de *Takens* abordam somente problemas univariados, não contribuindo para a mensuração de interdependência entre a série em estudo e variáveis exógenas. Além disso, técnicas de controle de complexidade baseadas em validação cruzada apresentam os empecilhos teóricos e práticos já apresentados na seção 2.2.

YUAN e FINE [151] evidenciam a necessidade da seleção de variáveis de entrada juntamente com o controle de complexidade do modelo na busca por modelos com boa capacidade de generalização, principalmente em casos práticos onde existem poucos dados para treinamento. Para seleção de variáveis de entrada, os autores utilizam uma estimativa da variância residual da saída. Baseada em diferenças, esta estimativa busca mensurar variações na saída em função de variações na entrada em estudo. Intuitivamente, se pequenas variações de uma dada entrada produzem pequenas variações na saída, é esperado que esta entrada seja uma boa variável explicativa da

saída. Por outro lado, se pequenas variações na entrada produzem elevadas variações na saída, a relação entre estas variáveis pode ser bastante ruidosa, comprometendo a explicação da saída por parte desta variável em função do limitado conjunto de dados. Se infinitos dados fossem disponíveis, relações desse tipo poderiam ser estimadas, porém a metodologia é proposta para casos práticos apresentando conjuntos finitos de dados. Para controle de complexidade, os autores utilizam estabilização de estrutura através da seleção do número de neurônios na camada oculta. Tal seleção é feita utilizando técnicas estatísticas baseadas em modelos conhecidos como *projection pursuit regression* (PPR) [50] e *slicing inverse regression* (SIR) [152]. Especificamente, metodologias desenvolvidas para PPR, que pode ser visto como uma estrutura *feedforward* com uma única camada oculta, são utilizadas para determinação do número de neurônios na camada escondida, com SIR sendo aplicada para estimação das direções de projeção que definem este modelo. Obtida a estrutura, o modelo é então estimado utilizando o algoritmo de retropropagação do erro tradicional.

O procedimento descrito acima inclui uma técnica de seleção de entradas que guarda analogia com a teoria da regularização. O pressuposto sobre relações suaves entre entrada e saída para determinação da importância preditiva da primeira sobre a segunda é similar à premissa de suavidade da função a ser aproximada na teoria da regularização. Entretanto, a estimativa para a variância residual é extremamente sensível a pontos anormais, conhecidos como *outliers*, visto que o estimador utilizado é baseado em diferenças. Além disso, é necessária a definição de um limiar por parte do usuário a partir do qual as variáveis possam ser consideradas irrelevantes. A definição de um valor para este limiar não é trivial, sendo extremamente dependente das características da série a ser aproximada.

As propostas de modelos neurais para previsão de carga apresentadas acima incluem procedimentos automáticos para seleção de sinais de entrada e controle de complexidade. Apesar de algumas incluírem métodos não indicados a modelos neurais, como análise de índices de correlação linear, ou procedimentos para controle de complexidade baseados em validação cruzada, que trazem consigo restrições de ordem prática e teórica, estas técnicas buscam seguir um procedimento geral para especificação de modelos neurais. Especificamente, é necessária a definição de um conjunto inicial de entradas, que pode ser definido através de técnicas de filtragem ou por meio do conhecimento de especialistas. Posteriormente, para fins de redução de cardinalidade e melhoria do desempenho de previsão, este conjunto deve ser processado, utilizando novamente técnicas de filtragem ou encapsuladas. Selecionadas as entradas, a estrutura que melhor representa o conjunto de dados deve ser determinada. Para o caso de MLPs, deve ser definido o número de neurônios na camada oculta. Para SVMs, devem ser especificados os parâmetros que definem o modelo, ou seja, a constante de regularização C, a tolerância ε da função de perda e os parâmetros do *kernel* utilizado. Por fim, o modelo escolhido e alimentado com as variáveis de entrada selecionadas deve ser estimado, gerando as previsões requeridas.

O procedimento geral descrito acima constitui a receita para o desenvolvimento de modelos neurais autônomos para previsão de carga. Porém, como mostra a literatura, as poucas propostas que trilham este caminho utilizam procedimentos inadequados para seleção de entradas e controle de complexidade, além de abordar de forma desacoplada estes problemas. Este fato motiva o trabalho desenvolvido nesta tese, visando à obtenção de metodologias automáticas, acopladas e analíticas para seleção do espaço de entrada e regularização de previsores neurais de carga.

2.4 Resumo e discussão

Este capítulo apresentou modelos não-lineares conhecidos como redes neurais, com destaque para os problemas de seleção do espaço de entrada e controle de complexidade, ou regularização, da estrutura. Conforme mostra a revisão bibliográfica apresentada, estas importantes questões no desenvolvimento de previsores neurais de carga não vêm recebendo a devida atenção na literatura.

O conhecimento de especialistas e operadores do sistema em estudo ainda constitui a principal ferramenta para determinação do conjunto de entradas. Mesmo sendo uma abordagem interessante sob o ponto de vista teórico, a mesma requer a intervenção constante de especialistas, afora o conhecimento de operadores, o que nem sempre é disponível em termos práticos. Além disso, visando estender este tipo de modelo ao nível de barramento, o tratamento individualizado de cada série é impraticável, diante do elevado número de barras a serem consideradas em sistemas de distribuição.

Os métodos encapsulados e de filtragem apresentados na seção 2.1 necessitam da definição de um conjunto inicial de entradas. Em muitas aplicações de sistemas inteligentes, como análise de crédito, visão computacional e reconhecimento de padrões, tal conjunto inicial é disponibilizado. Entretanto, no desenvolvimento de sistemas de previsão somente os históricos das séries envolvidas são disponíveis, evidenciando a necessidade de métodos para escolha do conjunto inicial de entradas.

Métodos clássicos de identificação de sistemas lineares fornecem ferramentas para definição de tal conjunto. Porém, ao capturarem somente relacionamentos lineares, estas técnicas não são indicadas a modelos neurais. A teoria da informação apresenta índices capazes de mensurar interdependências de qualquer natureza entre sinais.

Entretanto, além da ausência de limiares de relevância, esta teoria ainda não apresenta mecanismos fundamentados para identificação não-linear de sistemas.

O conceito de sincronismo generalizado baseado na teoria do caos surge como alternativa para escolha do conjunto inicial. Dados dois sistemas caóticos reconstruídos pelo teorema de *Takens*, a existência de relacionamentos entre eles pode ser identificada utilizando técnicas baseadas neste conceito. Metodologias baseadas em sincronismo generalizado são desenvolvidas nesta tese para definição do conjunto inicial de entradas, sendo apresentadas no capítulo 3.

Para casos práticos com poucos dados disponíveis para treinamento, a seleção das variáveis de entrada adequadas juntamente com o controle de complexidade do modelo é de vital importância na busca por modelos com boa capacidade de generalização [151]. Por outro lado, ao contrário do que recomenda a literatura em seleção de entradas de modelos neurais [59], [60], [105], os métodos encapsulados, mais indicados para fins de previsão visto que consideram a utilidade da variável no desempenho do modelo, não vem sendo utilizados em previsão de carga. Estas questões motivam o desenvolvimento de técnicas encapsuladas de seleção de entradas de modelos neurais para previsão de carga.

Conforme mencionado na seção 2.1.2, a solução direta do problema dado em (2.20) é um problema combinatorial, visto que é necessária a avaliação de todos os possíveis conjuntos de entrada. O relaxamento da restrição do problema traz consigo a necessidade da escolha de uma medida de desempenho adequada, que permita a solução do problema em tempo prático. Conforme será apresentado nos capítulos 4 e 5, a inferência *bayesiana* e o desenvolvimento de um limite superior para o erro de generalização dão origem a medidas de desempenho que permitem solucionar (2.20). Além disso, é necessária a definição de limiares de relevância a partir do qual as

variáveis podem ser descartadas. Visando reduzir a intervenção de especialistas na modelagem, um método para definição empírica de limiares de irrelevância baseado na inserção artificial de variáveis aleatórias de prova [68] é desenvolvido, sendo apresentado no capítulo 3.

Analogamente ao problema de seleção de entradas, a preocupação com o controle de complexidade, ou regularização, de previsores neurais de carga ainda não mereceu destaque na literatura. A utilização de heurísticas baseadas em validação cruzada para seleção do modelo e estimação dos parâmetros compromete a automatização do processo, requisitando o desenvolvimento de métodos analíticos. Além disso, diante do inter-relacionamento entre este problema e a escolha adequada do espaço de entrada, a abordagem independente destas questões não é recomendada visando à obtenção de modelos com elevada capacidade de generalização.

As metodologias propostas nesta tese incluem procedimentos analíticos e automáticos para seleção de entradas e controle de complexidade, evitando o uso de um conjunto de validação específico e os problemas associados a este procedimento. Utilizando funcionais que consideram o ajuste do modelo aos dados e a complexidade da estrutura, estes métodos buscam a estimação de modelos com desempenho satisfatório para novos dados.

A inferência *bayesiana* aplicada ao desenvolvimento de MLPs busca determinar a estrutura mais provável à luz dos dados. Baseada na maximização da evidência, esta metologia inclui mecanismos acoplados para estimação do modelo, seleção de entradas e definição da estrutura. Através da otimização de uma estimativa não-tendenciosa da capacidade de generalização de SVMs, baseada na minimização do limite superior do erro estimado por validação cruzada única (*leave-one-out*), a metodologia automática para especificação deste tipo de modelo para previsão de carga é desenvolvida. Vale

destacar que esta estimativa é obtida sem a necessidade de cálculo efetivo do erro via validação cruzada única, mas sim através de uma expressão analítica para o limite superior desta estatística. A otimização dos parâmetros do *kernel gaussiano* permite divisar um método encapsulado para seleção de entradas. A maximização da evidência de forma análoga à aplicada para MLPs, mesclando as vantagens da inferência *bayesiana* com a aproximação esparsa gerada pelos modelos baseados em *kernel*, dá origem ao método automático para desenvolvimento RVMs para previsão de carga. Estes procedimentos serão apresentados a seguir, começando com o método para definição do conjunto inicial de entradas.

3 Seleção do conjunto de entradas

Conforme apresentado no capítulo 2, a seleção do espaço de entrada constitui uma das principais tarefas no desenvolvimento de modelos neurais para previsão de carga. Na literatura de modelos neurais, os métodos de seleção de variáveis de entrada são divididos em dois grupos, englobando as técnicas de filtragem e as encapsuladas. De uma maneira geral, estas técnicas necessitam da definição de um conjunto inicial de sinais que podem ser utilizados como entradas dos modelos. As formas de processamento deste conjunto, utilizando estatísticas para mensuração de relacionamentos entre variáveis ou medidas de desempenho do modelo, constituem o marco divisor entre as duas classes.

Algumas aplicações de sistemas inteligentes fornecem diretamente o conjunto inicial de entradas. Como exemplos, podem ser citados o problema de análise de crédito, onde informações obtidas por meio de cadastros formam a base de tal conjunto, e reconhecimento de imagens, onde os pixels constituintes da imagem fornecem as informações iniciais. Em previsão de séries temporais, no início do desenvolvimento dos modelos somente os históricos envolvidos são disponíveis. Desta forma, são necessárias técnicas para definição do conjunto inicial a partir das próprias séries, obtendo sinais de entrada relacionados com valores atrasados dos respectivos históricos. A aplicação de métodos clássicos de identificação linear de sistemas em modelos neurais não é indicada, diante da natureza não-linear destas estruturas.

Ao buscarem o conjunto de entradas mais adequado ao modelo em questão, os métodos encapsulados são mais indicados para problemas de previsão. Entretanto, estas técnicas necessitam da definição de limiares de relevância a partir dos quais as variáveis podem ser descartadas. Como mostra a literatura, tais limiares são definidos de forma heurística, requisitando a intervenção constante de especialistas na modelagem. Tendo

em vista o desenvolvimento de modelos neurais autônomos para previsão de carga, é necessária uma técnica para definição empírica de tais patamares.

Técnicas automáticas para definição do conjunto inicial e dos limiares de relevância são desenvolvidas neste capítulo. A teoria do caos, através do teorema de *Takens* e do conceito de sincronismo generalizado, fornece ferramentas para identificação não-linear e detecção de relacionamentos entre sistemas caóticos, surgindo como alternativa para definição do conjunto inicial. Para definição empírica dos limiares de relevância, um método baseado na inserção de variáveis aleatórias de prova é desenvolvido. Maiores detalhes sobre as técnicas são apresentadas a seguir, começando pela teoria do caos.

3.1 Teoria do caos

O desenvolvimento da teoria do caos encontra motivação no estudo de sistemas dinâmicos relativamente sensíveis às condições iniciais. Nestes sistemas determinísticos, variações irregulares de comportamento atribuídas a componentes aleatórias encontram fundamento em pequenas variações nas condições iniciais. Considerando intervalos de tempo discretos, ou seja, $t \in \mathbb{N}$, um sistema dinâmico $F(\underline{X}): \mathbb{R}^D \to \mathbb{R}^D$ em um espaço de estados $\underline{X} \in \mathbb{R}^D$ pode ser definido pela seguinte expressão:

$$\underline{X}(t+1) = F\left[\underline{X}(t)\right] \tag{3.1}$$

Em (3.1), $\underline{X}(t)$ representa o estado do sistema no instante de tempo t. Para o caso em que a variável temporal t é contínua, o sistema dado em (3.1) é substituído por um conjunto de equações diferenciais.

Em sistemas determinísticos como o da equação (3.1), a partir do estado atual $\underline{X}(t)$, todos os estados subjacentes do sistema podem ser obtidos. Assim, além de

depender de $F(\underline{X})$, a evolução ou trajetória do sistema no espaço de estados depende do estado inicial do mesmo. O conjunto de condições iniciais que conduzem assintoticamente o sistema para uma dada região do espaço de estados são chamadas bases de atração para esta região, denominada atrator [153]. Os sistemas dinâmicos estudados na teoria do caos apresentam esta característica. Invariantes em relação à evolução do sistema, estas regiões apresentam formatos geométricos exóticos, sendo por isso também conhecidas como atratores estranhos [65].

As definições apresentadas acima são válidas no espaço multidimensional ao qual o sistema $F(\underline{X})$ está confinado. Contudo, na prática somente registros escalares $x(t)$, $t = 1, 2, ..., N$, deste sistema, verificados através de uma função de medição $s(\underline{X}) : \mathbb{R}^D \to \mathbb{R}$, são disponíveis, ou seja,

$$x(t) = s\left[\underline{X}(t)\right] + \eta(t) \tag{3.2}$$

Na equação acima, $\eta(t)$ representa o ruído de medição verificado no instante t.

A função de medição $s(\underline{X})$ comprime a informação multivariada contida em $\underline{X}(t)$ em uma medida escalar $x(t)$, projetando variáveis não-observáveis do sistema em uma escala real. Diante do desconhecimento acerca de tal função em conjunto com a presença do ruído de medição $\eta(t)$, a reconstrução fidedigna do espaço original $\underline{X}(t)$ a partir da sequência de medições $x(t)$ é impossível. Entretanto, a estimação perfeita do espaço original é desnecessária, sendo suficiente a definição de um novo espaço de representação cujo atrator seja equivalente ao existente no espaço original. As condições para obtenção deste novo espaço de representação a partir exclusivamente das medições $x(t)$ são definidas no teorema de *Takens*, discutido na próxima seção.

3.1.1 Teorema de *Takens*

Diante da incapacidade de obter o espaço de estados original a partir de um conjunto de medidas oriundas de um sistema caótico, TAKENS [65] mostrou ser desnecessária a obtenção perfeita de tal espaço para o estudo de sistemas determinísticos como o apresentado na equação (3.1). Este sistema pode ser analisado em um novo espaço de representação $\underline{x} \in \mathbb{R}^d$, denominado espaço reconstruído, obtido através de um conjunto de medições $x(t)$ e definido pela equação:

$$\underline{x}(t) = \left[x(t) \quad x(t-\tau) \quad \ldots \quad x\left(t-(d-1)\tau\right) \right]^t \tag{3.3}$$

Na equação (3.3), τ e d são parâmetros conhecidos como atraso e dimensão da imersão.

Para equivalência entre os atratores nos espaços reconstruído $\underline{x} \in \mathbb{R}^d$ e original $\underline{X} \in \mathbb{R}^D$, algumas condições devem ser atendidas [154]. Primeiro, deve existir um mapeamento $Z(\underline{x}) : \mathbb{R}^d \to \mathbb{R}^D$ contínuo e biunívoco, ou seja, pontos distantes no espaço $\underline{x}$ não podem ser mapeados de forma próxima no espaço original $\underline{X}$. Este mapeamento e o respectivo mapeamento inverso $Z^{-1}(\underline{X}) : \mathbb{R}^D \to \mathbb{R}^d$ também devem ser suaves e continuamente diferenciáveis. Atendidas estas condições, o espaço reconstruído $\underline{x} \in \mathbb{R}^d$ é uma imersão de $\underline{X} \in \mathbb{R}^D$, visto que o atrator neste novo espaço está relacionado de forma suave com o atrator no espaço original, preservando propriedades físicas do atrator desconhecido [156].

Para existência do mapeamento $Z(\underline{x}) : \mathbb{R}^d \to \mathbb{R}^D$ e atendimento das condições supracitadas, o atraso τ e a dimensão da imersão d devem ser escolhidos de forma adequada. Considerando um conjunto de dados de cardinalidade infinita e precisão ilimitada, ou seja, $\eta(t) = 0$ em (3.2), o teorema de *Takens* demonstra preocupação

somente com a definição da dimensão d, sendo válido para escolhas arbitrárias do atraso da imersão τ [65]. Porém, em aplicações práticas com número limitado de dados contaminados por ruído, a escolha deste parâmetro também é crucial para a reconstrução adequada do espaço de estados [153], [155], [156].

Observando a equação (3.3) e considerando uma reconstrução bidimensional ($d = 2$) utilizando uma base de dados finita contaminada por ruído, a escolha de valores pequenos para τ pode produzir trajetórias confinadas à diagonal do espaço reconstruído $\underline{x}(t) = \left[x(t) \quad x(t-\tau) \right]^t$, comprimindo o atrator neste espaço [155]. De outra forma, valores elevados para este parâmetro conduzem a reconstruções que utilizam direções praticamente independentes, descaracterizando o relacionamento com o atrator original. Estas questões confirmam a necessidade da definição de critérios para especificação adequada do atraso da imersão τ.

Existem diversos critérios propostos na literatura para definição de τ, baseados em argumentos geométricos [157], [158] e estatísticos, com os últimos sendo mais utilizados [153]. Dentre os métodos estatísticos, o estudo da função de autocorrelação amostral do sinal $x(t)$, $r_{XX}(k)$, dada pela equação (2.8), é a técnica mais simples e popular. Buscando um compromisso entre compressão do atrator e reconstruções baseadas em direções descorrelacionadas, o primeiro mínimo do módulo de $r_{XX}(k)$ pode ser utilizado como estimativa para o atraso da imersão. Esta escolha evita o confinamento em torno da diagonal do espaço reconstruído, em virtude do correlacionamento mínimo entre as direções, trazendo consigo alguma informação sobre o sistema no instante de tempo atual, visto que o mínimo global do módulo de $r_{XX}(k)$ sinalizando ausência de interdependência linear ainda não foi atingido.

A capacidade de mensurar somente relacionamentos lineares consiste na principal restrição à aplicação da função de autocorrelação $r_{XX}(k)$ na análise para definição do atraso τ. De uma maneira geral, a escolha do primeiro mínimo do módulo de $r_{XX}(k)$ não evita o confinamento do atrator, visto que interdependências não-lineares eventualmente existentes para este atraso podem comprimir o atrator em trajetórias desta natureza. Conforme mencionado na seção 2.1.1, a informação mútua avalia dependências gerais entre variáveis, podendo ser utilizada de forma análoga à função de autocorrelação para escolha do atraso da imersão τ.

A definição da informação mútua $I_X(k)$ entre realizações do sinal $x(t)$ defasadas k instantes de tempo depende da forma na qual as respectivas entropias são estimadas. Considerando histogramas da forma definida no APÊNDICE A, com intervalo de discretização h_{hist} gerando p subintervalos, as entropias definidas nas equações (2.13) e (2.14) passam a ser dadas por:

$$H_X(k) = -\sum_{i=1}^{p} P\left[x(t-k) \in \nu_i\right] \log P\left[x(t-k) \in \nu_i\right] \tag{3.4}$$

$$H_{XX}(k) = -\sum_{i=1}^{p}\sum_{j=1}^{p} P\left[x(t) \in \nu_i, x(t-k) \in \nu_j\right] \log P\left[x(t) \in \nu_i, x(t-k) \in \nu_j\right] \tag{3.5}$$

Nas equações acima, $H_X(k)$ representa a entropia do sinal defasado $x(t-k)$, $t = k+1, k+2, ..., N$, com $H_{XX}(k)$ respondendo pela entropia conjunta entre o sinal original $x(t)$ e o respectivo processo defasado. Nestas equações, $P\left[x(t-k) \in \nu_i\right]$ representa a probabilidade de uma dada realização do sinal defasado $x(t-k)$ pertencer ao *i*-ésimo intervalo ν_i. Vale destacar que $H_X(0)$ em (3.4) é a entropia da sequência de medições $x(t)$.

Considerando métodos para estimação de funções de densidade de probabilidade, as entropias dadas pelas equações (3.4) e (3.5) são substituídas pelas respectivas entropias diferenciais. Conforme discutido no APÊNDICE A, as entropias diferenciais podem ser estimadas da seguinte forma:

$$\hat{h}_X(k) = -\sum_{t=k+1}^{N} \hat{f}\left[x(t-k)\right] \log \hat{f}\left[x(t-k)\right] \tag{3.6}$$

$$\hat{h}_{XX}(k) = -\sum_{t=k+1}^{N} \sum_{j=k+1}^{N} \hat{q}\left[x(t), x(t-k)\right] \log \hat{q}\left[x(t), x(t-k)\right] \tag{3.7}$$

Em (3.6) e (3.7), $\hat{f}\left[x(t-k)\right]$ é a estimativa da função de densidade de probabilidade do sinal defasado $x(t-k)$, com $\hat{q}\left[x(t), x(t-k)\right]$ representando a função de densidade conjunta estimada entre o sinal original $x(t)$ e seu correspondente defasado.

Utilizando a relação entre entropia e informação mútua dada na equação (2.16), a informação mútua $I_X(k)$ entre realizações do sinal $x(t)$ defasadas k instantes de tempo pode ser estimada através da seguinte expressão:

$$I_X(k) = H_X(0) + H_X(k) - H_{XX}(k) \tag{3.8}$$

Conforme discutido na seção 2.1.1, a equação (3.8) também é válida para entropias diferenciais.

De forma análoga à análise baseada na função de autocorrelação, o primeiro mínimo da informação mútua $I_X(k)$ pode ser utilizado como estimativa para o atraso da imersão τ. Porém, ao contrário da função de autocorrelação $r_{XX}(k)$, o cálculo de $I_X(k)$ não é trivial devido às dificuldades relacionadas à estimação das probabilidades por meio de histogramas ou das funções de densidade de probabilidade utilizando métodos não-paramétricos. Apesar deste empecilho, este método é o mais recomendado

em virtude da capacidade de $I_X(k)$ mensurar níveis de interdependência de qualquer natureza [153], [155], [156].

Definido o atraso τ da imersão, resta escolher a dimensão d do espaço reconstruído. Segundo TAKENS [65], o espaço reconstruído $\underline{x} \in \mathbb{R}^d$ obtido pela expressão (2.3) é uma imersão do espaço original $\underline{X} \in \mathbb{R}^D$ se $d > 2D$. Visto que a dimensão D do espaço de estados original é desconhecida, métodos para definição da dimensão de imersão d são necessários. Existem diversas técnicas na literatura, baseadas na decomposição de valores singulares da matriz de covariância da matriz reconstruída $\underline{x}(t)$ e no cálculo de características invariantes do atrator, discutidas em [155], [156]. Além de computacionalmente intensivas, estas técnicas são subjetivas, requerendo a intervenção de especialistas na análise.

A reconstrução em espaços de cardinalidade reduzida produz intersecções espúrias no atrator, em função da representação inadequada do sistema. Cruzamentos de trajetória desta natureza devem ser evitados, visando atender as características do mapeamento $\mathrm{Z}(\underline{x}): \mathbb{R}^d \to \mathbb{R}^D$ que garantem o espaço reconstruído como uma imersão do espaço original. Uma das técnicas mais populares para estimação da dimensão de imersão d tem por base a identificação destas trajetórias, sendo conhecido como método dos falsos vizinhos mais próximos [159]. Esta denominação encontra fundamento na forma na qual as intersecções espúrias podem ser identificadas observando a mudança na vizinhança de um dado ponto em função do aumento da dimensão de imersão. Pontos vizinhos devido à dinâmica do sistema permanecem nesta condição quando d sofre acréscimo. Aqueles que deixam a vizinhança em virtude do aumento da dimensão são denominados falsos vizinhos, visto que estão situados na vizinhança devido à reconstrução incompleta do atrator.

Para identificação dos falsos vizinhos, alguns parâmetros devem ser especificados, aumentando a subjetividade do método. Para contornar esta questão, CAO [160] desenvolveu um método baseado na idéia dos falsos vizinhos mais próximos, porém com menos parâmetros a serem especificados. Utilizando a reconstrução em espaço de estados dada na equação (3.3), seja $\Delta(i,j,d)$ a distância entre pontos $\underline{x}(i)$ e $\underline{x}(j)$ reconstruídos na dimensão d, calculada pela expressão:

$$\Delta(i,j,d) = \max_{k=1,\ldots,d} \left| \underline{x}_k(i) - \underline{x}_k(j) \right| \tag{3.9}$$

Para o cálculo de $\Delta(i,j,d)$ pela equação acima, também conhecida como norma infinita, $\underline{x}_k(i)$ representa o *k*-ésimo elemento do vetor de estados verificado no instante i. O vizinho mais próximo de $\underline{x}_k(i)$ é o ponto para o qual $\Delta(i,j,d)$ é mínimo, ou seja,

$$n(i,d) = \arg\left[\min_{j=(d-1)\tau+1,\ldots,N} \Delta(i,j,d) \right] \tag{3.10}$$

Em (3.10), $n(i,d)$ é o índice associado com o vetor $\underline{x}\left[n(i,d)\right]$ que minimiza a distância $\Delta(i,j,d)$ entre $\underline{x}(i)$ e todos os demais pontos disponíveis no espaço reconstruído de dimensão d.

Seja a relação $a(i,d)$ entre vizinhos mais próximos em dimensões consecutivas dada por:

$$a(i,d) = \frac{\Delta\left[i, n(i,d), d+1\right]}{\Delta\left[i, n(i,d), d\right]} \tag{3.11}$$

Em (3.11), se $\Delta\left[i, n(i,d), d\right]$ for nulo, $n(i,d)$ é substituído pelo índice do vizinho mais próximo adjacente, obtido desconsiderando o ponto $\underline{x}\left[n(i,d)\right] = \underline{x}(i)$. Na proposta original do método dos falsos vizinhos mais próximos [159], uma estatística similar à relação $a(i,d)$ é calculada, com um ponto sendo qualificado como falso

vizinho se esta estatística for maior que um patamar especificado heuristicamente. Para evitar a especificação deste patamar, o valor médio da relação $a(i,d)$ pode ser analisado, dando origem à estatística $J(d)$ calculada pela expressão:

$$J(d) = \frac{1}{N-(d-1)\tau} \sum_{i=(d-1)\tau+1}^{N} a(i,d) \tag{3.12}$$

A variação relativa $\delta(d)$ desta estatística em função do aumento da dimensão da imersão é representada por:

$$\delta(d) = \frac{J(d+1)}{J(d)} \tag{3.13}$$

Segundo [160], para séries temporais oriundas de um atrator, a variação $\delta(d)$ estabiliza quando a dimensão de imersão d é maior que um valor d_0. Portanto, uma estimativa para a dimensão mínima de imersão é dada por $d = d_0 + 1$.

A detecção da dimensão d_0 a partir da qual $\delta(d)$ é estabilizada insere uma componente heurística na implementação automática da técnica. Em [160], a identificação da dimensão é feita de forma visual, através do acompanhamento por meio de gráficos da evolução desta estatística. Visando automatizar este processo, métodos para monitoramento de $\delta(d)$ e identificação automática de d_0 são necessários.

A variação no comportamento de $\delta(d)$ pode servir como indicativo da estabilização desta estatística. Entretanto, a escolha deste critério requer a definição de um novo patamar para estabilidade. De outra forma, o ajuste da curva de evolução de $\delta(d)$ em função de d fornece mecanismos para identificação deste patamar. Considerando um modelo de regressão linear, testes de relevância sobre o coeficiente de

inclinação da reta modelando $\delta(d)$ e d podem ser utilizados para detecção de estabilidade.

Especificamente, seja $d_{\max}$ a dimensão máxima para a qual a estatística $\delta(d)$ é calculada, supondo que a estabilização de $\delta(d)$ já tenha ocorrido para algum $d_0 < d_{\max}$. De posse dos pares $\left[d, \delta(d)\right]$, $d = 1, 2, ..., d_{\max}$, um modelo de regressão linear é estimado, sendo realizado um teste ao nível de significância α para a hipótese nula considerando o coeficiente angular igual a zero. Se a hipótese nula puder ser rejeitada, o primeiro par $\left[d, \delta(d)\right]$ é retirado, sendo estimado um novo modelo considerando os pontos para $d = 2, ..., d_{\max}$. Este procedimento é repetido até a hipótese nula não ser rejeitada, com a dimensão de imersão sendo dada pelo primeiro ponto utilizado na estimação do modelo linear.

A heurística definida acima depende da definição de dois parâmetros, a saber: dimensão máxima $d_{\max}$ e nível de significância α do teste de hipótese. A escolha do nível de significância α, apesar de heurística, é mais intuitiva e conhecida na literatura [161], [162]. A definição de $d_{\max}$ está diretamente relacionada com o esforço computacional, visto que devem ser realizados cálculos para dimensões maiores que d_0.

O teorema de *Takens* fornece as condições para reconstrução de sistemas dinâmicos a partir de medições escalares oriunda deste sistema. Assim como o teorema, os métodos para definição do atraso τ e da dimensão de imersão d partem do pressuposto de que a série temporal considerada comprime informações multivariadas sobre um sistema caótico de ordem elevada. Desta forma, para correta aplicação das técnicas, são necessários testes para detecção de características caóticas no conjunto de

dados em estudo. Um método para identificação de traços caóticos é baseado no cálculo dos chamados expoentes de *Lyapunov*, apresentados na próxima seção.

3.1.2 Expoentes de *Lyapunov*

Uma das principais características de sistemas caóticos reside na incapacidade de prever o seu comportamento futuro apesar do traço fundamentalmente determinístico de tais sistemas. Esta questão encontra fundamento na instabilidade das soluções em virtude da sensibilidade a condições iniciais. Soluções inicialmente próximas no espaço de estados divergem à medida que o tempo evolui, diminuindo o nível de previsibilidade do sistema. Para sistemas caóticos, esta divergência entre trajetórias ocorre a taxas exponenciais cujo estudo fornece indicadores do grau de caoticidade de um dado sistema, conhecidos como expoentes de *Lyapunov*.

Considerando sistemas em tempo contínuo, ou seja, $t \in \mathbb{R}$, seja uma trajetória $\underline{\gamma}(t) \in \mathbb{R}^D$ representando uma solução do sistema dinâmico dado por:

$$\frac{d}{dt}\underline{\gamma}(t) = \varphi\left[\underline{\gamma}(t)\right] \tag{3.14}$$

Linearizando a equação acima, a evolução do vetor tangente $\underline{\xi}(t) \in \mathbb{R}^D$ em um espaço tangente a um dado ponto da trajetória $\underline{\gamma}(t)$ pode ser obtida pela expressão [163]:

$$\frac{d}{dt}\underline{\xi}(t) = T\left[\underline{\gamma}(t)\right]\underline{\xi}(t) \tag{3.15}$$

Em (3.15), $T\left[\underline{\gamma}(t)\right]$ representa a matriz jacobiana de $\varphi\left[\underline{\gamma}(t)\right]$. A solução deste sistema linear não-autônomo é dada por:

$$\underline{\xi}(t) = A^t\,\underline{\xi}(0) \tag{3.16}$$

Na equação acima, $A\left[\underline{\gamma}(t)\right]: \mathbb{R}^D \to \mathbb{R}^D$ é o operador linear responsável pelo mapeamento de $\underline{\xi}(0)$ para $\underline{\xi}(t)$, diretamente relacionado com o ponto da trajetória

$\underline{\gamma}(t)$ no qual o sistema (3.14) é linearizado. Desta forma, a taxa de divergência exponencial média do vetor tangente $\underline{\xi}(t)$ é definida pela expressão [163]:

$$\lambda\left[\underline{x}(0),\underline{\xi}(0)\right]=\lim_{t\to\infty}\left[\frac{1}{t}\ln\frac{\left\|\underline{\xi}(t)\right\|}{\left\|\underline{\xi}(0)\right\|}\right] \tag{3.17}$$

Em (3.17), $\left\|\underline{\xi}(t)\right\|$ representa uma norma em relação a métricas de *Riemannian*. Além disso, existe um conjunto de vetores ortonormais $\underline{e}_i \in \mathbb{R}^D$ que formam uma base para $\underline{\xi}(0)$, de forma que $\lambda_i\left[\underline{x}(0)\right]=\lambda\left[\underline{x}(0),\underline{e}_i\right]$. Estas taxas podem ser ordenadas de forma decrescente de acordo com a sua magnitude, dando origem ao espectro de expoentes de *Lyapunov* $\Lambda=\left(\lambda_1,\lambda_2,\ldots,\lambda_D\right)$ independente das condições iniciais $\underline{x}(0)$ para sistemas ergódicos [163].

O espectro de expoentes de *Lyapunov* de um dado sistema dinâmico é uma das diversas medidas invariantes a transformações que caracterizam os atratores de sistemas caóticos. Desta forma, os expoentes estimados para o atrator no espaço reconstruído fornecem subsídios para qualificação de aspectos dinâmicos do sistema original. Espectros com expoente máximo negativo indicam a existência de um ponto de estabilidade para a evolução do sistema, aproximando de forma exponencial trajetórias distintas. Sistemas que convergem para trajetórias cíclicas apresentam expoente máximo nulo, sendo classificados como marginalmente estáveis [153]. Visto que uma das principais características de sistemas caóticos reside na divergência entre trajetórias inicialmente próximas, espectros com expoente de *Lyapunov* máximo positivo indicam existência de caos [153]. De acordo com a definição dada pela equação (3.17), sistemas determinísticos contaminados por ruído, indicando ausência de previsibilidade perfeita em função da componente aleatória, apresentam expoente máximo infinito.

A definição apresentada na equação (3.17) para sistemas caóticos em tempo contínuo apresenta pouca utilidade prática, visto que as equações de movimento necessárias para obtenção do espaço tangente são geralmente desconhecidas. Além disso, intervalos de tempo infinitesimais necessários para estimação dos vetores tangentes $\underline{\xi}(t)$ são inacessíveis em dados experimentais [164]. Por último, na presença de ruído associado a erros de medição, o cálculo do espectro de *Lyapunov* para dados oriundos de sistemas determinísticos, segundo (3.17), produzirá resultados espúrios relacionados a sistemas puramente estocásticos. Estas questões confirmam a necessidade de métodos práticos para estimação do espectro de expoentes de *Lyapunov* de sistemas dinâmicos a partir de dados experimentais.

A partir de um conjunto de medições amostradas em tempo discreto $t \in \mathbb{N}$, seja uma reconstrução $\underline{x}(t) \in \mathbb{R}^d$, dada pela equação (3.3), do espaço de estados original $\underline{X}(t) \in \mathbb{R}^D$. Neste novo espaço, os vetores tangentes podem ser estimados observando a evolução temporal das vizinhanças de um dado ponto. Para isso, seja $\underline{x}(t)$ um ponto no espaço reconstruído verificado no instante t e $\underline{x}\left[n(t,d)\right]$ seu respectivo vizinho mais próximo, no senso das equações (3.9) e (3.10). Novamente, se $\underline{x}\left[n(t,d)\right] = \underline{x}(t)$, $n(t,d)$ é dado pelo vizinho mais próximo obtido desconsiderado $\underline{x}\left[n(t,d)\right]$. A evolução temporal da diferença entre estes pontos é dada pela seguinte expressão:

$$\underline{x}(t+k) - \underline{x}\left[n(t,d)+k\right] = \mathrm{K}\left\{\underline{x}(t) - \underline{x}\left[n(t,d)\right]\right\} \tag{3.18}$$

Na equação acima, $\mathrm{K}\left[\underline{x}(t)\right]: \mathbb{R}^d \to \mathbb{R}^d$ é um mapeamento relacionando a diferença entre pontos próximos no instante t e a discrepância entre estes pontos k passos à frente. Assim, as diferenças $\underline{x}(t) - \underline{x}\left[n(t,d)\right]$ e $\underline{x}(t+k) - \underline{x}\left[n(t,d)+k\right]$ representam

aproximações dos vetores tangentes $\underline{\xi}(t)$, com $\mathrm{K}\left[\underline{x}(t)\right]:\mathbb{R}^d \to \mathbb{R}^d$ responsável pela estimação do operador linear $A\left[\underline{\gamma}(t)\right]:\mathbb{R}^D \to \mathbb{R}^D$.

A partir da aproximação do espaço tangente dada pela (3.18), existem diversas propostas para o cálculo do espectro de expoentes de *Lyapunov*. A principal diferença entre as técnicas reside na forma na qual o mapeamento $\mathrm{K}\left[\underline{x}(t)\right]:\mathbb{R}^d \to \mathbb{R}^d$ é estimado para cada instante de tempo. Grande parte das aplicações utiliza linearizações de primeira ordem da equação (3.18) em séries de *Taylor* [163], [164], [165], [166], podendo também ser utilizadas aproximações de ordem elevada [167] ou até mesmo modelos não-lineares como redes neurais [168]. Apesar das considerações apresentadas em [167] recomendando o uso de aproximações polinomiais, a utilização de modelos mais complexos neste contexto aumenta o requisito de dados, comprometendo a aplicabilidade da metodologia. Na prática, considerando um número limitado de medições, linearizações de primeira ordem de $\mathrm{K}\left[\underline{x}(t)\right]$ são recomendadas. Considerando um mapeamento linear, a equação (3.18) passa a ser dada por:

$$\underline{x}(t+k)-\underline{x}\left[n(t,d)+k\right]=\underline{\underline{\mathrm{K}}}_t\left\{\underline{x}(t)-\underline{x}\left[n(t,d)\right]\right\} \tag{3.19}$$

Em (3.19), $\underline{\underline{\mathrm{K}}}_t \in \mathbb{R}^d \times \mathbb{R}^d$ é uma matriz representando uma aproximação de primeira ordem do mapeamento $\mathrm{K}\left[\underline{x}(t)\right]$. Para $k=\tau$ (atraso da imersão), a matriz cheia $\underline{\underline{\mathrm{K}}}_t \in \mathbb{R}^d \times \mathbb{R}^d$ é substituída pela expressão [167]:

$$\underline{x}(t+\tau)-\underline{x}\left[n(t,d)+\tau\right]=\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & 1 \\ k_1(t) & k_2(t) & k_3(t) & \dots & k_d(t) \end{bmatrix}\left\{\underline{x}(t)-\underline{x}\left[n(t,d)\right]\right\} \tag{3.20}$$

Para estimação dos d parâmetros $k_i(t)$ da matriz dada na equação (3.20), o método de mínimos quadrados pode ser utilizado. Para tornar o problema determinado, são necessários no mínimo d pontos, ou seja, devem ser selecionados os d vizinhos mais próximos de $\underline{x}(t)$ em relação à distância definida na equação (3.9). Entretanto, a escolha deste número mínimo de vizinhos permite a estimação de um mapeamento único, diminuindo a redundância da estimativa. Desta forma, é recomendado o aumento do número de vizinhos N_V escolhidos para o cálculo de $\underline{\underline{\mathrm{K}}}_t$. Segundo [166], [167], uma escolha razoável reside no dobro do total de parâmetros a serem estimados, ou seja, devem ser selecionados $N_V = 2d$ vizinhos mais próximos do ponto $\underline{x}(t)$.

Utilizando a equação (3.20), a matriz $\underline{\underline{\mathrm{K}}}_t$ responsável pela aproximação do mapeamento no espaço de vetores tangentes pode ser calculada para os pontos $\underline{x}(t)$ tais que $t = 1, 2, ..., N - \tau$. Conforme mostram ECKMANN *et.al.* [165], a decomposição sucessiva destas matrizes em duas componentes, uma ortogonal $\underline{\underline{Q}}_t$ e outra triangular superior $\underline{\underline{R}}_t$ com elementos diagonais positivos, pode ser utilizada para estimação do espectro de expoentes de *Lyapunov* do sistema reconstruído $\underline{x}(t)$. Conhecida como fatoração QR, esta decomposição é dada por:

$$\underline{\underline{\mathrm{K}}}_t \underline{\underline{Q}}_{t-1} = \underline{\underline{Q}}_t \underline{\underline{R}}_t, \; t = 1, 2, ..., N - \tau \tag{3.21}$$

Em (3.21), $\underline{\underline{Q}}_0$ é uma matriz unitária e ortogonal a $\underline{\underline{\mathrm{K}}}_1$, ou seja, suas colunas formam uma base para o espaço representado pelas colunas de $\underline{\underline{\mathrm{K}}}_1$. A decomposição dada em (3.21) é única, exceto para os casos onde os elementos da diagonal de $\underline{\underline{R}}_t$ são nulos, podendo ser obtida através do algoritmo modificado de *Gram-Schmidt* para ortogonalização de matrizes [166].

De posse de um número suficiente de matrizes $\underline{\underline{R}}_t$, $t = 1, 2, ..., N - \tau$, o espectro de expoentes de *Lyapunov* do espaço reconstruído, $\hat{\Lambda} = \left(\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_d\right)$ pode ser estimado pela equação [167]:

$$\hat{\lambda}_i = \frac{1}{K\tau} \sum_{t=1}^{N-\tau} \ln\left[\underline{\underline{R}}_t\right]_{ii}, \; i = 1, 2, ..., d \tag{3.22}$$

Em (3.22), $\left[\underline{\underline{R}}_t\right]_{ii}$ representa o *i*-ésimo elemento da diagonal da matriz $\underline{\underline{R}}_t$.

Conforme mencionado anteriormente, a análise dos expoentes de *Lyapunov* permite a identificação de traços caóticos. Assim, a obtenção de $\hat{\lambda}_1$ positivo sinaliza a presença de características caóticas no sistema reconstruído. Visto que à luz do teorema de *Takens* este espaço de representação é uma imersão do espaço original, este fato evidencia a existência de caoticidade em tal sistema.

Além de detectar características caóticas, os expoentes de *Lyapunov* podem ser utilizados para o cálculo da dimensão do atrator. Dado um espectro $\Lambda = \left(\lambda_1, \lambda_2, \ldots, \lambda_d\right)$ associado a um dado sistema caótico, a dimensão ν do atrator deste sistema pode ser estimada pela seguinte expressão [169]:

$$\nu = l + \frac{1}{\lambda_{l+1}} \sum_{i=1}^{l} \lambda_i \tag{3.23}$$

Em (3.23), l é o maior inteiro para o qual a soma dada em (3.23) é positiva.

Diante dos cálculos envolvidos nas equações (3.21) e (3.22), a análise do espectro de *Lyapunov* do sistema reconstruído pode ser comprometida na presença de dados ruidosos. Valores positivos e de pequena magnitude para $\hat{\lambda}_1$ podem suscitar dúvidas sobre a existência de características caóticas no sistema original, visto que podem estar associados a questões diversas como precisão numérica na estimativa das matrizes $\underline{\underline{K}}_t$, $\underline{\underline{Q}}_t$ e $\underline{\underline{R}}_t$ e ruído de medição.

Visando contornar estas questões, GENÇAY [170] desenvolveu um método baseado em amostragem para obtenção da distribuição empírica dos expoentes de *Lyapunov*, a qual pode ser utilizada para testar a hipótese de existência de características caóticas. O algoritmo proposto pode ser resumido da maneira que segue:

1. Calcule o espectro de *Lyapunov* utilizando todo o conjunto de dados em conjunto com as equações (3.21) e (3.22).
2. Faça $\hat{\lambda}_{\max} = \hat{\lambda}_1$, o expoente máximo obtido no passo anterior e armazene as matrizes $\underline{\underline{R}}_t$ calculadas em (3.21).
3. Para $l = 1, 2, ..., N$, repita os seguintes passos:
   3.1. Sorteie com reposição $M = N/d$ matrizes $\underline{\underline{R}}_t$.
   3.2. Utilizando somente as matrizes sorteadas, calcule na equação (3.22) o espectro $\hat{\Lambda}_l = \left(\hat{\lambda}_{1l}, \hat{\lambda}_{2l}, \ldots, \hat{\lambda}_{dl}\right)$.
4. Ordene os l valores estimados para o expoente máximo $\hat{\lambda}_{1l}$, gerando a distribuição empírica estimada por amostragem para esta grandeza.

A distribuição empírica obtida através do algoritmo acima pode ser utilizada para realização de inferências sobre o expoente máximo $\hat{\lambda}_{\max}$ obtido utilizando todo o conjunto de dados. Considerando a hipótese nula $H_0 : \hat{\lambda}_1 = \hat{\lambda}_{\max}$, intervalos podem ser gerados para teste contra a hipótese alternativa $H_1 : \hat{\lambda}_1 \neq \hat{\lambda}_{\max}$. Com grau de confiança α, o intervalo $A(\alpha) = \left[\hat{\lambda}_\alpha, \hat{\lambda}_{1-\alpha}\right]$ é definido pelos valores $\hat{\lambda}_\alpha$ e $\hat{\lambda}_{1-\alpha}$ tais que α % dos valores estimados na distribuição empírica estão situados respectivamente à esquerda e à direita de $\hat{\lambda}_\alpha$ e $\hat{\lambda}_{1-\alpha}$. Assim, se $\hat{\lambda}_{\max} \in A(\alpha)$, a hipótese nula não pode ser rejeitada.

O método proposto em [170] fornece ferramentas para identificação de aspectos caóticos através da análise do espectro de expoentes de *Lyapunov*. Assim, o estudo

destas características invariantes do atrator pode ser utilizado em conjunto com o teorema de *Takens* para identificação de sistemas caóticos. Entretanto, estas técnicas permitem analisar somente a dinâmica de sistemas autônomos dados pela equação (3.1). Em análise de séries temporais, variáveis externas, denominadas exógenas, podem influenciar sobremaneira o comportamento do histórico em estudo, indicando a existência de relacionamentos entre os sinais. Esta questão evidencia a necessidade de técnicas para detecção de relacionamentos entre sistemas dinâmicos. Metodologias para esta tarefa podem ser obtidas através do estudo do sincronismo entre sistemas caóticos, conforme será discutido na próxima seção.

3.1.3 Sincronismo entre sistemas caóticos

Diante da sensibilidade a condições iniciais inerente a trajetórias oriundas de sistemas caóticos, a existência de sincronismo entre sistemas deste tipo soa paradoxal. Apesar de possuírem o mesmo atrator no espaço de estados, sistemas caóticos idênticos que evoluem a partir de condições iniciais próximas divergem ao longo do tempo. Apesar desta característica, dinâmicas associadas a sistemas distintos porém acoplados podem estar relacionadas, no que é conhecido como sincronismo entre sistemas caóticos.

O estudo do sincronismo entre sistemas caóticos teve origem no trabalho de PECORA e CARROLL [171], que estudaram sistemas caóticos formados a partir do acoplamento entre dois subsistemas idênticos. Este tipo de sincronismo, conhecido como sincronização idêntica ou convencional, entre sistemas caóticos idênticos é de fácil detecção, visto que corresponde a um colapso da evolução do sistema em torno do hiperplano diagonal no espaço de estados completo [67]. Por outro lado, a detecção de sincronismo entre sistemas caóticos distintos é mais complicada, visto que a obtenção de características invariantes das regiões para as quais as trajetórias convergem não é

trivial. Esta forma mais geral de acoplamento entre sistemas caóticos é conhecida como sincronismo generalizado [66].

Sejam dois sistemas caóticos em tempo discreto $t \in \mathbb{N}$, um autônomo $\underline{X} \in \mathbb{R}^D$ conhecido como guia e outro $\underline{Y} \in \mathbb{R}^R$, denominado guiado, com dinâmicas dadas pelas seguintes equações:

$$\begin{aligned} \underline{X}(t+1) &= F\left[\underline{X}(t)\right] \\ \underline{Y}(t+1) &= U\left[\underline{Y}(t), \underline{X}(t)\right] \end{aligned} \tag{3.24}$$

Em (3.24), $F(\underline{X}): \mathbb{R}^D \to \mathbb{R}^D$ e $U(\underline{Y}, \underline{X}): \mathbb{R}^R \times \mathbb{R}^D \to \mathbb{R}^R$ representam as dinâmicas dos sistemas guia e guiado. Estes sistemas estam em sincronismo generalizado se suas trajetórias ao longo dos respectivos espaços de estado estiverem diretamente relacionadas, ou seja, existe uma função $\varphi(\underline{X}): \mathbb{R}^D \to \mathbb{R}^R$ tal que:

$$\underline{Y}(t) = \varphi\left[\underline{X}(t)\right] \tag{3.25}$$

Visto que as equações que regem as dinâmicas são desconhecidas, juntamente com a função de acoplamento $\varphi(\underline{X})$, são necessários métodos para detecção destas condições a partir de conjuntos de dados oriundos destes sistemas.

RULKOV *et. al.* [66] desenvolveram um método baseado na idéia dos falsos vizinhos mais próximos para detecção de sincronismo. Conhecido como falsos vizinhos mais próximos mútuos, a técnica parte da premissa da existência de uma função suave $\varphi(\underline{X})$, ou seja, continuamente diferenciável. Assim, pontos localizados em uma dada vizinhança do sistema $\underline{X}$ estarão associados a pontos próximos no espaço de estados do sistema guiado $\underline{Y}$.

Seja $\underline{X}\left[n(t,D)\right]$ o vizinho mais próximo de $\underline{X}(t)$. Supondo a validade da equação (3.25) e que a distância entre vizinhos nos respectivos espaços de estados seja pequena, é obtida a seguinte relação aproximada [66]:

$$\underline{Y}(t)-\underline{Y}\left[n(t,D)\right]\approx D\left[\underline{X}(t)\right]\left\{\underline{X}(t)-\underline{X}\left[n(t,D)\right]\right\} \tag{3.26}$$

Em (3.26), $D(\underline{X}):\mathbb{R}^D\to\mathbb{R}^R\times\mathbb{R}^D$ é a matriz jacobiana do mapeamento $\varphi(\underline{X})$. De forma análoga, observando o vizinho mais próximo de $\underline{Y}(t)$ no espaço de estados do sistema guiado, denotado por $\underline{Y}\left[n(t,R)\right]$,

$$\underline{Y}(t)-\underline{Y}\left[n(t,R)\right]\approx D\left[\underline{X}(t)\right]\left\{\underline{X}(t)-\underline{X}\left[n(t,R)\right]\right\} \tag{3.27}$$

As relações dadas em (3.25) e (3.26) permitem definir a medida $M\left[\underline{X}(t),\underline{Y}(t)\right]$ dada por:

$$M\left[\underline{X}(t),\underline{Y}(t)\right]=\frac{\left\|\underline{Y}(t)-\underline{Y}\left[n(t,D)\right]\right\|}{\left\|\underline{X}(t)-\underline{X}\left[n(t,D)\right]\right\|}\frac{\left\|\underline{X}(t)-\underline{X}\left[n(t,R)\right]\right\|}{\left\|\underline{Y}(t)-\underline{Y}\left[n(t,R)\right]\right\|} \tag{3.28}$$

Em (3.28), $\|\underline{Z}\|$ é uma norma definida no respectivo espaço. Desta forma, se o mapeamento suave $\varphi(\underline{X})$ existe, é esperado que a estatística $M\left[\underline{X}(t),\underline{Y}(t)\right]$ seja da ordem da unidade para todos os instantes de tempo t.

Visto que reconstruções do espaço de estados baseadas no teorema de Takens apresentado na seção 3.1.1 são imersões dos respectivos espaços originais, o estudo de $M\left[\underline{X}(t),\underline{Y}(t)\right]$ para os respectivos espaços reconstruídos permanece válido. Porém, o uso direto de (3.28) apresenta alguns empecilhos, visto que além das distâncias serem calculadas em espaços distintos, em dimensões de imersão elevadas a distância entre vizinhos pode assumir valores consideráveis, violando uma das premissas para obtenção

de (3.26) e (3.27). Assim, $M\left[\underline{X}(t),\underline{Y}(t)\right]$ dado por (3.28) pode assumir valores elevados mesmo para sistemas caóticos sincronizados, porém de dimensão elevada.

Para modificação da estatística $M\left[\underline{X}(t),\underline{Y}(t)\right]$, seja a reconstrução $\underline{y}(t)\in\mathbb{R}^r$, dada pela equação (3.3), do sistema guiado $\underline{Y}$. Para o sistema $\underline{X}$, seja a imersão $\underline{x}(t)\in\mathbb{R}^d$ também obtida de acordo com a equação (3.3). Por último, seja a reconstrução auxiliar $\underline{x}'(t)\in\mathbb{R}^{d'}$ do sistema guia, seguindo o teorema de *Takens* porém com dimensão de imersão igual a do sistema guiado, ou seja, $d'=r$. De posse destes espaços, são obtidos os respectivos vizinhos mais próximos em cada espaço reconstruído, no senso das equações (3.9) e (3.10), com $\underline{y}\left[n(t,r)\right]$ respondendo pelo vizinho mais próximo de $\underline{y}(t)$, $\underline{x}\left[n(t,d)\right]$ pelo vizinho de $\underline{x}(t)$ e $\underline{x}'\left[n(t,d')\right]$ pelo vizinho de $\underline{x}'(t)$. Assim, a estatística $m\left[\underline{x}(t),\underline{y}(t)\right]$ conhecida como falsos vizinhos mais próximos mútuos pode ser definida pela seguinte relação [66]:

$$m\left[\underline{x}(t),\underline{y}(t),d,r\right]=\frac{\left\|\underline{x}'(t)-\underline{x}'\left[n(t,d')\right]\right\|}{\left\|\underline{x}'(t)-\underline{x}'\left[n(t,d)\right]\right\|}\frac{\left\|\underline{y}(t)-\underline{y}\left[n(t,d)\right]\right\|}{\left\|\underline{y}(t)-\underline{y}\left[n(t,r)\right]\right\|} \tag{3.29}$$

De forma análoga à análise de $M\left[\underline{X}(t),\underline{Y}(t)\right]$, é esperado que o valor médio da estatística $m\left[\underline{x}(t),\underline{y}(t),d,r\right]$ seja igual à unidade para sistemas caóticos fortemente sincronizados. Para sistemas sem sincronismo, ou seja, o mapeamento $\varphi(\underline{X})$ não existe e a equação (3.25) não é mais válida, são esperados valores elevados para a média de $m\left[\underline{x}(t),\underline{y}(t),d,r\right]$.

O método desenvolvido por RULKOV *et. al.* [66] permite a detecção de sincronismo onde o acoplamento ocorre por meio de um mapeamento $\varphi(\underline{X})$ suave. Entretanto, sistemas caóticos podem estar relacionados de formas mais gerais,

caracterizando outras formas de sincronismo. Segundo PYRAGAS [67], se $\varphi(\underline{X})$ for suave, ou seja, continuamente diferenciável, o sincronismo é classificado como forte. Se o mapeamento contínuo $\varphi(\underline{X})$ existir porém não for suave, o sincronismo entre $\underline{X}$ e $\underline{Y}$ é dito fraco. Assim, o método dos falsos vizinhos mais próximos mútuos identifica somente sincronismo forte, motivando o desenvolvimento de métodos para detecção de instâncias mais gerais de sincronismo.

O tipo de sincronismo pode ser classificado através do estudo dos expoentes de *Lyapunov* condicionais [169]. Sistemas guiados como o representado pela equação (3.24) apresentam espectro de *Lyapunov* contendo $R+D$ expoentes, sendo $\Lambda_D = \left(\lambda_1^D, \lambda_2^D, \ldots, \lambda_D^D\right)$ o espectro relacionado com o sistema guia $\underline{X}$ e $\Lambda_R = \left(\lambda_1^R, \lambda_2^R, \ldots, \lambda_R^R\right)$ os expoentes denominados expoentes de *Lyapunov* condicionais. O espectro global $\Lambda_G = \left(\lambda_1^G, \lambda_2^G, \ldots, \lambda_{R+D}^G\right)$ é formado a partir da ordenação decrescente do conjunto obtido pela junção do espectro do sistema guia Λ_D com o conjunto relacionado com os expoentes condicionais Λ_R. Segundo [67], para existência de sincronismo generalizado, o expoente condicional máximo λ_1^R deve ser negativo. Além disso, se o mapeamento $\varphi(\underline{X})$ não for suave, a dimensão ν_Y do atrator no espaço de estados completo $\mathbb{R}^R \times \mathbb{R}^D$ será maior do que a dimensão ν_X do atrator no espaço do sistema guia. Por outro lado, se $\varphi(\underline{X})$ for continuamente diferenciável, as dimensões serão iguais, ou seja, $\nu_Y = \nu_X$. Utilizando a equação (3.23) em conjunto com os espectros Λ_G e Λ_D, as respectivas dimensões ν_Y e ν_X podem ser estimadas da forma que segue:

$$\nu_Y = l_Y + \frac{1}{\lambda_{l_Y+1}} \sum_{i=1}^{l_Y} \lambda_i^G \tag{3.30}$$

$$\nu_X = l_X + \frac{1}{\lambda_{l_X+1}} \sum_{i=1}^{l_X} \lambda_i^D \tag{3.31}$$

Para que os sistemas sejam fortemente sincronizados, $\lambda_1^R < \lambda_{l_Y+1}^D$. Desta forma, a dimensão do atrator do espaço de estados completo $\mathbb{R}^R \times \mathbb{R}^D$ é independente da trajetória no espaço de estados $\mathbb{R}^R$ do sistema guiado [67].

O espectro de *Lyapunov* do sistema no espaço completo pode ser estimado de forma análoga à apresentada na seção 3.1.2, incluindo os expoentes do sistema guia e os expoentes condicionais. Dadas as respectivas reconstruções $\underline{y}(t) \in \mathbb{R}^r$ e $\underline{x}(t) \in \mathbb{R}^d$, realizadas considerando diferentes atrasos de imersão τ_y e τ_x, o mapeamento no espaço tangente pode ser estimado através de uma aproximação de primeira ordem dada por [169]:

$$\underline{\underline{A}}_t \left\{ \underline{x}(t) - \underline{x}\left[n(t,d) \right] \right\} + \underline{\underline{B}}_t \left\{ \underline{y}(t) - \underline{y}\left[n(t,r) \right] \right\} = \underline{y}(t+m) - \underline{y}\left[n(t,r) + m \right] \tag{3.32}$$

Em (3.32), $\underline{\underline{A}}_t \in \mathbb{R}^r \times \mathbb{R}^d$ e $\underline{\underline{B}}_t \in \mathbb{R}^r \times \mathbb{R}^r$ são matrizes responsável pelo mapeamento linear do espaço tangente. Para $m = \tau_y$, estas matrizes cheias são substituídas pelas matrizes esparsas dadas por:

$$\underline{\underline{A}}_t = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \\ a_1(t) & a_2(t) & \cdots & a_d(t) \end{bmatrix}; \; \underline{\underline{B}}_t = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ b_1(t) & b_2(t) & b_3(t) & \cdots & b_r(t) \end{bmatrix} \tag{3.33}$$

Os $d + r$ parâmetros que definem as matrizes dadas em (3.33) podem ser estimados por mínimos quadrados. De forma análoga à discutida na seção 3.1.2, um número mínimo

N_v de vizinhos deve ser selecionado para tornar a estimativa robusta. Seguindo a recomendação de [169], $N_v = 2\left(d + r\right)$.

O método dos falsos vizinhos mais próximos mútuos e a análise dos expoentes de *Lyapunov* condicionais permitem detectar formas gerais de sincronismo entre sistemas caóticos. Em outras palavras, junto com o teorema de *Takens* estes índices fornecem ferramentas para identificação não-linear de sistemas, no contexto da seleção de entradas e detecção de relacionamentos entre séries temporais reconstruídas. Por outro lado, esta teoria detecta somente a existência de sincronismo entre os sistemas, sendo necessários, para fins de previsão, métodos para estimação do mapeamento $U\left(\underline{Y}, \underline{X}\right): \mathbb{R}^R \times \mathbb{R}^D \to \mathbb{R}^R$ dado em (3.24), conforme será discutido na próxima seção.

3.1.4 Previsão

De posse das imersões $\underline{x}\left(t\right) \in \mathbb{R}^d$ e $\underline{y}\left(t\right) \in \mathbb{R}^r$ dos sistemas em estudo e identificada existência de sincronismo entre eles, para fins de previsão são necessárias técnicas para estimação dos mapeamentos que regem as dinâmicas destes sistemas. Matematicamente, supondo que a série a ser prevista seja representada pelo sistema $\underline{y}\left(t\right)$, deve ser estimado o mapeamento $u\left(\underline{y}, \underline{x}\right): \mathbb{R}^r \times \mathbb{R}^d \to \mathbb{R}^r$ dado por:

$$\underline{y}\left(t+1\right) = u\left[\underline{y}\left(t\right), \underline{x}\left(t+1\right)\right] \tag{3.34}$$

Visto que o objetivo reside na modelagem da série $y\left(t\right) \in \mathbb{R}$, a estimação do mapeamento completo $u\left(\underline{y}, \underline{x}\right)$ não é necessária. Para fins de previsão, deve ser estimada somente a função $f\left(\underline{y}, \underline{x}\right): \mathbb{R}^r \times \mathbb{R}^d \to \mathbb{R}$ que relaciona pontos no espaço reconstruído com valores futuros da série em estudo, dada por:

$$y\left(t+1\right) = f\left[\underline{y}\left(t\right), \underline{x}\left(t+1\right)\right] + \varepsilon_t \tag{3.35}$$

Em (3.35), ε_t representa o resíduo de modelagem.

As técnicas para estimação de $f\left(\underline{y},\underline{x}\right)$ podem ser divididas em dois grupos. Os métodos globais buscam estimar funções $f\left(\underline{y},\underline{x}\right)$ que modelam o atrator como um todo, utilizando diretamente todo o conjunto de dados. Por outro lado, as técnicas conhecidas como locais utilizam agrupamentos da base de dados visando à estimação independente de aspectos específicos do atrator.

Para utilização de modelos locais, são necessárias técnicas para seleção dos pontos relacionados à região do atrator envolvida na modelagem. Especificamente, para previsão a partir do instante de tempo t, devem ser definidos os pares $\left[\underline{y}^t\left(k\right),\underline{x}^t\left(k\right)\right]^t$ integrantes da mesma região do atrator na qual o ponto $\left[\underline{y}^t\left(t\right),\underline{x}^t\left(t\right)\right]^t$ está localizado. Esta tarefa pode ser efetuada através da seleção dos n vizinhos mais próximos de $\left[\underline{y}^t\left(t\right),\underline{x}^t\left(t\right)\right]^t$ ou por meio do agrupamento de todo conjunto de dados, sendo definidas as K diferentes regiões do atrator a serem modeladas. Estes métodos requerem a especificação de alguns parâmetros, a saber, o número n de vizinhos ou a quantidade K de regiões na qual o atrator deve ser dividido, inviabilizando a automatização do processo de modelagem como um todo. De fato, muitas aplicações de previsão de carga encontradas na literatura utilizam partições definidas de forma heurística, tendo por base o conhecimento de especialistas acerca do histórico em estudo. Esta abordagem será utilizada em duas das três bases de dados analisadas nesta tese.

A escolha entre modelos locais e globais é dependente do problema, não existindo uma recomendação geral. Ao modelarem de forma isolada e independente características específicas de um dado mapeamento, estratégias locais podem apresentar vantagens quando aplicadas a sistemas apresentando diferentes regimes ou pontos de

operação. Entretanto, a definição empírica de tais regimes e das regiões de transição entre eles não é trivial, requisitando a intervenção de especialistas e comprometendo o nível de automatização do processo.

Definida a estratégia para modelagem do atrator, estruturas para a função $f\left(\underline{y},\underline{x}\right)$ devem ser especificadas. Neste trabalho, são desenvolvidos modelos neurais *feedforward*, os quais serão apresentados detalhadamente nos capítulos 4 e 5.

3.1.5 Método automático para seleção de entradas

De uma forma geral, as técnicas apresentadas nesta seção podem ser resumidas em um algoritmo para seleção de entradas de modelos de sistemas caóticos o qual é apresentado a seguir:

1. Dado um conjunto de séries temporais, defina a série a ser prevista $y(k) \in \mathbb{R}$, $k = 1,2,...,N$, e as séries exógenas $x_i(k) \in \mathbb{R}$, $k = 1,2,...,N$, $i = 1,2,...,S$, onde N responde pelo número de dados e S pelo número de séries disponíveis.
2. Utilizando o teorema de *Takens* e as técnicas apresentadas na seção 3.1.1, determine os parâmetros d_y e τ_y da imersão da série alvo no espaço aumentado $\underline{y}(k) \in \mathbb{R}^{d_y}$, dada pela equação (3.3) com $k = \left(d_y - 1\right)\tau_y + 1, \left(d_y - 1\right)\tau_y + 2, ..., N$.
3. Verifique a existência de traços caóticos em $\underline{y}(k) \in \mathbb{R}^{d_y}$ através da análise dos expoentes de *Lyapunov*, conforme apresentado na seção 3.1.2. Existindo características caóticas, vá para o próximo passo. Do contrário, encerre o algoritmo e utilize alguma metodologia para estudo de sistemas não-caóticos.
4. Para cada série exógena $x_i(k) \in \mathbb{R}$, ou seja, para $i = 1,2,...,S$, execute os seguintes passos:

2.1. Utilizando o teorema de *Takens* e as técnicas apresentadas na seção 3.1.1, determine os parâmetros d_{x_i} e τ_{x_i} da imersão da série exógena no espaço aumentado $\underline{x}_i(k) \in \mathbb{R}^{d_{x_i}}$, dada pela equação (3.3) com $k = \left(d_{x_i} - 1\right)\tau_{x_i} + 1, \left(d_{x_i} - 1\right)\tau_{x_i} + 2, ..., N$.

2.2. Verifique a existência de sincronismo utilizando a estatística dos falsos vizinhos mais próximos mútuos, dada pela equação (3.29), e classifique o mesmo através da análise dos expoentes de *Lyapunov* condicionais, calculados utilizando os métodos apresentados na seção 3.1.3.

2.3. Na existência de sincronismo fraco ou forte, armazene a imersão $\underline{x}_i(k) \in \mathbb{R}^{d_{x_i}}$ no conjunto de entradas. Do contrário, descarte essa série e retorne ao passo 2.1 para estudo da próxima série exógena.

5. Inclua no conjunto de entradas exógenas a imersão $\underline{y}(k) \in \mathbb{R}^{d_y}$ da série alvo.

6. Se o modelo de previsão for global, apresente o conjunto de entradas completo ao mesmo e realize previsões. Do contrário, particione o atrator reconstruído e obtenha os modelos locais de previsão.

No algoritmo listado acima, vale ressaltar a necessidade de existência de traços caóticos para utilização das técnicas em virtude dos pressupostos incluídos no desenvolvimento destas metodologias. Para modelagem e previsão de sistemas não-caóticos existem diversas metodologias consolidadas na literatura, tais como análise clássica de séries temporais.

3.2 Definição empírica de limiares de relevância

De posse do conjunto inicial de entradas, métodos de filtragem ou encapsulados podem ser utilizados para processamento deste conjunto, visando redução de dimensionalidade através da retirada de variáveis irrelevantes e/ou redundantes ou

melhoria direta do desempenho do modelo. Mais interessantes sob o ponto de previsão, as técnicas encapsuladas necessitam da definição de limiares a partir dos quais as variáveis podem ser consideradas desnecessárias para o cálculo das previsões, podendo assim ser descartadas.

A questão do descarte de variáveis pode ser abordada de duas maneiras. A primeira reside na especificação do número de entradas a serem selecionadas. Visto que em aplicações práticas o número de entradas relevantes não é uma informação disponível *a priori*, esta opção não é recomendável, já que esta escolha arbitrária pode levar à inclusão de variáveis irrelevantes ou à exclusão de sinais relevantes para o cálculo final das previsões.

A definição de limiares de relevância é outra forma de tratar este problema. Através da definição de um patamar σ_0, entradas com índice de relevância menor que este limiar são consideradas irrelevantes podendo ser excluídas do modelo final. Ao contrário de índices de correlação linear, cujos patamares de relevância podem ser definidos através de teste de hipótese conforme mostrado na seção 2.1.1, a definição de limiares analíticos para modelos não-lineares não é trivial, sendo necessária a intervenção de especialistas para definição heurística de tais patamares, tornando a abordagem extremamente dependente do problema.

Para contornar a questão de definição de limiares por parte de especialistas, STOPPIGLIA *et. al.* [68] introduziram o uso de variáveis aleatórias de prova para seleção de entradas. Na técnica de filtragem utilizada pelos autores, baseada em ortogonalização de *Gram-Schmidt*, é necessária a definição de um critério de parada visando à escolha das entradas mais relevantes. Utilizando implicitamente variáveis obtidas segundo uma distribuição normal e por construção descorrelacionadas com a saída, os autores divisam uma técnica baseada na probabilidade de escolher uma entrada

menos relacionada com a saída do que a variável de prova para seleção de entradas de modelos lineares. Através de um teste de hipótese, o método busca selecionar somente as variáveis que apresentem relação superior com a saída em relação à variável de prova.

Apesar de proposto para modelos lineares, a essência do método apresentado em [68] pode ser utilizada para definição do limiar de relevância σ_0. Analogamente, buscando determinar as entradas que apresentem relação superior com a saída quando comparadas à variável de prova, a inserção explícita desta última ao conjunto de treinamento pode ser utilizada para especificação de σ_0. Especificamente, utilizando o conjunto de treinamento estendido criado a partir da incorporação do sinal de prova ao conjunto de entradas original, um método encapsulado é aplicado para solução do problema dado em (2.20), com $\underline{\nu} \in \{0,1\}^n$ sendo substituído por $\underline{\sigma} \in \mathbb{R}^n$. Ao final do processo, o patamar de relevância σ_0 é determinado através do hiperparâmetro σ_i associado à variável de prova sabidamente não relacionada com a saída. As entradas do conjunto estendido são então ordenadas de forma decrescente segundo a magnitude dos hiperparâmetros $\underline{\sigma}$ estimados, sendo descartadas aquelas situadas abaixo da variável de prova, ou seja, apresentando σ_i menor que σ_0. Neste contexto, são descartadas as variáveis que contribuem menos para o cálculo da saída do que o sinal aleatório de prova, guardando analogia com [68].

A inserção de variáveis de prova deve seguir a natureza das entradas utilizadas. Em problemas que apresentem somente entradas contínuas, uma única variável de prova deste tipo deve ser inserida, sendo esta utilizada para mensuração da relevância. Por outro lado, para casos onde o conjunto de entradas inclui sinais contínuos e discretos, a utilização de uma única variável de prova não é recomendada, diante da natureza

distinta das diversas entradas. Em situações deste tipo, devem ser inseridas duas variáveis de prova, uma contínua e outra discreta. Desta forma, para entradas contínuas, é usado o patamar de relevância σ_C relacionado com o sinal de prova desta natureza, com o limiar de relevância σ_D, associado com o outro sinal de prova, sendo utilizado para variáveis discretas. Neste contexto, as variáveis contínuas e discretas são ordenadas em conjuntos separados, sendo descartadas aquelas situadas abaixo dos respectivos patamares de relevância σ_C e σ_D, respectivamente.

A geração das entradas de prova aleatórias deve ser feita seguindo as características dos sinais de entrada disponíveis. Em virtude da utilização de funções sigmoidais nos neurônios da camada oculta de MLPs, a normalização das entradas e saídas deste tipo de modelo é prática comum. Supondo que as entradas contínuas sejam normalizadas no intervalo $[a,b] \in \mathbb{R}$, o sinal de prova x_{PC} desta natureza é gerado a partir de uma distribuição uniforme $p(x_{PC})$ definida neste mesmo intervalo, dada por:

$$p(x_{PC}) = \begin{cases} \dfrac{1}{b-a}, \text{ se } a \le x_{PC} < b \\ 0, \text{ se } x_{PC} < a \text{ ou } x_{PC} > b \end{cases} \tag{3.36}$$

Outro tipo de normalização comumente utilizado diz respeito à padronização dos sinais, ou seja, tornar a média nula e a variância unitária. Visando obter variáveis de prova com as mesmas características, o intervalo $[a,b]$ definido acima deve ser igual a $\left[-\sqrt{3},\sqrt{3}\right]$, visto que sinais distribuídos uniformemente neste intervalo também são padronizados.

Para as entradas discretas apresentando um conjunto de eventos $C = \{x \in \mathbb{N} \mid x = 0,1,...,k\}$ com k possibilidades de ocorrência, a variável de prova discreta x_{PD} é gerada a partir de uma distribuição uniforme $p(x_{PD})$ definida no mesmo conjunto de ocorrências e dada por:

$$p\left(x_{PD}\right)=\begin{cases}\dfrac{1}{k},\ \text{se } x_{PC}\in C\\ 0,\ \text{se } x_{PC}\notin C\end{cases} \tag{3.37}$$

O método de inserção de variáveis de prova aleatórias descrito acima fornece subsídios para definição empírica de limiares de relevância nos métodos encapsulados desenvolvidos nesta tese. Apesar de ainda não ter sido testada, esta idéia pode também ser utilizada para definição de patamares no método de filtragem baseado na análise da informação mútua entre variáveis apresentado na seção 2.1.1.

3.3 Resumo e discussão

A teoria do caos, no contexto do teorema de *Takens* em conjunto com o conceito de sincronismo generalizado, apresenta mecanismos para identificação não-linear de sistemas caóticos. Apesar de desenvolvida para estudo de sistemas dinâmicos, os métodos desenvolvidos nesta teoria podem ser aplicados a conjuntos de dados reais contaminados por ruído, com este componente estocástico sendo considerado como uma pequena contaminação de um processo essencialmente determinístico [153]. Desta forma, as ferramentas apresentadas neste capítulo podem ser utilizadas para definição do conjunto inicial de entradas de previsores neurais, desde que as séries em estudo sejam caracterizadas como oriundas de sistemas caóticos multivariados.

Conforme destacado na seção 2.1, os métodos encapsulados de seleção de entradas são mais recomendados para fins de previsão, visto que buscam o conjunto mais adequado de variáveis para o modelo em questão. Para evitar a explosão combinatorial resultante da solução exaustiva do problema dado na equação (2.20), a inclusão de ponderadores reais das entradas traz consigo a necessidade da definição de limiares de relevância a partir dos quais os sinais podem ser descartados. A definição empírica de tais patamares compromete a automação da técnica, motivando o desenvolvimento do método apresentado na seção 3.2. Através da inserção de sinais

aleatórios de prova de natureza similar a das entradas utilizadas, esta técnica permite a obtenção empírica de limiares de relevância, reduzindo a intervenção de especialistas na modelagem.

As técnicas apresentadas neste capítulo fornecem meios para obtenção do conjunto inicial de entradas de modelos neurais para previsão de carga a partir somente das séries temporais em estudo. Para detecção da relevância de cada variável, o método das variáveis de prova fornece limiares empíricos. Assim, resta definir os modelos de previsão como também os métodos encapsulados utilizados para estimação da relevância de cada sinal de entrada. Estes tópicos serão assunto dos próximos capítulos, começando pela descrição da inferência *bayesiana* aplicada ao desenvolvimento de MLPs.

4 Inferência *bayesiana* aplicada ao desenvolvimento de MLPs

A aplicação de inferência *bayesiana* ao desenvolvimento de MLPs foi proposta originalmente por *David J.C. Mackay* em 1992 [54]. As principais vantagens desta abordagem são as seguintes [50]:

- O algoritmo de retropropagação do erro tradicional pode ser visto como um caso particular dos resultados obtidos através da aplicação de técnicas de inferência *bayesianas*.
- A teoria da regularização apresenta uma interpretação natural dentro desta abordagem. Um dos motivos reside na obrigatoriedade de inserção de algum conhecimento prévio sobre o problema para obtenção da solução, característica marcante tanto das técnicas de inferência *bayesianas* quanto da teoria de regularização de *Tikhonov*.
- Para problemas de regressão, intervalos de confiança podem ser gerados automaticamente.
- Este método fornece uma estimativa automática do parâmetro de regularização λ, o qual é atualizado ao longo do algoritmo de treinamento, sem a necessidade de técnicas de re-amostragem ou de qualificação analítica de modelos para estimativa deste parâmetro.
- Este procedimento permite o desenvolvimento de um algoritmo de determinação automática de relevância das entradas, do inglês *automatic relevance determination* (ARD), técnica que pode ser utilizada para seleção de variáveis de entrada de modelos neurais.
- Através do cálculo da evidência de cada modelo, relacionada com a probabilidade *a posteriori* de cada estrutura, esta metodologia permite a

comparação entre diferentes modelos utilizando somente os dados disponíveis para treinamento.

- Técnicas de inferência *bayesiana* permitem afirmar em qual região do espaço de entrada devem ser obtidos novos dados com o intuito de aumentar a informação contida no modelo, característica conhecida como aprendizado ativo.

Visto que esta tese utilizará o treinamento *bayesiano* para problemas de aproximação funcional, utilizando MLPs com uma única camada escondida e uma única saída linear, a discussão sobre este assunto estará restrita à apresentação deste algoritmo para este tipo de modelo. Maiores detalhes sobre a aplicação de técnicas *bayesianas* ao treinamento de MLPs podem ser encontrados em [50], [54] e [172].

Com base na maximização da evidência, os três níveis hierárquicos de inferência são explorados, desde a estimação dos parâmetros até a escolha do modelo mais provável à luz dos dados, passando pelo cálculo dos hiperparâmetros cuja análise permite o desenvolvimento do método encapsulado de seleção de entradas. Para facilitar a compreensão, cada um dos três níveis hierárquicos serão apresentados a seguir.

4.1 Treinamento *bayesiano* de MLPs

Definida a estrutura a ser utilizada, ou seja, número de camadas ocultas, número de neurônios por camada e tipo de função de ativação de cada neurônio, dado o conjunto $D=\{X,Y\}$, $X=\{\underline{x}_1,...,\underline{x}_N\}$, Y=$\{d_1,...,d_N\}$, $\underline{x}\in\mathbb{R}^n$, $\underline{x}=[x_1,...,x_n]^t$, $d\in\mathbb{R}$, $d=F(\underline{x})+\zeta$, o objetivo do treinamento do modelo, sob o ponto de vista da inferência *bayesiana*, reside na determinação do vetor de parâmetros $\underline{w}\in\mathbb{R}^M$ que maximize a probabilidade *a posteriori* $p(\underline{w}|Y,X)$, dada por:

$$p(\underline{w}|Y,X)=\frac{p(Y|\underline{w},X)\,p(\underline{w}|X)}{p(Y|X)} \tag{4.1}$$

Na equação (4.1), $p(Y|X) = \int p(Y|\underline{w}, X)\, p(\underline{w}|X)\, d\underline{w}$ é um fator de normalização, que garante que $\int p(\underline{w}|Y, X)\, d\underline{w} = 1$. Visto que MLPs não modelam a distribuição de probabilidade $p(\underline{x})$ geradora dos padrões de entrada e o conjunto X aparece como variável condicionante em todas as probabilidades envolvidas na equação (4.1), este conjunto será omitido da notação a partir deste ponto.

Portanto, para o cálculo da probabilidade a *posteriori* $p(\underline{w}|Y)$ do vetor $\underline{w}$, é necessário o conhecimento da distribuição de probabilidade *a priori* $p(\underline{w})$ deste vetor, como também a sua função de verossimilhança $p(Y|\underline{w})$, a qual está relacionada com a distribuição de probabilidade do ruído existente na saída desejada. Na ausência de conhecimento prévio sobre a solução, conforme é o caso do treinamento de MLPs, a escolha da distribuição $p(\underline{w})$ deve refletir tal falta de conhecimento. Visto que modelos apresentando componentes de $\underline{w}$ com pequena magnitude reproduzem mapeamentos suaves [50], uma escolha razoável para a distribuição $p(\underline{w})$ reside na distribuição *gaussiana* com vetor média nulo e matriz de covariância $\alpha^{-1}\underline{\underline{I}}$, $\underline{\underline{I}} \in \mathbb{R}^M \times \mathbb{R}^M$ igual à matriz identidade, dada por:

$$p(\underline{w}) = \frac{1}{Z_{\underline{W}}(\alpha)} e^{-\left(\frac{\alpha}{2}\|\underline{w}\|^2\right)} \tag{4.2}$$

$$Z_{\underline{W}}(\alpha) = \left(\frac{2\pi}{\alpha}\right)^{\frac{M}{2}}$$

Na equação (4.2), $\alpha \in \mathbb{R}^+$ é o chamado hiperparâmetro, cuja estimativa será apresentada ao longo desta seção, e que, para este estágio da apresentação do algoritmo, é admitido como uma constante de valor conhecido.

Além do pressuposto de reprodução de mapeamentos suaves, a escolha da distribuição de probabilidade *a priori* $p(\underline{w})$ dada pela equação (4.2) simplifica as análises subseqüentes, dando origem a um algoritmo analítico e iterativo para o cálculo de $\underline{w}$. Distribuições não-informativas, refletindo completo desconhecimento sobre o comportamento de $\underline{w}$, também podem ser utilizadas [173]. Neste contexto, a busca pelo valor mais provável de $\underline{w}$ à luz dos dados é substituída pela estimativa da distribuição de probabilidade *a posteriori* $p(d_{N+1}|\underline{d}_{N+1}, Y, X)$ da saída a ser prevista d_{N+1} dado o novo padrão de entrada $\underline{x}_{N+1}$ e o conjunto de dados $D = \{X, Y\}$. Esta distribuição é obtida pela marginalização de $p(d_{N+1}|\underline{x}_{N+1}, \underline{\theta})$ sobre todo o espaço $\underline{\theta}$ de parâmetros desconhecidos (incluindo $\underline{w}$), através da integral dada por:

$$p(d_{N+1}|\underline{x}_{N+1}, Y, X) = \int p(d_{N+1}|\underline{x}_{N+1}, \underline{\theta}) p(\underline{\theta}|Y, X) d\underline{\theta} \tag{4.3}$$

Como o cálculo analítico da equação (4.3) é impraticável para quaisquer $p(d_{N+1}|\underline{x}_{N+1}, \underline{\theta})$ e $p(\underline{\theta}|Y, X)$, a probabilidade $p(d_{N+1}|\underline{x}_{N+1}, Y, X)$ é obtida utilizando métodos de integração numérica baseados em simulações de Monte Carlo e modelos híbridos de Markov [174]. Além de computacionalmente dispendiosa, ao não fornecer estimativas pontuais para os hiperparâmetros esta abordagem inviabiliza o método de seleção de entradas proposto, sendo por isso desconsiderada neste trabalho.

Apesar de simplificar o desenvolvimento, a escolha de $p(\underline{w})$ na equação (4.2) não parece adequada. Diferentes conjuntos de pesos devem apresentar comportamentos distintos. Pesos que ligam entradas de natureza diversa possuem características diferentes, motivando a utilização de probabilidades a *priori* independentes para cada conjunto de parâmetros. Uma escolha de $p(\underline{w})$ segundo um agrupamento específico de pesos dá origem ao método de determinação automática de relevância, que será

apresentado em seguida. Porém, para fins de apresentação da teoria, neste ponto será admitida a probabilidade a priori $p(\underline{w})$ dada pela equação (4.2).

Definida a distribuição $p(\underline{w})$, resta agora especificar a distribuição de probabilidade do ruído ζ existente na saída desejada. Supondo que a função a ser aproximada $F(\underline{x})$ apresente certo grau de suavidade, e que o ruído ζ possui distribuição *gaussiana* com média nula e variância β^{-1}, a probabilidade da ocorrência de uma saída específica d_k, dado o vetor de entrada $\underline{x}_k$ e o vetor de parâmetros $\underline{w}$, é dada pela seguinte expressão:

$$p\left(d_k \middle| \underline{x}_k, \underline{w}\right) = \frac{e^{\left\{-\frac{\beta}{2}\left[d_k - f\left(\underline{x}_k, \underline{w}\right)\right]^2\right\}}}{\int e^{\left\{-\frac{\beta}{2}\left[d_k - f\left(\underline{x}_k, \underline{w}\right)\right]^2\right\}} dd_k} \tag{4.4}$$

Partindo do pressuposto que os padrões do conjunto de treinamento foram obtidos de maneira independente a partir desta distribuição, podemos obter uma expressão para a verossimilhança $p\left(Y \middle| \underline{w}\right)$, dada pela equação:

$$p\left(Y \middle| \underline{w}\right) = \frac{1}{Z_Y(\beta)} e^{\left\{-\frac{\beta}{2}\sum_{k=1}^{N}\left[d_k - f\left(\underline{x}_k, \underline{w}\right)\right]^2\right\}} \tag{4.5}$$

$$Z_Y(\beta) = \left(\frac{2\pi}{\beta}\right)^{\frac{N}{2}}$$

Em (4.5), $\beta \in \mathbb{R}^+$ é outro hiperparâmetro, cuja estimativa será apresentada ao longo desta seção. Da mesma forma que α, para este estágio da discussão é assumido que este parâmetro apresente um valor conhecido.

De posse das expressões (4.2) e (4.5), é possível calcular a probabilidade *a posteriori* de $\underline{w}$ dado o conjunto de saídas desejadas Y, através da aplicação da regra de *Bayes* descrita na equação (4.1), resultando na seguinte expressão:

$$p\left(\underline{w}\middle|Y\right)=\frac{1}{Z_S}e^{\left[-S(\underline{w})\right]} \tag{4.6}$$

$$Z_s=\int e^{-S(\underline{w})}d\underline{w}$$

$$S\left(\underline{w}\right)=\frac{\beta}{2}\sum_{k=1}^{N}\left[d_k-f\left(\underline{x}_k,\underline{w}\right)\right]^2+\frac{\alpha}{2}\sum_{j=1}^{M}w_j^2$$

O funcional $S\left(\underline{w}\right)$ apresenta duas parcelas. A primeira, a menos de um fator de escala proporcional ao número de padrões, é dada pelo risco empírico apresentado na equação (2.22), representando assim o ajuste do modelo aos dados disponíveis. A outra parcela, relacionada com a inserção de conhecimento prévio através da probabilidade *a priori* $p\left(\underline{w}\right)$, na teoria da regularização representa um funcional regularizador $E_C\left[f\left(\underline{x},\underline{w}\right)\right]$ conhecido como decaimento dos pesos, do inglês *weight decay*. A utilização deste funcional favorece modelos apresentando componentes do vetor $\underline{w}$ com pequena magnitude, buscando gerar mapeamentos suaves [50]. Desta forma, maximizar a probabilidade *a posteriori* de $\underline{w}$ é equivalente à aplicação do regularizador de decaimento de pesos. Além disso, visto que a busca é pelo ponto ótimo $\underline{w}^*$ que minimiza $S\left(\underline{w}\right)$, da equação (2.25), $\lambda=\alpha/\beta$. Portanto, o cálculo dos hiperparâmetros α e β fornece uma estimativa do parâmetro de regularização λ.

Supondo que os hiperparâmetros α e β eram conhecidos, a aplicação de inferência *bayesiana* foi limitada até aqui à estimativa do vetor de parâmetros $\underline{w}$. Entretanto, sabemos que estes valores são desconhecidos *a priori*, sendo necessária uma estimativa para estas variáveis. Portanto, visto que $\underline{w}$, α e β são desconhecidos, a probabilidade *a posteriori* de $\underline{w}$, $p\left(\underline{w}\middle|Y\right)$ passa a ser dada por:

$$p\left(\underline{w}\middle|Y\right)=\iint p\left(\underline{w},\alpha,\beta\middle|Y\right)d\alpha d\beta=\iint p\left(\underline{w}\middle|\alpha,\beta,Y\right)p\left(\alpha,\beta\middle|Y\right)d\alpha d\beta \tag{4.7}$$

Visto que $\underline{w}$ não é a única variável desconhecida, a probabilidade *a posteriori* $p\left(\underline{w}|Y\right)$ deve ser obtida através da integração da probabilidade *a posteriori* de todas as variáveis desconhecidas $p\left(\underline{w},\alpha,\beta|Y\right)$ sobre todo o espaço de hiperparâmetros. A partir da equação (4.7), existem duas abordagens para a estimativa de α e β. Uma utiliza a integração analítica sobre os hiperparâmetros, abordagem que foge do escopo desta tese. A segunda abordagem, conhecida como aproximação da evidência [50], proposta por *Mackay* [54], será utilizada. As duas abordagens levam a resultados semelhantes, conforme pode ser verificado em [50]. Uma breve discussão sobre estes dois procedimentos pode ser encontrada na mesma referência.

A abordagem proposta por *Mackay* [54] parte do pressuposto que a probabilidade *a posteriori* $p\left(\alpha,\beta|Y\right)$ apresenta pouca dispersão em torno dos valores mais prováveis α^* e β^*, permitindo a seguinte simplificação da equação (4.7):

$$p\left(\underline{w}|Y\right) \simeq p\left(\underline{w}|\alpha^*,\beta^*,Y\right)\iint p\left(\alpha,\beta|Y\right)d\alpha d\beta = p\left(\underline{w}|\alpha^*,\beta^*,Y\right) \tag{4.8}$$

Portanto, para determinação dos valores mais prováveis para os hiperparâmetros, a probabilidade *a posteriori* destes, $p\left(\alpha,\beta|Y\right)$, deve ser maximizada. Aplicando a regra de *Bayes*, é obtida a seguinte expressão para esta probabilidade:

$$p\left(\alpha,\beta|Y\right) = \frac{p\left(Y|\alpha,\beta\right)p\left(\alpha,\beta\right)}{p\left(Y\right)} \tag{4.9}$$

Da equação (4.9), é visto que algum conhecimento prévio sobre α e β deve ser inserido, na forma da distribuição de probabilidade $p\left(\alpha,\beta\right)$. Visto que pouco, ou mesmo nenhum conhecimento sobre os hiperparâmetros é disponível, a única informação prévia que pode ser inserida diz respeito a esta ausência de conhecimento. Portanto, a probabilidade $p\left(\alpha,\beta\right)$ deve ser escolhida de tal forma que esta distribuição

seja insensível a valores específicos de α e β. Logo, visto que a probabilidade $p(Y)$ é independente dos hiperparâmetros, a maximização da probabilidade *a posteriori* $p(\alpha,\beta|Y)$ é obtida através da maximização da probabilidade $p(Y|\alpha,\beta)$, também conhecida como evidência para os hiperparâmetros [50]. Esta probabilidade pode ser obtida através da seguinte expressão:

$$p(Y|\alpha,\beta)=\int p(Y|\underline{w},\alpha,\beta)\,p(\underline{w}|\alpha,\beta)\,d\underline{w} \tag{4.10}$$

Visto que α está relacionado somente com a probabilidade a *priori* do vetor $\underline{w}$, e que β está associado apenas com a distribuição do ruído aditivo ζ existente na saída desejada, a equação (4.10) passa a ser dada por:

$$p(Y|\alpha,\beta)=\int p(Y|\underline{w},\beta)\,p(\underline{w}|\alpha)\,d\underline{w} \tag{4.11}$$

Portanto, utilizando as equações (4.2) e (4.5) na equação (4.11), a seguinte expressão é obtida:

$$p(Y|\alpha,\beta)=\frac{Z_s(\alpha,\beta)}{Z_Y(\beta)\,Z_{\underline{W}}(\alpha)} \tag{4.12}$$
$$Z_s(\alpha,\beta)=\int e^{\left[-S(\underline{w})\right]}d\underline{w}$$

Na equação (4.12), $S(\underline{w})$ é dado pela equação (4.6). Considerando uma aproximação *gaussiana* da distribuição de probabilidade *a posteriori* de $\underline{w}$, $p(\underline{w}|Y)$, o que equivale à aproximação quadrática em séries de *Taylor* do funcional $S(\underline{w})$ em torno do ponto $\underline{w}^*$, o funcional $Z_s(\alpha,\beta)$ passa a ser dado por:

$$Z_s(\alpha,\beta)=e^{-S(\underline{w}^*)}(2\pi)^{\frac{M}{2}}\left\{\det\left[\underline{\underline{A}}(\underline{w})\big|_{\underline{w}=\underline{w}^*}\right]\right\}^{-\frac{1}{2}} \tag{4.13}$$
$$\underline{\underline{A}}(\underline{w})\big|_{\underline{w}=\underline{w}^*}=\beta\,\underline{\underline{H}}(\underline{w})\big|_{\underline{w}=\underline{w}^*}+\alpha\,\underline{\underline{I}}$$

Na equação (4.13), $\underline{w}^*$ é o vetor de parâmetros que minimiza o funcional $S(\underline{w})$, $\underline{\underline{H}}(\underline{w})\big|_{\underline{w}=\underline{w}^*}$ a matriz *hessiana* do funcional $E_S(\underline{w})$ calculada no ponto $\underline{w}^*$ e $\underline{\underline{I}} \in \mathbb{R}^M \times \mathbb{R}^M$ a matriz identidade. Utilizando esta equação em conjunto com as expressões obtidas para $Z_{\underline{W}}(\alpha)$ e $Z_Y(\beta)$, dadas pelas expressões (4.2) e (4.5), respectivamente, o logaritmo *neperiano* da expressão (4.12) é dado por:

$$\ln p(Y|\alpha,\beta) = -\frac{\alpha}{2}\sum_{j=1}^{M}\left(w_j^*\right)^2 - \frac{\beta}{2}\sum_{k=1}^{N}\left[d_k - f\left(\underline{x}_k, \underline{w}^*\right)\right]^2 - \frac{1}{2}\ln\left\{\det\left[\underline{\underline{A}}(\underline{w})\big|_{\underline{w}=\underline{w}^*}\right]\right\} + \frac{M}{2}\ln\alpha + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) \tag{4.14}$$

A maximização, em relação à α do logaritmo natural da evidência para α e β, dado pela equação (4.14), resulta na seguinte expressão:

$$\gamma = \alpha\sum_{j=1}^{M}\left(w_j^*\right)^2 = M - trace\left\{\left[\underline{\underline{A}}(\underline{w})\big|_{\underline{w}=\underline{w}^*}\right]^{-1}\right\} \tag{4.15}$$

Na equação (4.15), γ é o chamado número efetivo de parâmetros. Com o intuito de evitar a inversão da matriz $\underline{\underline{A}}(\underline{w})\big|_{\underline{w}=\underline{w}^*}$, seja o conjunto de M autovalores da matriz *hessiana* $\underline{\underline{H}}(\underline{w})\big|_{\underline{w}=\underline{w}^*}$, dado por $\lambda = \{\nu_1, \nu_2, ..., \nu_M\}$. Desta forma, a equação (4.15) passa a ser dada por:

$$\gamma = \sum_{i=1}^{M}\frac{\nu_i}{\nu_i + \alpha} \tag{4.16}$$

A maximização do logaritmo natural da evidência para os hiperparâmetros α e β, dado pela equação (4.14), em relação à β resulta na seguinte equação:

$$\beta\sum_{k=1}^{N}\left[d_k - f\left(\underline{x}_k, \underline{w}^*\right)\right]^2 = N - \gamma \tag{4.17}$$

As expressões (4.15) e (4.17) foram obtidas a partir da aproximação quadrática do funcional $S(\underline{w})$ em torno do seu ponto de mínimo $\underline{w}^*$. Métodos de segunda ordem

de treinamento de MLPs, como *Levenberg-Marquardt* [50], utilizam a cada iteração uma aproximação quadrática do funcional de risco empírico em torno do ponto de operação $\underline{w}(l)$. Guardando analogia com estes métodos, a aproximação quadrática do funcional $S(\underline{w})$ pode também ser utilizada em um algoritmo iterativo, dando origem às seguintes equações recursivas para estimativa dos hiperparâmetros α e β:

$$\gamma(l+1) = M - trace\left\{\left[\underline{\underline{A}}(\underline{w})\Big|_{\underline{w}=\underline{w}(l)}\right]^{-1}\right\} = \sum_{i=1}^{M}\frac{\nu_i(l)}{\nu_i(l)+\alpha(l)} \tag{4.18}$$

$$\alpha(l+1) = \frac{\gamma(l+1)}{\sum_{j=1}^{M}\left[w_j(l)\right]^2} \tag{4.19}$$

$$\beta(l+1) = \frac{N-\gamma(l+1)}{\sum_{k=1}^{N}\left\{d_k - f\left[\underline{x}_k, \underline{w}(n)\right]\right\}^2} \tag{4.20}$$

Na equação (4.20), $\nu_i(l)$ representa o *i*-ésimo autovalor da matriz *hessiana* $\underline{\underline{H}}(\underline{w})\Big|_{\underline{w}=\underline{w}(l)}$.

A escolha da probabilidade *a priori* $p(\underline{w})$ dada pela equação (4.2), conduzindo ao funcional $S(\underline{w})$ apresentado na equação (4.6), apresenta inconsistência em relação às propriedades de escalonamento dos mapeamentos a serem realizados pelos MLPs. Quando aplicadas transformações lineares às entradas e/ou à saída, funcionais consistentes devem dar origem a modelos similares àqueles treinados com o conjunto original de dados, a menos de transformações aplicadas aos seus parâmetros. Esta propriedade garante que este tipo de funcional não favorece de forma arbitrária um modelo em detrimento do outro, visto que ambos são equivalentes. Um funcional regularizador como o apresentado na equação (4.6), que aborda de maneira equânime

todo o conjunto de parâmetros $\underline{w}$, não satisfaz esta propriedade [50], sendo necessária a escolha de outro tipo de funcional, ou seja, outra distribuição *a priori* $p(\underline{w})$.

Além de questões relacionadas à consistência, intuitivamente a escolha de uma única distribuição de probabilidade $p(\underline{w})$ para todo o conjunto de pesos não parece razoável. Esta especificação pressupõe comportamento semelhante para todos os parâmetros, não considerando a função específica de cada um. Ao contrário desta premissa, é esperado que alguns grupos de pesos, como aqueles que ligam as entradas à camada oculta, apresentem comportamento distinto em relação a outro conjunto de parâmetros, como os oriundos da camada oculta que alimentam a saída. Desta forma, é natural especificar uma distribuição *a priori* $p(\underline{w})$ que reflita tal conhecimento prévio sobre o problema. A escolha de diferentes distribuições *gaussiana*s, todas com vetor média nulo e diferindo somente nas matrizes de covariância para distintos grupos de pesos, atende a este objetivo. Além de refletir de maneira mais fidedigna algum conhecimento prévio, esta escolha conduz a funcionais regularizadores consistentes, ao contrário da especificação dada pela equação (4.2).

Para esta nova distribuição de probabilidade *a priori* $p(\underline{w})$, seja g o número de conjuntos nos quais os pesos e bias são agrupados, $\underline{w}_i \in \mathbb{R}^{M_i}$, $\underline{w}_i = \begin{bmatrix} w_{i1} & w_{i2} & \dots & w_{iM_i} \end{bmatrix}^t$, o vetor contendo os M_i elementos do *i*-ésimo grupo, e α_i o hiperparâmetro associado. A probabilidade *a priori* $p(\underline{w}_i)$ associada ao conjunto de pesos representado pelo vetor $\underline{w}_i$ é dada por:

$$p(\underline{w}_i) = \frac{1}{\left(\frac{2\pi}{\alpha_i}\right)^{\frac{M_i}{2}}} e^{-\frac{1}{2}\alpha_i \|\underline{w}_i\|^2} \tag{4.21}$$

Considerando independência entre os grupos de pesos, a distribuição *a priori* $p(\underline{w})$ para todo o conjunto de pesos $\underline{w} \in \mathbb{R}^M$, $\underline{w} = \left[\underline{w}_1 \quad \underline{w}_2 \quad ... \quad \underline{w}_g \right]^t$, $M = \sum_{i=1}^{g} M_i$, passa a ser dada por:

$$p(\underline{w}) = \prod_{i=1}^{g} p(\underline{w}_i) = \frac{1}{\prod_{i=1}^{g} \left(\frac{2\pi}{\alpha_i} \right)^{\frac{M_i}{2}}} e^{-\frac{1}{2}\sum_{i=1}^{g} \alpha_i \|\underline{w}_i\|^2} \tag{4.22}$$

Analogamente ao desenvolvimento anterior, ou seja, substituindo (4.22) juntamente com a distribuição do ruído aditivo ζ dada pela equação (4.5) na regra de *Bayes* apresentada na equação (4.1), a maximização da probabilidade *a posteriori* $p(\underline{w}|Y)$ é obtida através da minimização do funcional $S(\underline{w})$ dado por:

$$S(\underline{w}) = \frac{\beta}{2} \sum_{k=1}^{N} \left[d_k - f(\underline{x}_k, \underline{w}) \right]^2 + \frac{1}{2} \sum_{i=1}^{g} \left(\alpha_i \sum_{j=1}^{M_i} w_{ij}^2 \right) \tag{4.23}$$

Mantendo a analogia, neste contexto visando à maximização do logaritmo natural da evidência para $\underline{\alpha} \in \mathbb{R}^g$, $\underline{\alpha} = \left[\alpha_1 \quad \alpha_2 \quad ... \quad \alpha_g \right]^t$, são obtidas as seguintes expressões para estimativa destes hiperparâmetros:

$$\gamma_i = \alpha_i \sum_{j=1}^{M_i} \left(w_{ij}^* \right)^2 = M_i - trace\left\{ \underline{\underline{B}}_i \right\} \tag{4.24}$$

$$\gamma = \sum_{i=1}^{g} \gamma_i$$

$$\underline{\underline{B}}_i = \left[\underline{\underline{A}}(\underline{w}) \Big|_{\underline{w}=\underline{w}^*} \right]^{-1} \underline{\underline{I}}_i \tag{4.25}$$

Nas expressões acima, $\underline{\underline{I}}_i \in \mathbb{R}^M \times \mathbb{R}^M$ é uma matriz diagonal nula, com elementos iguais a um apenas para as componentes do vetor $\underline{w}$ pertencentes ao *i*-ésimo grupo de pesos, com γ_i respondendo pelo número efetivo de parâmetros associado a este conjunto.

Visto que a distribuição de probabilidade *a priori* do ruído ζ continua sendo dada pela equação (4.5), a maximização do logaritmo natural da evidência para β ainda é expressa pela equação (4.17).

Seguindo a analogia, as expressões iterativas para o cálculo dos hiperparâmetros $\underline{\alpha}$ são dadas por:

$$\underline{\underline{B}}_i(l) = \left[\underline{\underline{A}}(\underline{w})\Big|_{\underline{w}=\underline{w}(l)} \right]^{-1} \underline{\underline{I}}_i \tag{4.26}$$

$$\gamma_i(l+1) = M_i - trace\left\{ \underline{\underline{B}}_i(l) \right\} \tag{4.27}$$

$$\alpha_i(l+1) = \frac{\gamma_i(l+1)}{\left\| \underline{w}_i(l) \right\|^2} \tag{4.28}$$

$$\gamma(l+1) = \sum_{i=1}^{g} \gamma_i(l+1) \tag{4.29}$$

De posse destas equações, o algoritmo de treinamento *bayesiano* de MLP's pode ser resumido da forma que segue:

1. Faça $l = 0$.
2. Inicialize o vetor de parâmetros $\underline{w}(l)$ e os hiperparâmetros $\alpha(l)$ e $\beta(l)$.
3. Utilizando alguma técnica de otimização, atualize o vetor de parâmetros $\underline{w}(l+1)$ através da minimização do funcional $S(\underline{w})$ dado pela equação (4.23).
4. Se o critério de parada for atendido, encerre o algoritmo. Do contrário, vá para o passo 5.
5. Calcule a matriz *hessiana* $\underline{\underline{H}}(\underline{w})\Big|_{\underline{w}=\underline{w}(l)}$. Um algoritmo completo para cálculo desta matriz para MLPs pode ser encontrado em [175].
6. Atualize os hiperparâmetros $\alpha_i(l+1)$ e $\beta(l+1)$, utilizando as equações (4.26) a (4.29) para $\alpha_i(l+1)$ e a equação (4.17) para $\beta(l+1)$, respectivamente.

7. Faça $l = l + 1$ e retorne ao passo 3.

Apesar de alguns pressupostos não verificados na prática, por exemplo aproximação *gaussiana* da probabilidade *a posteriori* de $\underline{w}$, o treinamento *bayesiano* apresenta como principal vantagem a estimativa automática dos diversos parâmetros de regularização, através do cálculo dos hiperparâmetros $\underline{\alpha}$ e β, evitando a necessidade de um conjunto de validação.

Além do procedimento analítico para estimativa dos parâmetros de regularização, este método fornece técnicas para seleção automática de entradas e avaliação de estruturas. Um agrupamento específico de pesos dá origem a uma metodologia para mensuração da importância de cada entrada conhecida como determinação automática de relevância (ARD), com o cálculo da evidência para o modelo fornecendo um índice para comparação entre diversas estruturas. Este índice pode ser utilizado para determinação do número de neurônios na camada oculta de MLPs. Estas técnicas serão apresentadas a seguir, começando com a técnica de seleção de entradas conhecida como ARD.

4.2 Determinação Automática de Relevância – ARD

A utilização de funcionais regularizadores da forma dada nas equações (4.6) e (4.23) favorece estruturas com parâmetros apresentando pequena magnitude, visando à modelagem de mapeamentos suaves. O equilíbrio entre o ajuste do modelo aos dados e a suavidade do mapeamento gerado é responsabilidade dos parâmetros de regularização utilizados, ou seja, dos hiperparâmetros $\underline{\alpha}$ e β no contexto *bayesiano*. Estas equações mostram que o hiperparâmetro β pondera diretamente o ajuste dos dados, com os demais hiperparâmetros $\underline{\alpha}$ sendo responsáveis pelo controle da magnitude dos respectivos conjuntos de parâmetros do modelo.

Observando a equação (4.23) sob o ponto de vista de otimização, elevados valores de α_i aumentam a importância do *i*-ésimo funcional na otimização de $S(\underline{w})$. Desta forma, para minimização de $S(\underline{w})$, conjuntos de pesos possuindo hiperparâmetros α_i com valor considerável devem apresentar pequena magnitude. No contexto probabilístico, para estimativa de $p(\underline{w}|Y)$, a inferência *bayesiana* atualiza a informação prévia $p(\underline{w})$ à luz dos dados. Como cada conjunto de pesos $\underline{w}_i$ apresenta distribuição normal com vetor média nulo e matriz de covariância regida por α_i, elevados valores para este hiperparâmetro diminuem a incerteza da informação prévia, reduzindo a influência dos dados no cálculo da probabilidade *a posteriori* $p(\underline{w}|Y)$. Portanto, quanto maior α_i, menor a magnitude de $\underline{w}_i$ obtido através da maximização de $p(\underline{w}|Y)$.

A partir de um agrupamento específico dos parâmetros que definem o modelo neural, a relação entre o valor do hiperparâmetro α_i e a magnitude do vetor de pesos $\underline{w}_i$ pode ser utilizada para mensuração da relevância de cada entrada no cálculo da saída do modelo. Este procedimento, conhecido como determinação automática de relevância (ARD), monitora a contribuição de cada entrada para o cálculo da saída, atribuindo aos sinais mais relevantes pesos de magnitude elevada, com aqueles menos relevantes possuindo conjunto de parâmetros com reduzida magnitude.

Este método de seleção de entradas particiona o vetor de parâmetros $\underline{w} \in \mathbb{R}^M$, em $n+3$ grupos, ou seja, $\underline{w} = [\underline{w}_1 \quad \underline{w}_2 \quad \ldots \quad \underline{w}_{n+3}]^t$, $M = \sum_{i=1}^{n+3} M_i$. Os n primeiros grupos representam os pesos que ligam cada uma das entradas à camada oculta. Portanto, para modelos com m neurônios na camada escondida, cada grupo apresenta

um total de pesos M_i igual a m. Os três conjuntos restantes são responsáveis pelos demais parâmetros do modelo, a saber: os bias dos neurônios na camada intermediária, perfazendo m parâmetros; os m pesos que ligam a camada oculta à saída; e o bias da única saída linear. Esta escolha específica de agrupamento dos pesos, visando aglutinar a contribuição de cada entrada no cálculo da saída, permite ponderar a relevância de cada sinal através da análise dos n hiperparâmetros α_i. Entradas com valores elevados para α_i ao final do processo de treinamento são ligadas ao modelo através de pesos com pequena magnitude, contribuindo menos para o cálculo da saída. Portanto, quanto maior α_i, menor a importância da respectiva entrada no cálculo da saída.

Utilizando este agrupamento específico de pesos, a análise dos hiperparâmetros $\underline{\alpha}$ obtidos ao final do treinamento *bayesiano* fornece uma metodologia para avaliação da relevância de cada entrada. Mesmo sendo capaz de ordenar estas variáveis segundo a importância de cada uma no cálculo da saída, este método não apresenta ferramentas para detecção de entradas irrelevantes. A literatura mostra que, além de aumentar os requisitos de armazenamento e aquisição de sinal, a utilização de sinais desta natureza pode comprometer o desempenho de previsão do modelo final [60]. Desta forma, além da mensuração da importância de cada entrada, é necessária a definição de um limiar de relevância α_0 para detecção de variáveis irrelevantes.

O método baseado na inserção de variáveis aleatórias de prova apresentado na seção 3.2 é utilizado para definição empírica do limiar de relevância α_0. Neste caso, a evidência para os hiperparâmetros desempenha o papel da medida de desempenho, com os hiperparâmetros $\underline{\alpha}$ responsáveis pela mensuração da relevância de cada entrada. Assim, após o treinamento do modelo utilizando o espaço de entrada estendido obtido pela inserção de variáveis de prova ao espaço original, variáveis apresentando α_i maior

que o respectivo patamar α_0 (α_C para variáveis contínuas e α_D para discretas) são descartadas do modelo final, visto que estas contribuem menos que a variável sabidamente irrelevante para o cálculo da saída.

Além de um procedimento automático para seleção de sinais de entrada, a aplicação de inferência *bayesiana* ao treinamento de MLPs fornece um método para seleção de modelos baseado no cálculo da evidência. Este procedimento, que visa determinar o modelo mais provável à luz dos dados, pode ser utilizado para determinação do número de neurônios na camada oculta, sendo apresentado com mais detalhes na próxima seção.

4.3 Seleção *Bayesiana* de Modelos

A inferência *bayesiana* também pode ser utilizada para seleção da melhor estrutura em uma série de hipóteses $\mathrm{H} = \{H_1, H_2, ..., H_K\}$. Pela regra de *Bayes*, a distribuição de probabilidade *a posteriori* $p(H_h|Y)$ da hipótese H_h é dada por:

$$p(H_h|Y) = \frac{p(Y|H_h)\,p(H_h)}{p(Y)} \tag{4.30}$$

Visto que $p(Y)$ é um fator de normalização e admitindo que todas as hipóteses H_h são equiprováveis *a priori*, a evidência $p(Y|H_h)$ pode ser utilizada para avaliação de modelos, sendo selecionado aquele com maior probabilidade *a posteriori* $p(H_h|Y)$, ou seja, maior evidência [50]. Considerando MLPs com uma única camada oculta contendo m neurônios e utilizando uma aproximação *gaussiana* em torno dos hiperparâmetros $\underline{\alpha}$ e β obtidos no final do processo de treinamento, é obtida a seguinte expressão para $\ln p(Y|H_h)$:

$$\ln p\left(Y\middle|H_h\right) = -S\left(\underline{w}\right) - \frac{1}{2}\ln\left|\underline{\underline{A}}\left(\underline{w}\right)\right| + \frac{1}{2}\sum_{i=1}^{g} M_i\alpha_i + \frac{N}{2}\ln\beta \tag{4.31}$$
$$+2\ln m + \ln m! + \frac{1}{2}\sum_{i=1}^{g}\ln\left(\frac{2}{\gamma_i}\right) + \frac{1}{2}\ln\left(\frac{2}{N-\gamma}\right)$$

Na expressão acima, o funcional $S\left(\underline{w}\right)$ e a matriz $\underline{\underline{A}}\left(\underline{w}\right)$ são avaliados no ponto $\underline{w} = \underline{w}^*$ obtido ao final do algoritmo do treinamento, sendo também utilizados nesta equação os hiperparâmetros $\underline{\alpha}$, β e os respectivos números efetivos de parâmetros γ_i estimados ao longo do processo. Além disso, utilizando ARD, o número de grupos g é igual a $n+3$, onde n responde pelo número de entradas.

A aplicação da regra de *Bayes* dada pela equação (4.30) para avaliação de modelos permite a comparação entre diversas hipóteses, incluindo por exemplo redes de função de base radial e modelos lineares. Porém, neste trabalho este índice será utilizado somente para determinação do número de neurônios na camada oculta de MLPs, através da escolha da estrutura que apresentar maior evidência, ou seja, a mais provável à luz dos dados. Apesar da evidência não apresentar relação direta com o erro de generalização, dispensando inclusive o uso de um conjunto de validação para avaliação do modelo, a incorporação de mecanismos de penalização de modelos excessivamente complexos por parte do treinamento *bayesiano* permite inserir a evidência para os modelos no contexto de índices analíticos de avaliação de estruturas, como AIC, BIC [172], NIC [125], MDL [50], [61] e dimensão VC [61], [62].

4.4 Resumo e discussão

O cálculo da evidência para os modelos como mecanismo de seleção de estrutura reflete o caráter hierárquico da inferência *bayesiana*. No início da apresentação da teoria, a regra de *Bayes* foi aplicada para estimação dos parâmetros do modelo, supondo que os hiperparâmetros que regem as probabilidades *a priori* envolvidas eram

conhecidos e sendo definida uma estrutura para o modelo. Visto que os hiperparâmetros também são desconhecidos, o fator de normalização $p(Y)$ presente na equação (4.1) nada mais é que a evidência para os hiperparâmetros, utilizada na regra de *Bayes* apresentada na equação (4.9) e maximizada posteriormente. Neste ponto, é suposto que o modelo é conhecido, o que não corresponde à realidade. Assim, a parcela $p(Y)$ presente no denominador da equação (4.9) é a evidência para os modelos, utilizada para seleção da estrutura e apresentada na equação (4.31). Desta forma, a evidência nos níveis inferiores do processo de estimação corresponde ao fator de normalização no estágio seguinte, realçando o aspecto hierárquico da inferência *bayesiana.*

Neste trabalho, uma abordagem *bayesiana* ao problema de especificação e treinamento de MLPs, através do procedimento proposto por *Mackay* [54] e conhecido como maximização da evidência, é utilizada para o desenvolvimento deste tipo de modelo para previsão de carga. Todos os níveis de inferência, desde a estimação do vetor de parâmetros $\underline{w}$ até a escolha do modelo mais provável à luz dos dados, são percorridos. Os resultados obtidos no segundo nível de inferência, relacionado com a estimação dos hiperparâmetros, são analisados em um procedimento de seleção de entradas, oriundo da utilização de distribuições *a priori* definidas no método conhecido como determinação automática de relevância. O modelo alimentado somente com as entradas relevantes previamente selecionadas é então treinado novamente, com a evidência para este modelo sendo calculada. Finalmente, para realização das previsões, é utilizado o modelo que apresentar maior evidência. O algoritmo *bayesiano* para desenvolvimento de modelos neurais de previsão de carga pode ser resumido da maneira que segue:

1. Escolha o número mínimo $m_{\min}$ e máximo $m_{\max}$ de neurônios na camada oculta.
2. Para $i = m_{\min}, ..., m_{\max}$, faça:

2.1. Insira variáveis de prova no conjunto de entradas original seguindo as distribuições de probabilidade apresentadas nas equações (3.36) e (3.37). Se as variáveis de entrada forem somente contínuas, insira somente o sinal de prova desta natureza. Do contrário, insira também a variável de prova discreta.

2.2. Estime o vetor de parâmetros $\underline{w}$ e o conjunto de hiperparâmetros $\underline{\alpha}$ e β, utilizando o algoritmo apresentado na seção 4.1.

2.3. Separe os hiperparâmetros α_i associados a entradas contínuas e discretas em vetores distintos e ordene de forma decrescente estes vetores.

2.4. Em cada vetor, selecione as variáveis de entrada situadas acima do respectivo limiar de relevância, α_C para variáveis contínuas e α_D para as discretas.

2.5. Utilizando somente as variáveis relevantes selecionadas no passo 2.4, estime novamente o vetor de parâmetros $\underline{w}$ e o conjunto de hiperparâmetros $\underline{\alpha}$ e β, utilizando o algoritmo apresentado na seção 4.1.

2.6. Calcule o logaritmo natural da evidência do modelo, $\ln p\left(Y\middle|H_h\right)$, dado pela equação (4.31).

3. Escolha o modelo com maior evidência e realize as previsões.

O algoritmo descrito acima é utilizado neste trabalho no desenvolvimento de modelos neurais para previsão de carga. São testados MLPs com número de neurônios sigmoidais na camada oculta variando de 1 a 10, sendo selecionada a estrutura que apresentar a maior evidência, ou seja, aquela mais provável à luz dos dados. Vale destacar que, dado um conjunto inicial de entradas, são determinadas automaticamente as variáveis mais relevantes para cada modelo. Visto que o nível de não-linearidade da

estrutura neural está diretamente associado à representação do espaço de entrada [151], é esperado que o conjunto de entradas relevantes varie entre as diversas estruturas. Assim, utilizando o método de seleção do conjunto inicial de entradas apresentado na seção 3.1, a intervenção do usuário é requerida somente para definição dos limites para a busca pelo número de neurônios na camada escondida, mostrando o considerável grau de automação e adaptabilidade do método.

Apresentada a metodologia *bayesiana* para desenvolvimentos de MLPs, resta apresentar a teoria relacionada ao outro conjunto de modelos utilizados nesta tese. Originárias da teoria de aprendizado estatístico e guardando analogia com as redes de regularização, as máquinas baseadas em *kernel* utilizadas neste trabalho também possuem técnicas automáticas para seleção de entradas e de estrutura, conforme será apresentado no próximo capítulo.

5 Modelos baseados em *kernel*

Os modelos neurais discutidos no capítulo 4 armazenam o conhecimento sobre uma dada tarefa em um vetor de parâmetros $\underline{w}$, estimado a partir da iteração do modelo $f(\underline{x},\underline{w})$ com um conjunto de dados D. Encerrada a fase de treinamento, para realização de previsões o conjunto de dados pode ser descartado, com toda a informação contida neste conjunto sendo comprimida em $\underline{w}$.

Em reconhecimento de padrões, existe uma classe de modelos que utilizam o conjunto de treinamento, ou em alguns casos um subconjunto deste, no estágio de previsão. Conhecidos como modelos baseados em *kernel*, estes métodos realizam previsões a partir de combinações das saídas de funções centradas em cada um dos pontos disponíveis. As funções utilizadas para ponderação de cada dado do conjunto de treinamento são denominadas *kernels*.

O método de *Parzen* para estimação não-paramétrica de funções de densidade de probabilidade, sucintamente apresentado no APÊNDICE A, é provavelmente o método baseado em *kernel* mais conhecido. No contexto de reconhecimento de padrões, as máquinas de vetor suporte vêm ganhando espaço ao longo dos últimos anos. Conforme será discutido neste capítulo, apesar de promissora esta técnica necessita do ajuste de alguns parâmetros, motivando o surgimento de outros métodos baseados em *kernel*, como as máquinas de vetores relevantes. Estas metodologias são utilizadas nesta tese e apresentadas neste capítulo, começando pelas máquinas de vetor suporte.

5.1 Máquinas de vetor suporte (SVMs)

As máquinas de vetor suporte (SVMs) foram desenvolvidas com base em um novo paradigma da área de aprendizado de máquina, conhecido como aprendizado estatístico. Diferentemente da abordagem clássica de problemas de classificação, que

necessitam de uma quantidade elevada de dados em conjunto com a inserção de conhecimento prévio sobre o problema, a teoria do aprendizado estatístico foi desenvolvida para solução de problemas cuja quantidade de dados disponíveis é reduzida e pouco, ou até mesmo nenhum, conhecimento prévio pode ser utilizado, características comumente encontradas em aplicações reais [61].

A teoria de SVM foi originalmente elaborada para solução de problemas de classificação, através da aplicação do conceito de hiperplano ótimo, baseado na maximização da margem de separação ρ. A Figura 5.1 ilustra a margem ρ para o caso de padrões linearmente separáveis. Nesta figura, a reta vermelha representa o hiperplano ótimo de separação, com os chamados vetores suporte sendo aqueles situados exatamente em cima das retas negras tracejadas. Estes vetores recebem esta denominação em virtude da sua proximidade da superfície de decisão, contribuindo de maneira decisiva para a definição de tal superfície [61].

A motivação para a maximização da margem ρ encontra fundamento em uma medida de complexidade conhecida como dimensão de *Vapnik* e *Chervonenkis* [61], [62], popularmente denominada dimensão VC. De acordo com o dilema bias-variância [50], o desempenho do modelo para novos dados pode ser decomposto em duas parcelas conflitantes, bias e variância, as quais estão relacionadas com o ajuste aos dados disponíveis e com o nível de flexibilidade da função estimada, respectivamente. Modelos excessivamente ajustados aos padrões de treinamento irão apresentar bias reduzido, porém elevada variância em virtude do grau de complexidade fornecido. Analogamente, modelos com elevada dimensão VC, apesar de ajustarem de forma satisfatória os dados de treinamento, apresentarão reduzida capacidade de generalização.

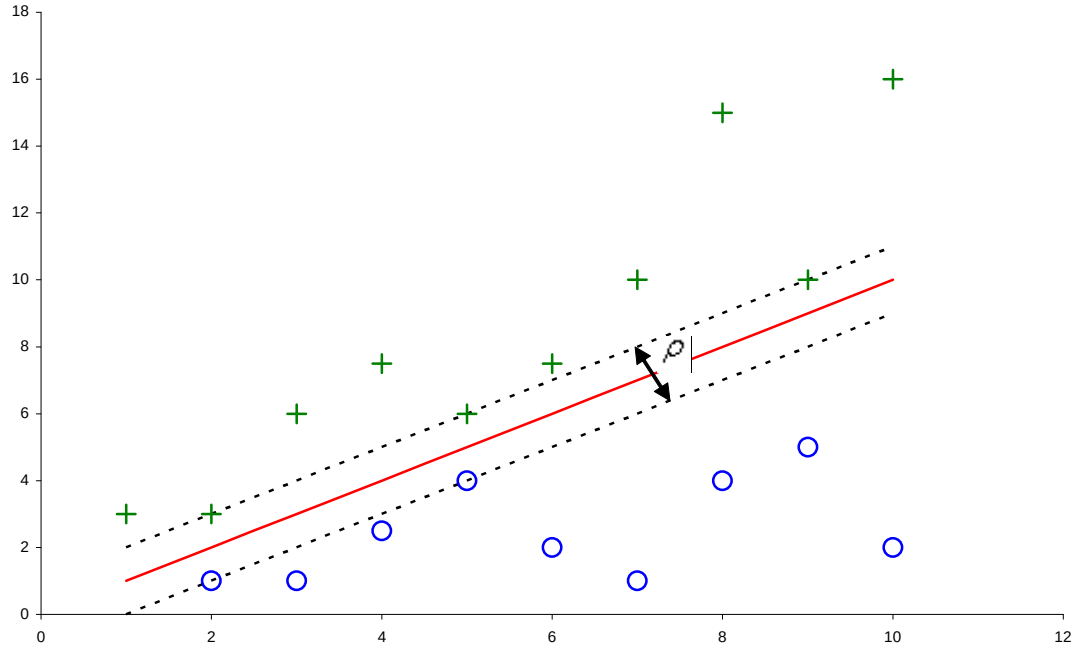

Figura 5.1 – Ilustração da margem de separação ρ para o caso de duas classes linearmente separáveis.

Ao contrário de outros índices utilizados para mensuração da complexidade, como AIC, BIC, dentre outros, esta medida não está diretamente relacionada com o número de parâmetros do modelo. Para problemas de classificação binária, a dimensão VC é dada pela cardinalidade do maior conjunto de padrões que pode ser corretamente classificado pela máquina de aprendizagem [61]. Para modelos não-lineares, o cálculo analítico desta medida de complexidade ainda é um campo em aberto, sendo sabido que a dimensão VC de modelos *feedforward* como MLPs e SVMs é finita [61]. Além disso, a dimensão VC de hiperplanos $f\left(\underline{x},\underline{W}\right)$ apresentando margem de separação ρ, $VC\left[f\left(\underline{x},\underline{W}\right)\right]$, é limitada superiormente pela relação:

$$VC\left[f\left(\underline{x},\underline{W}\right)\right] \leq \frac{R^2}{\rho^2} \tag{5.1}$$

Na equação (5.1), R é o raio da menor hiperesfera que engloba a imagem de todos os padrões no espaço de características. Portanto, maximizar a margem ρ conduz à minimização do limite superior para a dimensão VC do modelo estimado, reduzindo a complexidade da estrutura visando à obtenção de considerável capacidade de generalização.

O conceito de hiperplano ótimo de separação, também conhecido como hiperplano de margem máxima, pode ser expandido para problemas de classificação de padrões não-linearmente separáveis, através do mapeamento do espaço original de representação em um espaço de dimensão elevada, onde o problema passa a ser linearmente separável. Desta forma, as SVMs podem ser vistas como máquinas lineares aplicadas a um espaço de representação expandido, de dimensão maior que o espaço de representação original do problema, com o mapeamento que governa esta expansão sendo obtido de maneira intrínseca. Seguindo esta idéia, matematicamente, a saída de uma SVM pode ser dada por:

$$f\left(\underline{x},\underline{W},b\right)=\sum_{j=1}^{N}W_j\phi_j\left(\underline{x}\right)+b=\underline{W}^t\,\underline{\phi}\left(\underline{x}\right)+b \tag{5.2}$$

$$\underline{W}=\left[W_1 \quad W_2 \quad \ldots \quad W_N\right]^t$$

$$\underline{\phi}\left(\underline{x}\right)=\left[\phi_1\left(\underline{x}\right) \quad \phi_2\left(\underline{x}\right) \quad \ldots \quad \phi_N\left(\underline{x}\right)\right]^t$$

Na equação (5.2), $\underline{\phi}\left(\underline{x}\right):\mathbb{R}^n\rightarrow\mathbb{R}^N$ representa o mapeamento não-linear das entradas $\underline{x}$ do espaço original de representação $\mathbb{R}^n$ para um espaço de dimensionalidade elevada $\mathbb{R}^N$, $N>n$, com $\underline{W}\in\mathbb{R}^N$ representando o conjunto de parâmetros que define a máquina linear aplicada no espaço expandido. Na equação (5.2), $b\in\mathbb{R}$ representa o bias do modelo. O objetivo do mapeamento não-linear $\underline{\phi}\left(\underline{x}\right)$ consiste na mudança do espaço de representação do problema, originalmente não-linearmente separável em $\mathbb{R}^n$, para um espaço $\mathbb{R}^N$ onde o problema passa a ser linearmente separável, podendo ser

aplicado neste novo espaço o conceito de hiperplano ótimo de separação. Este novo espaço de representação também é conhecido como espaço de características [61], [62]. A Figura 5.2 apresenta um diagrama esquemático de uma SVM.

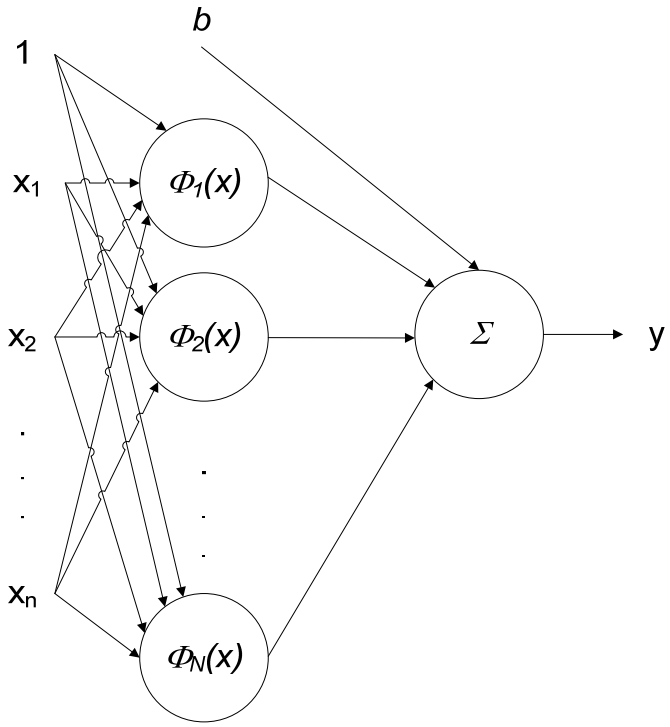

Figura 5.2 – Diagrama esquemático de uma SVM

A idéia de mapear o espaço de representação original em um espaço de maior dimensão com o intuito de tornar o problema linearmente separável encontra justificativa no teorema de *Cover*. De uma maneira informal, este teorema afirma que se o mapeamento $\underline{\phi}(\underline{x})$ for não-linear e a dimensionalidade N do espaço de características for suficientemente elevada, a probabilidade do problema ser linearmente separável neste novo espaço de representação é elevada [176].

A teoria de SVM desenvolvida para problemas de classificação foi expandida para problemas mais gerais de reconhecimento de padrões, como problemas de aproximação funcional, regressão e processamento de sinais, aumentando assim a aplicabilidade deste tipo de modelo. Visto que o problema abordado nesta tese pode ser

enquadrado na classe de problemas de aproximação funcional, a apresentação da teoria de SVMs estará restrita a essa área de aplicação, podendo ser estendida para problemas de regressão, identificação de sistemas e processamento de sinais de maneira direta.

Conforme mencionado na seção 2.2, problemas de aproximação funcional buscam a aproximação, ou interpolação, de uma função contínua $F(\underline{x}):\mathbb{R}^n \to \mathbb{R}$, por uma função $f(\underline{x},\underline{w}):\mathbb{R}^n \to \mathbb{R}$, definida por um vetor de parâmetros $\underline{w} \in \mathbb{R}^M$, utilizando para tal um conjunto de exemplos $D = \{\underline{x}_k, d_k\}$, $k = 1,2,...,N$, $\underline{x}_k \in \mathbb{R}^n$, e $d_k = F(\underline{x}_k)$, para problemas de interpolação, ou $d_k = F(\underline{x}_k) + \zeta_k$, para problemas de aproximação. Para problemas de aproximação, categoria na qual os problemas de regressão podem ser inseridos, apresentando ruído aditivo ζ com distribuição normal de probabilidade, a minimização do erro médio quadrático resulta na melhor estimativa não-tendenciosa do vetor $\underline{w}$ [61]. Entretanto, a maioria dos problemas reais de regressão não apresenta ruído aditivo *gaussiano*, trazendo à tona a necessidade da utilização de outros tipos de função de erro. Em virtude disso, SVMs aplicadas a problemas de regressão utilizam funções de erro conhecidas como funções de perda com tolerância ε, $L_\varepsilon(x,y):\mathbb{R}^2 \to \mathbb{R}$, que de uma maneira geral podem ser representadas pela seguinte equação:

$$L_\varepsilon\left[d, f(\underline{x},\underline{W},b)\right] = \begin{cases} g\left[d, f(\underline{x},\underline{W},b)\right], & \left|d - f(\underline{x},\underline{W},b)\right| \geq \varepsilon \\ 0, & \left|d - f(\underline{x},\underline{W},b)\right| < \varepsilon \end{cases} \tag{5.3}$$

Na equação (5.3), $f(\underline{x},\underline{W},b)$ representa a saída calculada pela SVM, dada pela equação (5.2), d a saída desejada associada ao vetor $\underline{x}$, $g\left[d, f(\underline{x},\underline{W},b)\right]:\mathbb{R}^2 \to \mathbb{R}$ uma função de perda e ε é um parâmetro especificado pelo usuário. Para problemas de regressão com ruído aditivo *gaussiano*, $\varepsilon \in \mathbb{R}^+$ pode representar a variância de tal ruído [177]. As

funções de perda com tolerância ε apresentam como ponto comum a definição de uma banda na qual os erros não são computados. Observando a equação (5.3), erros absolutos menores do que ε não são considerados. A principal diferença entre estas funções reside na função de perda $g\left[d, f\left(\underline{x}, \underline{W}, b\right)\right]$ utilizada na ponderação dos erros. Podem ser citadas as funções de perda com tolerância ε linear, quadrática e de *Huber*, dadas pelas seguintes equações, respectivamente:

$$L_\varepsilon\left[d, f\left(\underline{x}, \underline{W}, b\right)\right] = \begin{cases} \left|d - f\left(\underline{x}, \underline{W}, b\right)\right| - \varepsilon, & \left|d - f\left(\underline{x}, \underline{W}, b\right)\right| \geq \varepsilon \\ 0, & \left|d - f\left(\underline{x}, \underline{W}, b\right)\right| < \varepsilon \end{cases} \tag{5.4}$$

$$L_\varepsilon\left[d, f\left(\underline{x}, \underline{W}, b\right)\right] = \begin{cases} \left[\left|d - f\left(\underline{x}, \underline{W}, b\right)\right| - \varepsilon\right]^2, & \left|d - f\left(\underline{x}, \underline{W}, b\right)\right| \geq \varepsilon \\ 0, & \left|d - f\left(\underline{x}, \underline{W}, b\right)\right| < \varepsilon \end{cases} \tag{5.5}$$

$$L_\varepsilon\left[d, f\left(\underline{x}, \underline{W}, b\right)\right] = \begin{cases} \varepsilon\left|d - f\left(\underline{x}, \underline{W}, b\right)\right| - \dfrac{\varepsilon^2}{2}, & \left|d - f\left(\underline{x}, \underline{W}, b\right)\right| \geq \varepsilon \\ \dfrac{1}{2}\left[\left|d - f\left(\underline{x}, \underline{W}, b\right)\right|\right]^2, & \left|d - f\left(\underline{x}, \underline{W}, b\right)\right| < \varepsilon \end{cases} \tag{5.6}$$

Os gráficos da Figura 5.3 à Figura 5.5 apresentam as funções de perda com tolerância ε dadas pelas equações (5.4), (5.5) e (5.6), respectivamente.

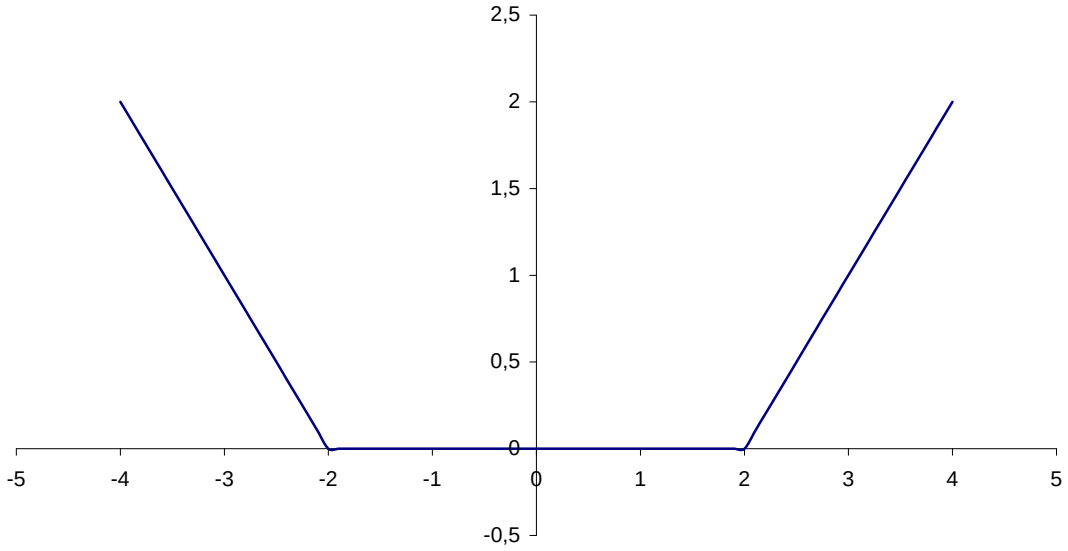

Figura 5.3 – Gráfico da função linear de perda dada pela equação (5.4), para $\varepsilon = 2$

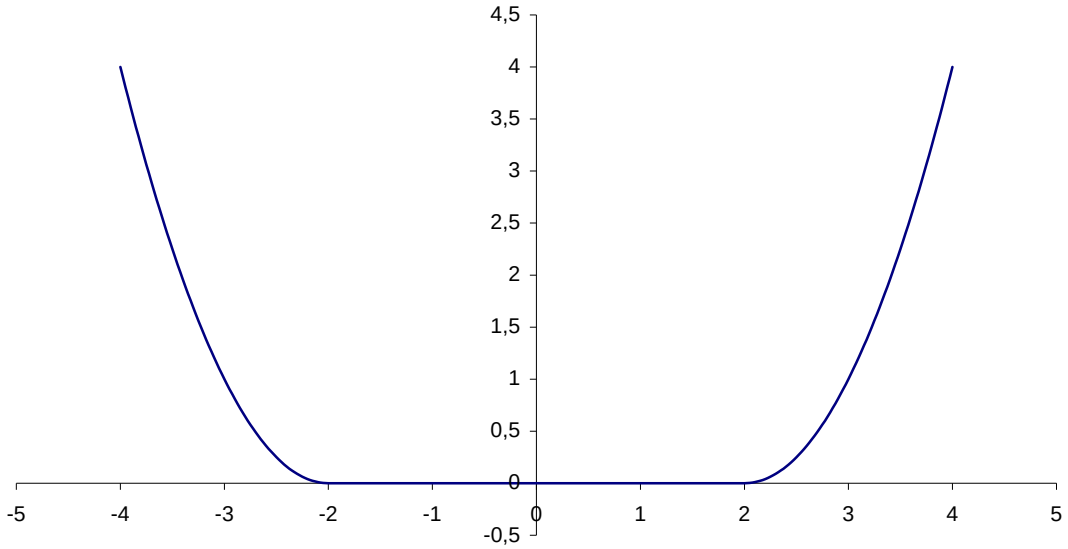

Figura 5.4 – Gráfico da função quadrática de perda dada pela equação (5.5), para $\varepsilon = 2$

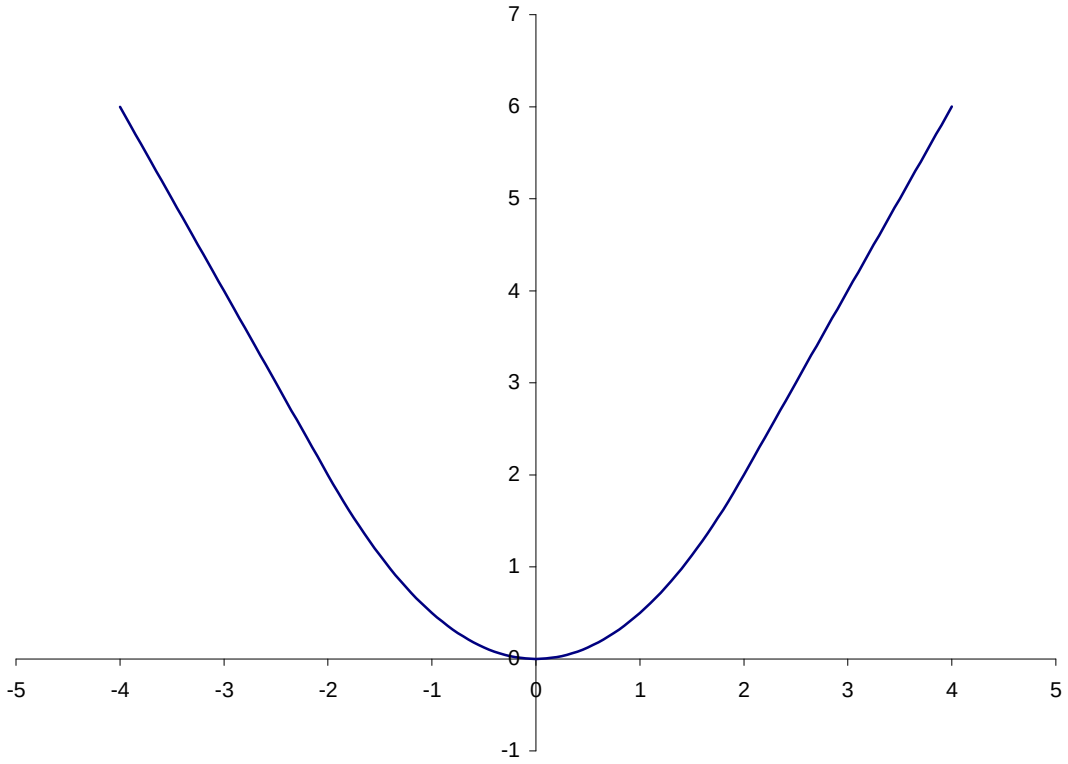

Figura 5.5 – Gráfico da função de perda de *Huber* dada pela equação (5.6), para $\varepsilon = 2$

O uso deste tipo de função de erro também encontra motivação nos problemas de classificação, para os quais as SVMs foram originalmente desenvolvidas. Nestes

problemas, existe uma vasta área do espaço de representação cujo valor da função de erro é nulo, ou seja, os padrões são corretamente classificados. Em outras palavras, só contribuem para o processo de otimização responsável pela determinação do hiperplano ótimo os padrões situados no interior da margem ilustrada na Figura 5.1, para o caso específico de padrões linearmente separáveis. Com o intuito de manter a analogia com o desenvolvimento das SVMs para problemas de classificação, é importante que a função de erro utilizada para aproximação funcional também apresente uma região cujo valor seja nulo, característica marcante das funções de perda com tolerância ε. A Figura 5.6 ilustra esta característica. Nesta Figura, a linha verde representa a função de aproximação $f\left(\underline{x},\underline{W},b\right)$, com as linhas vermelhas determinando a margem, ou tolerância, igual a $f\left(\underline{x},\underline{W},b\right)\pm\varepsilon$, da aproximação realizada por $f\left(\underline{x},\underline{W},b\right)$. Desta forma, serão penalizados, ou seja, apresentarão valores não-nulos da função $L_\varepsilon\left[d, f\left(\underline{x},\underline{W},b\right)\right]$, apenas os pontos situados fora da banda determinada por $f\left(\underline{x},\underline{W},b\right)\pm\varepsilon$.

Dada uma função de perda com tolerância ε, o objetivo do treinamento de uma SVM para problemas de aproximação funcional reside na minimização restrita do risco empírico $E_s\left(\underline{w}\right)$ dado pela equação:

$$\min_{\underline{W},b} E_s\left(\underline{W}\right)=\frac{1}{N}\sum_{i=1}^{N} L_\varepsilon\left[d_k, f\left(\underline{x}_k,\underline{W},b\right)\right] \tag{5.7}$$

$$s.a.$$

$$\left\|\underline{W}\right\|^2 \le c_0$$

A restrição do problema de otimização descrito na equação (5.7) tem origem na maximização da margem de separação ρ para problemas de classificação, com $c_0 \in \mathbb{R}^+$ sendo uma constante responsável pela regularização do modelo.

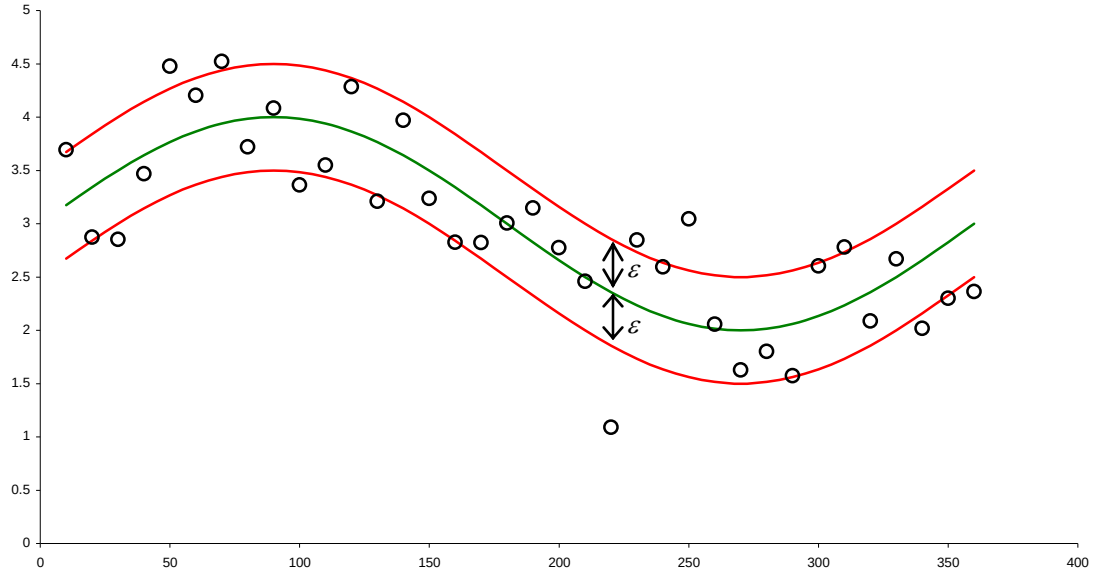

Figura 5.6 – Ilustração do papel do parâmetro ε

Apesar de terem sido apresentados três tipos de funções de perda com tolerância ε, este trabalho focará apenas na função quadrática, dada pela equação (5.5). A utilização desta função permite a obtenção de limites superiores analíticos e diferenciáveis para o erro de generalização estimado por validação cruzada única (*leave-one-out*). A otimização destes limites utilizando algoritmos de descida em gradiente pode ser utilizada para estimação dos parâmetros que definem a SVM. O desenvolvimento da teoria de SVM utilizando as funções dadas pelas equações (5.4) e (5.6) pode ser encontrado em [61] e [62].

Conforme apresentado na equação (5.5) e ilustrado na Figura 5.4, a função quadrática de perda com tolerância ε não é continuamente diferenciável. Esta característica indesejada pode ser abordada através da introdução de dois conjuntos de variáveis de folga, $\underline{\xi}_1$, $\underline{\xi}_2 \in \mathbb{R}^N$, $\underline{\xi}_1 = [\xi_{11}, \xi_{12}, ..., \xi_{1N}]^t$ e $\underline{\xi}_2 = [\xi_{21}, \xi_{22}, ..., \xi_{2N}]^t$, definidas pelas seguintes equações:

$$
\begin{aligned}
& d_k - f\left(\underline{x}_k, \underline{W}, b\right) = d_k - \underline{W}^t \underline{\phi}\left(\underline{x}_k\right) - b \le \varepsilon + \xi_{1k} \\
& -\left[d_k - f\left(\underline{x}_k, \underline{W}, b\right)\right] = \underline{W}^t \underline{\phi}\left(\underline{x}_k\right) + b - d_k \le \varepsilon + \xi_{2k} \\
& \xi_{1k} \ge 0 \\
& \xi_{2k} \ge 0 \\
& k = 1, 2, ..., N
\end{aligned}
\tag{5.8}
$$

Utilizando a função quadrática de perda com tolerância ε, dada pela equação (5.5), em conjunto com as variáveis de folga $\underline{\xi}_1$ e $\underline{\xi}_2$, o problema de otimização descrito na equação (5.7) pode ser formulado de forma equivalente pelo seguinte problema de minimização restrito:

$$
\begin{aligned}
& \min_{\underline{W}, b} \mathrm{A}\left(\underline{\xi}_1, \underline{\xi}_2\right) = \sum_{k=1}^{N} \left(\xi_{1k}\right)^2 + \left(\xi_{2k}\right)^2 \\
& s.a \\
& d_k - \underline{W}^t \underline{\phi}\left(\underline{x}_k\right) - b \le \varepsilon + \xi_{1k} \\
& \underline{W}^t \underline{\phi}\left(\underline{x}_k\right) + b - d_k \le \varepsilon + \xi_{2k} \\
& \xi_{1k} \ge 0 \\
& \xi_{2k} \ge 0 \\
& \left\|\underline{W}\right\|^2 \le c_0 \\
& k = 1, 2, ..., N
\end{aligned}
\tag{5.9}
$$

O problema dado pela equação (5.8) apresenta uma restrição não-linear, $\left\|\underline{W}\right\|^2 \le c_0$, impossibilitando a aplicação de técnicas analíticas de otimização desenvolvidas para problemas com restrições lineares. Para contornar esta questão, esta restrição pode ser abordada diretamente na função objetivo de um novo problema de otimização quadrática, dado por [61]:

$$
\begin{aligned}
& \min_{\underline{W}, b} \Phi\left(\underline{W}, b, \underline{\xi}_1, \underline{\xi}_2\right) = C \sum_{k=1}^{N} \left[\left(\xi_{1k}\right)^2 + \left(\xi_{2k}\right)^2\right] + \frac{1}{2} \underline{W}^t \underline{W} \\
& s.a \\
& d_k - \underline{W}^t \underline{\phi}\left(\underline{x}_k\right) - b \le \varepsilon + \xi_{1k} \\
& \underline{W}^t \underline{\phi}\left(\underline{x}_k\right) + b - d_k \le \varepsilon + \xi_{2k} \\
& \xi_{1k} \ge 0; \xi_{2k} \ge 0; k = 1, 2, ..., N
\end{aligned}
\tag{5.10}
$$

Neste novo problema, $C \in \mathbb{R}^+$ é uma constante responsável pelo equilíbrio entre o ajuste dos dados de treinamento e a complexidade do modelo, seguindo o princípio de minimização do risco estrutural. Este compromisso entre o desempenho para o conjunto de treinamento e a complexidade da estrutura estimada guarda analogia com a teoria da regularização, apresentada na seção 2.2, e com o dilema bias-variância. Conforme mencionado anteriormente, no contexto da teoria de aprendizado estatístico a variância é dada pela dimensão VC do modelo, com o equilíbrio entre a sua minimização implícita e o ajuste dos dados sendo responsabilidade do parâmetro C utilizado na equação (5.10).

Visto que a descontinuidade da função de perda foi tratada através da inserção das variáveis de folga $\underline{\xi}_1$ e $\underline{\xi}_2$, e a restrição não-linear $\|\underline{W}\|^2 \leq c_0$ foi abordada diretamente na função objetivo descrita na equação (5.10), técnicas analíticas podem ser utilizadas para solução deste problema. Uma das técnicas mais populares para solução de problemas de otimização restrita da forma apresentada na equação (5.10) é conhecida como regra dos multiplicadores de *Lagrange.* Sejam $\underline{\alpha}_1, \underline{\alpha}_2, \underline{\gamma}_1, \underline{\gamma}_2 \in \mathbb{R}^N$, $\underline{\alpha}_1 = [\alpha_{11} \quad ... \quad \alpha_{1N}]^t$, $\underline{\alpha}_2 = [\alpha_{21} \quad ... \quad \alpha_{2N}]^t$, $\underline{\gamma}_1 = [\gamma_{11} \quad ... \quad \gamma_{1N}]^t$, $\underline{\gamma}_2 = [\gamma_{21} \quad ... \quad \gamma_{2N}]^t$, os vetores contendo os multiplicadores de *Lagrange*, com o funcional *Lagrangeano* $L\left(\underline{W}, b, \underline{\xi}_1, \underline{\xi}_2, \underline{\alpha}_1, \underline{\alpha}_2, \underline{\gamma}_1, \underline{\gamma}_2\right)$ relacionado ao problema descrito na equação (5.10) dado por:

$$L\left(\underline{W}, b, \underline{\xi}_1, \underline{\xi}_2, \underline{\alpha}_1, \underline{\alpha}_2, \underline{\gamma}_1, \underline{\gamma}_2\right) = C\sum_{k=1}^{N}\left[\left(\xi_{1k}\right)^2 + \left(\xi_{2k}\right)^2\right] + \frac{1}{2}\underline{W}^t\underline{W} \tag{5.11}$$
$$-\sum_{k=1}^{N}\left(\gamma_{1k}\xi_{1k} + \gamma_{2k}\xi_{2k}\right) - \sum_{k=1}^{N}\alpha_{1k}\left\{\underline{W}^t\underline{\phi}\left(\underline{x}_k\right) + b - d_k + \varepsilon + \xi_{1k}\right\}$$
$$-\sum_{k=1}^{N}\alpha_{2k}\left\{d_k - \underline{W}^t\underline{\phi}\left(\underline{x}_k\right) - b + \varepsilon + \xi_{2k}\right\}$$

O ponto de sela do funcional dado pela equação (5.11) fornece a solução do problema de otimização descrito na equação (5.10) [61]. Este ponto é determinado através da minimização, em relação ao vetor de parâmetros $\underline{W}$, ao bias b e às variáveis de folga $\underline{\xi}_1$ e $\underline{\xi}_2$, de $L\left(\underline{W},b,\underline{\xi}_1,\underline{\xi}_2,\underline{\alpha}_1,\underline{\alpha}_2,\underline{\gamma}_1,\underline{\gamma}_2\right)$, e a posterior maximização deste mesmo funcional em relação aos multiplicadores de *Lagrange* $\underline{\alpha}_1$, $\underline{\alpha}_2$, $\underline{\gamma}_1$ e $\underline{\gamma}_2$. O problema de minimização a ser resolvido também é conhecido como problema primal, com o posterior problema de maximização sendo chamado de problema dual.

Utilizando as condições de otimalidade do cálculo, ou seja, $\nabla L\left(\underline{W},b,\underline{\xi}_1,\underline{\xi}_2,\underline{\alpha}_1,\underline{\alpha}_2,\underline{\gamma}_1,\underline{\gamma}_2\right)=\underline{0}$, a minimização de (5.11) em relação a $\underline{W}$, b, $\underline{\xi}_1$ e $\underline{\xi}_2$ resulta nas seguintes expressões, respectivamente:

$$\nabla_{\underline{W}} L\left(\underline{W},b,\underline{\xi}_1,\underline{\xi}_2,\underline{\alpha}_1,\underline{\alpha}_2,\underline{\gamma}_1,\underline{\gamma}_2\right)=\underline{0}\Rightarrow \underline{W}=\sum_{k=1}^{N}\left(\alpha_{1k}-\alpha_{2k}\right)\underline{\phi}\left(\underline{x}_k\right) \tag{5.12}$$

$$\frac{\partial}{\partial b} L\left(\underline{W},b,\underline{\xi}_1,\underline{\xi}_2,\underline{\alpha}_1,\underline{\alpha}_2,\underline{\gamma}_1,\underline{\gamma}_2\right)=0\Rightarrow \sum_{k=1}^{N}\left(\alpha_{1k}-\alpha_{2k}\right)=0 \tag{5.13}$$

$$\nabla_{\underline{\xi}_1} L\left(\underline{W},b,\underline{\xi}_1,\underline{\xi}_2,\underline{\alpha}_1,\underline{\alpha}_2,\underline{\gamma}_1,\underline{\gamma}_2\right)=\underline{0}\Rightarrow \underline{\gamma}_1=2\underline{\xi}_1 C-\underline{\alpha}_1 \tag{5.14}$$

$$\nabla_{\underline{\xi}_2} L\left(\underline{W},b,\underline{\xi}_1,\underline{\xi}_2,\underline{\alpha}_1,\underline{\alpha}_2,\underline{\gamma}_1,\underline{\gamma}_2\right)=\underline{0}\Rightarrow \underline{\gamma}_2=2\underline{\xi}_2 C-\underline{\alpha}_2 \tag{5.15}$$

Nas equações acima, $\nabla_{\underline{a}} L$ significa o vetor constituindo pelos componentes do gradiente $\nabla L\left(\underline{W},b,\underline{\xi}_1,\underline{\xi}_2,\underline{\alpha}_1,\underline{\alpha}_2,\underline{\gamma}_1,\underline{\gamma}_2\right)$ associados ao vetor $\underline{a}$. Estas equações constituem a solução do problema primal. Substituindo estes resultados na equação (5.11), é obtido o problema dual de maximização, dado por [179]:

$$\max_{\underline{\alpha}_1,\underline{\alpha}_2} \Psi\left(\underline{\alpha}_1,\underline{\alpha}_2\right)=\sum_{k=1}^{N} d_k\left(\alpha_{1k}-\alpha_{2k}\right)-\varepsilon\sum_{k=1}^{N}\left(\alpha_{1k}+\alpha_{2k}\right) \tag{5.16}$$
$$+\frac{1}{2}\sum_{k=1}^{N}\sum_{j=1}^{N}\left(\alpha_{1k}-\alpha_{2k}\right)\left(\alpha_{1j}-\alpha_{2j}\right)\left[K\left(\underline{x}_k,\underline{x}_j\right)+\frac{\delta\left(\underline{x}_k,\underline{x}_j\right)}{C}\right]$$

$s.a$

$$\sum_{k=1}^{N}\left(\alpha_{1k}-\alpha_{2k}\right)=0$$

$$\alpha_{1k}\geq 0$$

$$\alpha_{2k}\geq 0$$

$$k=1,2,...,N$$

No problema descrito na equação (5.16), $\delta\left(\underline{x}_k,\underline{x}_j\right):\mathbb{R}^{2N}\rightarrow\mathbb{R}$ representa a função delta de *Kronecker*, dada por:

$$\delta\left(\underline{x}_k,\underline{x}_j\right)=\begin{cases}1, \text{ se } \underline{x}_k=\underline{x}_j\\ 0, \text{ se } \underline{x}_k\neq\underline{x}_j\end{cases} \tag{5.17}$$

Ainda na equação (5.16), $K\left(\underline{x}_k,\underline{x}_j\right):\mathbb{R}^{2N}\rightarrow\mathbb{R}$ é o chamado núcleo do produto interno,

$$K\left(\underline{x}_k,\underline{x}_j\right)=\left[\underline{\phi}\left(\underline{x}_k\right)\right]^t\underline{\phi}\left(\underline{x}_j\right) \tag{5.18}$$

O núcleo do produto interno $K\left(\underline{x}_k,\underline{x}_j\right)$, também conhecido como *kernel*, deve ser definido segundo o teorema de *Mercer* [61], [62] e [177]. Dentre alguns exemplos de *kernels* $K\left(\underline{x}_k,\underline{x}_j\right)$, podem ser citadas as funções polinomiais, *gaussiana*s e sigmoidais, dadas pelas equações:

$$K\left(\underline{x}_k,\underline{x}_j\right)=\left\{\left[\underline{x}_k\right]^t\underline{x}_j+1\right\}^p \tag{5.19}$$

$$K\left(\underline{x}_k,\underline{x}_j\right)=e^{-\sigma^2\left\|\underline{x}_k-\underline{x}_j\right\|^2} \tag{5.20}$$

$$K\left(\underline{x}_k,\underline{x}_j\right)=\tanh\left\{\beta_0\left[\underline{x}_k\right]^t\underline{x}_j+\beta_1\right\} \tag{5.21}$$

Para os *kernels* descritos acima, $p\in\mathbb{R}$ e $\sigma\in\mathbb{R}^+$ são parâmetros definidos *a priori*. Vale ressaltar que, para as funções sigmoidais, as condições de *Mercer* são satisfeitas apenas para $\beta_0\in\mathbb{R}^+$ e $\beta_1\in\mathbb{R}^-$ [62]. Portanto, os MLPs e as redes de função de base radial também podem ser vistas como SVMs, porém com tipos específicos de *kernel*.

Utilizando a definição de *kernel* $K\left(\underline{x}_k, \underline{x}_j\right)$, e substituindo a equação (5.12) na equação (5.2), a saída de uma SVM passa a ser dada por:

$$f\left(\underline{x}, \underline{W}, b\right) = \sum_{k=1}^{N} \left(\alpha_{1k} - \alpha_{2k}\right) K\left(\underline{x}, \underline{x}_k\right) + b \tag{5.22}$$

A solução do problema de maximização descrito na equação (5.16) apresenta $\alpha_{1k} \neq \alpha_{2k}$ apenas para alguns vetores $\underline{x}_k$ integrantes do conjunto $D = \left\{\underline{x}_k, d_k\right\}$. Estes vetores são os chamados vetores suporte. Conceitualmente, estes padrões estão situados fora da banda definida por ε na Figura 5.6.

A equação (5.22) evidencia a forma intrínseca em que é realizado o mapeamento do espaço de representação original no espaço de características. A função $\underline{\phi}\left(\underline{x}\right)$ que define tal mapeamento não precisa ser diretamente especificada, sendo necessária apenas a definição do *kernel* $K\left(\underline{x}, \underline{x}_k\right)$, ou seja, o produto interno neste novo espaço. Esta equação também mostra que as SVMs podem ser entendidas como modelos *feedforward* com uma única camada escondida contendo neurônios definidos por $K\left(\underline{x}, \underline{x}_k\right)$. A Figura 5.7 ilustra a estrutura final de uma SVM, com S representando o número de vetores suporte.

A solução do problema dual apresentado na equação (5.16) produz estimativas para os parâmetros $\underline{\alpha}_1$ e $\underline{\alpha}_2$ que definem o modelo dado pela equação (5.22). Resta estimar o bias b. As condições de otimalidade de *Karush-Kuhn-Tucker* (KKT) afirmam que no ponto de sela da função *Lagrangeana* o produto entre as restrições e os respectivos multiplicadores de *Lagrange* deve ser nulo [178], [179]. Portanto, utilizando as equações (5.14) e (5.15), para $k = 1, 2, ..., N$,

$$\alpha_{1k}\left\{\underline{W}^t\,\underline{\phi}\left(\underline{x}_k\right)+b-d_k+\varepsilon+\xi_{1k}\right\}=0 \tag{5.23}$$
$$\alpha_{2k}\left\{d_k-\underline{W}^t\,\underline{\phi}\left(\underline{x}_k\right)-b+\varepsilon+\xi_{2k}\right\}=0$$

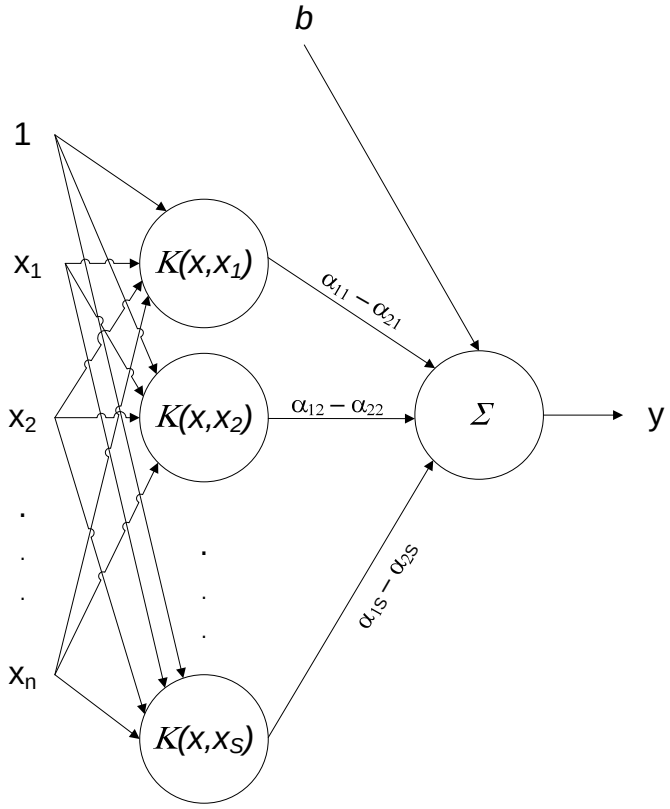

Figura 5.7 – Diagrama esquemático de uma SVM, com destaque para os vetores suporte

Algumas conclusões podem ser tiradas das equações (5.23). Primeiro, para $k=1,2,...,N$, $\alpha_{1k}\alpha_{2k}=0$, significando que nunca existirá um par de multiplicadores de *Lagrange* α_{1k} e α_{2k} simultaneamente nulos [62]. Além disso, excetuando os vetores suporte, localizados fora da banda definida pelo parâmetro ε, para os demais padrões ξ_{1k} e ξ_{2k} são nulos. Assim, para o *k*-ésimo vetor situado no interior da faixa especificada por ε e possuindo α_{1k} ou α_{2k} diferente de zero, da equação (5.23), as seguintes relações são obtidas:

$$\alpha_{1k} > 0 \Rightarrow b = d_k - \sum_{k=1}^{N} (\alpha_{1k} - \alpha_{2k}) K(\underline{x}_k, \underline{x}_k) - \varepsilon \tag{5.24}$$

$$\alpha_{2k} > 0 \Rightarrow b = d_k - \sum_{k=1}^{N} (\alpha_{1k} - \alpha_{2k}) K(\underline{x}_k, \underline{x}_k) + \varepsilon$$

Portanto, a partir de um padrão do conjunto de treinamento não classificado como vetor suporte, as equações (5.24) fornecem meios para estimativa do bias *b*.

O desenvolvimento das SVMs conduz a uma metodologia que une a escolha da estrutura e o treinamento de modelos *feedforward*, visto que o número de neurônios na camada oculta surge como subproduto do algoritmo de treinamento, através da definição dos vetores suporte. Além disso, ao utilizar o princípio da minimização do risco estrutural, o treinamento de SVMs inclui na sua essência uma parcela responsável pelo controle de complexidade do modelo, objetivando a estimação de estruturas com considerável capacidade de generalização.

Apesar destas características interessantes, as SVMs necessitam da especificação de alguns parâmetros, como as constantes C e ε, além da definição do tipo de *kernel* $K(\underline{x}, \underline{x}_k)$ utilizado, incluindo nesta escolha a estimação dos parâmetros que definem esta função. A prática comum encontrada na literatura utiliza validação cruzada para estimação destes parâmetros. Conforme mencionado na seção 2.2, técnicas de re-amostragem apresentam algumas restrições práticas, como esforço computacional elevado e aumento do requisito de dados. Além disso, para *kernels* apresentando diversos parâmetros a serem otimizados, a busca intensiva empregada pelos métodos de validação cruzada é proibitiva sob o ponto de vista de esforço computacional.

Utilizando o conceito de extensão dos vetores suporte, CHAPELLE e VAPNIK [180] desenvolveram limites superiores analíticos para o erro de generalização estimado por validação cruzada única, método popularmente conhecido como *leave-one-out*. Esta técnica fornece uma estimativa não tendenciosa da capacidade de generalização de

modelos treinados com $N-1$ padrões [62]. Portanto, a otimização de limites superiores de uma medida "quase" não-tendenciosa do desempenho do modelo para novos dados surge como alternativa à utilização de técnicas de validação cruzada para estimação dos parâmetros da SVM. Esta abordagem é utilizada neste trabalho para especificação das constantes C e ε, juntamente com os parâmetros que definem o *kernel gaussiano* $K(\underline{x},\underline{x}_k)$ utilizado, cuja análise dá origem a uma técnica de seleção de entradas. Detalhes teóricos e práticos sobre a metodologia serão apresentados a seguir.

5.1.1 Limites Superiores do Erro de Generalização de SVMs estimado por validação cruzada única (*leave-one-out*)

Validação cruzada única, do inglês *leave-one-out*, é uma técnica computacionalmente intensiva para estimação da capacidade de generalização. Dado um conjunto $D=\{\underline{x}_k,d_k\}$ contendo N pares entrada-saída, este método busca avaliar o desempenho para bases de dados de cardinalidade $N-1$, através da retirada, a cada iteração, de um padrão $(\underline{x}_k,d_k)$ constituinte do conjunto de treinamento. O desempenho do modelo treinado com este conjunto reduzido é avaliado através do erro para o par $(\underline{x}_k,d_k)$ excluído. Visando a utilização de todos os padrões para validação, este procedimento é repetido N vezes, com a capacidade de generalização $E_{LOO}\left[f(\underline{x},\underline{w})\right]$ do modelo $f(\underline{x},\underline{w})$ sendo estimada através da média aritmética entre os erros obtidos a cada iteração, ou seja,

$$E_{LOO}\left[f(\underline{x},\underline{w})\right]=\frac{1}{N}\sum_{k=1}^{N}L\left[d_k,f_k(\underline{x}_k,\underline{w})\right] \tag{5.25}$$

onde $L\left[d_k, f_k\left(\underline{x}, \underline{w}\right)\right]: \mathbb{R}^2 \to \mathbb{R}$ representa uma função de perda e $f_k\left(\underline{x}_k, \underline{w}\right)$ a saída gerada para o padrão $\underline{x}_k$ pelo modelo $f_k\left(\underline{x}, \underline{w}\right)$ estimado utilizando o conjunto reduzido oriundo da retirada do padrão $\left(\underline{x}_k, d_k\right)$.

Pode ser mostrado [62] que esta técnica produz uma estimativa não-tendenciosa do erro de generalização de modelos treinados utilizando bases de dados contendo $N-1$ padrões. Neste sentido, esta estimativa pode ser considerada "quase" não-tendenciosa para avaliação da capacidade de generalização de estruturas treinadas utilizando conjuntos de treinamento de cardinalidade N.

A necessidade de treinamento e avaliação de N estruturas eleva os requisitos computacionais da validação cruzada única, impossibilitando a sua aplicação prática. Entretanto, para SVMs, existem alguns limites superiores analíticos para $E_{LOO}\left[f\left(\underline{x}, \underline{W}, b\right)\right]$. Estes índices, de cálculo relativamente simples, evitam o esforço computacional elevado requerido pela técnica, podendo assim ser utilizados para seleção de SVMs, ou seja, especificação das constantes C e ε, além da definição do tipo de *kernel* $K\left(\underline{x}, \underline{x}_k\right)$ juntamente com os seus parâmetros.

Para problemas de classificação, podem ser definidos diversos limites, tais como o número de vetores suporte obtidos [182] e a relação entre o raio R da maior hiperesfera que engloba todos os padrões de treinamento e a margem de separação ρ [180]. Existem também índices baseados na análise dos multiplicadores de *Lagrange* $\underline{\alpha}_1$ e $\underline{\alpha}_2$, como o limite de *Jaakkola-Haussler* [62], outros fundamentados na física estatística, como o limite de *Opper-Winther* [62], e limites baseados em características geométricas dos mapeamentos implícitos realizados pelas SVMs no espaço de características, como a estatística baseada na extensão dos vetores suporte [180].

No contexto de regressão, existem somente duas estimativas, desenvolvidas em [183], baseadas respectivamente na relação raio/margem e na extensão dos vetores suporte. Como mostrado em [180], o índice baseado na extensão dos vetores suporte é limitado superiormente pela estimativa associada com a relação raio/margem. Tendo em vista a minimização de tais limites, é preferível utilizar aquele mais rigoroso, sendo por isso selecionado neste trabalho o índice baseado na extensão dos vetores suporte.

Para simplificar a descrição teórica, a idéia de extensão dos vetores suporte será apresentada somente para casos linearmente separáveis. Segundo [179], a penalização quadrática de erros no treinamento de SVMs (utilização da função quadrática de perda com tolerância ε para problemas de regressão) equivale à estimação destes modelos considerando os dados linearmente separáveis, porém utilizando um *kernel* $\widehat{K}\left(\underline{x}_i, \underline{x}_j\right): \mathbb{R}^{2N} \to \mathbb{R}$ modificado, dado por:

$$\widehat{K}\left(\underline{x}_i, \underline{x}_j\right) = K\left(\underline{x}_i, \underline{x}_j\right) + \frac{\delta\left(\underline{x}_i, \underline{x}_j\right)}{C} \tag{5.26}$$

Na equação acima, $K\left(\underline{x}_i, \underline{x}_j\right)$ é o núcleo do produto interno dado por (5.18), utilizado no treinamento de modelos considerando inseparabilidade linear entre os dados, e $\delta\left(\underline{x}_i, \underline{x}_j\right)$ representa a função delta de *Kronecker* apresentada na equação (5.17). Visto que para problemas linearmente separáveis a restrição nos multiplicadores de *Lagrange* $\underline{\alpha}_1$ e $\underline{\alpha}_2$ está limitada à sua positividade, a equação (5.16) evidencia a equivalência entre os problemas. Assim, visto que este trabalho utiliza a função quadrática dada pela equação (5.5), os resultados obtidos para problemas linearmente separáveis podem ser estendidos para os demais casos, com a constante C passando a ser vista como mais um parâmetro do *kernel* $\widehat{K}\left(\underline{x}_i, \underline{x}_j\right)$.

Visando originalmente a abordagem de problemas de classificação, CHAPELLE e VAPNIK [180] desenvolveram o conceito de extensão dos vetores suporte. Seja $\Omega \in \mathbb{R}^n \times \mathbb{R}^S$, $\Omega = \{\underline{x}_k \in D : \alpha_{1k} \neq \alpha_{2k}\}$, o conjunto não-vazio obtido ao final do treinamento de uma SVM contendo S vetores suporte. Além disso, seja $\Lambda_i \in \mathbb{R}^N$ o conjunto associado com o *i*-ésimo vetor suporte $\underline{x}_i \in \Omega$ e definido por combinações lineares restritas de pontos no espaço de características associados com os demais vetores suporte, ou seja,

$$\Lambda_i = \left\{ \underline{\lambda} \in \mathbb{R}^N : \sum_{j=1, j\neq i}^{S} \mu_j \underline{\phi}\left(\underline{x}_j\right) : \underline{x}_j \in \Omega, \sum_{j=1, j\neq i}^{p} \mu_j = 1 \right\} \tag{5.27}$$

Assim, a extensão $\Psi_i^2 \in \mathbb{R}^+$ do vetor suporte $\underline{x}_i \in \Omega$ é definida pela distância *euclidiana* entre a imagem $\underline{\phi}\left(\underline{x}_i\right)$ deste vetor no espaço de características e o conjunto Λ_i. Em outras palavras, a extensão dos vetores suporte, do inglês *span of support vectors*, é dada pela mínima distância *euclidiana* entre $\underline{\phi}\left(\underline{x}_i\right)$ e qualquer uma das combinações lineares possíveis em (5.27), podendo ser formulada da maneira que segue:

$$\Psi_i^2 = \min_{\underline{\mu}} \left\| \underline{\phi}\left(\underline{x}_i\right) - \sum_{j=1, j\neq i}^{S} \mu_j \underline{\phi}\left(\underline{x}_j\right) \right\|^2 \tag{5.28}$$

s.t.

$$\underline{x}_j \in \Omega, \sum_{j=1, j\neq i}^{S} \mu_j = 1$$

Conforme mencionado anteriormente, SVMs realizam um mapeamento implícito do espaço de características, não sendo conhecida a transformação $\underline{\phi}\left(\underline{x}\right) : \mathbb{R}^n \to \mathbb{R}^N$, mas sim o produto interno $K\left(\underline{x}_i, \underline{x}_j\right) : \mathbb{R}^{2N} \to \mathbb{R}$ neste novo espaço. Desta forma, a solução de (5.28) fica comprometida, visto que o mapeamento $\underline{\phi}\left(\underline{x}\right)$ não é especificado.

Para contornar este problema, seja $\underline{\underline{K_{VS}}} \in \mathbb{R}^S \times \mathbb{R}^S$ a matriz contendo o produto interno $K\left(\underline{x}_i, \underline{x}_j\right)$ entre todos os vetores suporte pertencentes a Ω, ou seja, $K_{VSij} = K\left(\underline{x}_i, \underline{x}_j\right)$, $\underline{x}_i$, $\underline{x}_j \in \Omega$. A matriz estendida $\underline{\underline{\widetilde{K}}} \in \mathbb{R}^{S+1} \times \mathbb{R}^{S+1}$ pode então ser definida da forma que segue:

$$\underline{\underline{\widetilde{K}}} = \begin{bmatrix} \underline{\underline{K_{VS}}} & \underline{u} \\ \underline{u}^t & 0 \end{bmatrix} \tag{5.29}$$

Na equação (5.29), $\underline{u} \in \mathbb{R}^S$ representa um vetor unitário. De posse desta matriz, a solução de (5.28) fornecendo a extensão Ψ_i^2 do *i*-ésimo vetor suporte pode ser escrita como [181]:

$$\Psi_i^2 = \frac{1}{\left(\underline{\underline{\widetilde{K}}}^{-1}\right)_{ii}} \tag{5.30}$$

Portanto, Ψ_i^2 é dado pelo recíproco do *i*-ésimo elemento da diagonal da inversa da matriz $\underline{\underline{\widetilde{K}}}$.

Seja $L\left(x, y\right): \mathbb{R}^2 \to \mathbb{R}$ na equação (5.25) a função modular de perda. Assim, o número total de erros $T_{LOO}\left[f\left(\underline{x}, \underline{w}\right)\right]$ cometidos durante o procedimento de validação cruzada única é dado por:

$$T_{LOO}\left[f\left(\underline{x}, \underline{w}\right)\right] = NE_{LOO}\left[f\left(\underline{x}, \underline{w}\right)\right] = \sum_{i=1}^{N} L\left[d_k, f_k\left(\underline{x}, \underline{w}\right)\right] \tag{5.31}$$

$$T_{LOO}\left[f\left(\underline{x}, \underline{w}\right)\right] = \sum_{i=1}^{N} \left|d_k - f_k\left(\underline{x}, \underline{w}\right)\right|$$

Supondo que o conjunto de vetores suporte Ω não seja alterado ao longo do procedimento de validação cruzada única, ou seja, a retirada de um padrão do conjunto

de treinamento não modifique este conjunto, $T_{LOO}\left[f\left(\underline{x},\underline{w}\right)\right]$ é limitado superiormente por uma estatística $T\left[f\left(\underline{x},\underline{W},b\right)\right]$, ou seja

$$T_{LOO}\left[f\left(\underline{x},\underline{w}\right)\right] \leq T\left[f\left(\underline{x},\underline{W},b\right)\right] \tag{5.32}$$

Para problemas de regressão, a estatística $T\left[f\left(\underline{x},\underline{W},b\right)\right]$ é dada por [183]:

$$T\left[f\left(\underline{x},\underline{W},b\right)\right] = \sum_{i=1}^{S}\left(\alpha_{1i}+\alpha_{2i}\right)\Psi_i^2 + N\varepsilon \tag{5.33}$$

Na equação (5.33), α_{1i} e α_{2i} são os multiplicadores de *Lagrange* associados ao vetor suporte $\underline{x}_i$, obtidos a partir da solução de (5.16), com ε representando a tolerância da função quadrática de perda e N respondendo pelo número de dados.

O limite $T\left[f\left(\underline{x},\underline{W},b\right)\right]$ mostra que a capacidade de generalização de SVMs está relacionada a propriedades geométricas mais complexas do que a margem de separação ρ. Conforme mostra a definição da extensão dos vetores suporte Ψ_i^2, uma estimativa "quase" não tendenciosa do desempenho para novos dados é limitada superiormente por um índice relacionado com a distribuição dos vetores suporte no espaço de características. Portanto, maximizar somente a margem ρ não garante boa capacidade de generalização, sendo necessária a obtenção de mapeamentos concentrados no espaço de características. Esta questão pode ser ressaltada pela equação (5.1), onde é mostrado o limite superior da dimensão VC de hiperplanos com margem de separação ρ. Esta equação mostra que, além de maximizar ρ, é necessário minimizar R, o raio da menor hiperesfera que contêm a imagem de todos os padrões no espaço de características, para redução do limite superior da respectiva dimensão VC. Portanto, além da margem, características geométricas do mapeamento $\underline{\phi}\left(\underline{x}\right): \mathbb{R}^n \rightarrow \mathbb{R}^N$ devem ser otimizadas visando elevar a capacidade de generalização. Como a constante

C pode ser entendida como mais um parâmetro do *kernel* quando utilizadas funções de erro quadráticas e a tolerância ε está diretamente relacionada com o conjunto de vetores suporte, a obtenção de propriedades ótimas para os mapeamentos gerados por $\underline{\phi}(\underline{x})$ constitui a principal motivação para a busca por valores adequados para os parâmetros que definem a SVM.

Visto que validação cruzada única faz uso de todos os dados para avaliação do modelo, ou seja, a cada iteração um padrão diferente do conjunto de treinamento é retirado, o pressuposto de manutenção do conjunto de vetores suporte ao longo de todo o procedimento não é verificado. Apesar desta violação teórica, a otimização do limite $T\left[f\left(\underline{x},\underline{W},b\right)\right]$ para seleção dos parâmetros que definem as SVMs vem sendo utilizada tanto em problemas de classificação [180], [181] quanto de regressão [183], [184], mostrando ser uma medida mais rigorosa na prática, no sentido de obter limites superiores mais estreitos para o erro de generalização [62].

O limite $T\left[f\left(\underline{x},\underline{W},b\right)\right]$ pode ser minimizado através de dois procedimentos básicos. Através da definição de um conjunto de valores para os parâmetros a serem estimados, uma busca exaustiva pode ser realizada, sendo escolhido o conjunto que apresentar o menor valor para $T\left[f\left(\underline{x},\underline{W},b\right)\right]$, de maneira análoga aos procedimentos de validação. Apesar de fácil implementação, esta abordagem torna o problema intratável em tempo prático para buscas em espaços de dimensionalidade considerável, diante do elevado número de combinações a serem testadas, impossibilitando a utilização de *kernels* mais elaborados do que os apresentados nas equações (5.19), (5.20) e (5.21).

Visando reduzir o esforço computacional requerido para buscas em espaços de elevada cardinalidade, técnicas direcionadas de otimização, como descida em gradiente,

podem ser utilizadas. Apesar dos conhecidos problemas relacionados a mínimos locais, a característica orientada destes procedimentos possibilita a aplicação deste conjunto de métodos a problemas de busca em espaço de dimensionalidade considerável.

A aplicação direta de descida em gradiente para minimização de $T\left[f\left(\underline{x},\underline{\underline{W}},b\right)\right]$ dado pela equação (5.33) fica comprometida pela característica descontínua deste índice. Este traço indesejado de $T\left[f\left(\underline{x},\underline{\underline{W}},b\right)\right]$ está relacionado com a alteração do conjunto de vetores suporte em virtude da modificação dos parâmetros que definem a SVM, conforme mostrado em [181]. Esta questão pode ser contornada através da inserção de um funcional regularizador no cálculo da extensão Ψ_i^2 na equação (5.28), dando origem ao problema de minimização que define a extensão diferenciável $\widetilde{\Psi}_i^2$ do vetor suporte $\underline{x}_i$, dada por [181]:

$$\Psi_i^2 = \min_{\underline{\mu}} \left\| \underline{\phi}\left(\underline{x}_i\right) - \sum_{j=1, j\neq i}^{S} \mu_j \underline{\phi}\left(\underline{x}_j\right) \right\|^2 + \eta \sum_{j=1, j\neq i}^{S} \frac{1}{\left(\alpha_{1i} + \alpha_{2i}\right)} \mu_j^2 \tag{5.34}$$

s.t.

$$\underline{x}_j \in \Omega, \sum_{j=1, j\neq i}^{S} \mu_j = 1$$

Em (5.34), $\eta \in \mathbb{R}^+$ é uma constante definida pelo usuário, responsável pela diferenciabilidade de $\widetilde{\Psi}_i^2$. Nesta nova definição, de maneira análoga ao desenvolvimento para solução de (5.28), pode ser obtida uma estimativa para $\widetilde{\Psi}_i^2$ utilizando o *kernel* $K\left(\underline{x}_i, \underline{x}_j\right)$ através da seguinte expressão:

$$\widetilde{\Psi}_i^2 = \frac{1}{\left[\left(\underline{\underline{\widetilde{K}}} + \underline{\underline{D}}\right)^{-1}\right]_{ii}} - \underline{\underline{D}}_{ii} \tag{5.35}$$

Na equação acima, $\underline{\underline{D}}_{ii}$ representa o *i*-ésimo elemento da diagonal da matriz $\underline{\underline{D}} \in \mathbb{R}^{S+1} \times \mathbb{R}^{S+1}$ dada por:

$$\underline{\underline{D}} = \eta \begin{bmatrix} (\alpha_{11} + \alpha_{22})^{-1} & 0 & \cdots & 0 \\ 0 & \ddots & \cdots & 0 \\ \vdots & \vdots & (\alpha_{1S} + \alpha_{2S})^{-1} & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{5.36}$$

Observando as equações (5.34) e (5.35), anulando o parâmetro η, a extensão Ψ_i^2 dos vetores suporte é recuperada da sua versão $\widetilde{\Psi}_i^2$ diferenciável. A especificação de um valor adequado para η é um problema em aberto, fugindo do escopo deste trabalho. Seguindo as recomendações de [181] e [183], esta constante é feita igual a 0.1.

Substituindo a extensão diferenciável $\widetilde{\Psi}_i^2$ dos vetores suporte em (5.33), é obtida a estatística $\widetilde{T}\left[f\left(\underline{x}, \underline{\underline{W}}, b\right)\right]$ que limita superiormente o número total de erros cometidos durante o procedimento de validação cruzada única, dada por:

$$\widetilde{T}\left[f\left(\underline{x}, \underline{\underline{W}}, b\right)\right] = \sum_{i=1}^{S} (\alpha_{1i} + \alpha_{2i}) \widetilde{\Psi}_i^2 + N\varepsilon \tag{5.37}$$

Para o cálculo do gradiente $\nabla \widetilde{T}\left[f\left(\underline{x}, \underline{\underline{W}}, b\right)\right]$, seja $\underline{\theta}$ o vetor contendo os parâmetros a serem otimizados, a saber, as p variáveis que definem o *kernel*, a constante de regularização C e a tolerância ε da função de perda, perfazendo um total de $p+2$ parâmetros a serem especificados. Da equação (5.37), a derivada parcial de $\widetilde{T}\left[f\left(\underline{x}, \underline{\underline{W}}, b\right)\right]$ em relação à *k*-ésima variável θ_k é dada por:

$$\frac{\partial}{\partial \theta_k} \widetilde{T}\left[f\left(\underline{x}, \underline{\underline{W}}, b\right)\right] = \sum_{i=1}^{S} \frac{\partial}{\partial \theta_k} (\alpha_{1i} + \alpha_{2i}) \widetilde{\Psi}_i^2 + \sum_{i=1}^{S} (\alpha_{1i} + \alpha_{2i}) \frac{\partial}{\partial \theta_k} \widetilde{\Psi}_i^2 + \frac{\partial}{\partial \theta_k} (N\varepsilon) \tag{5.38}$$

Portanto, para o cálculo das derivadas parciais de $\widetilde{T}\left[f\left(\underline{x}, \underline{\underline{W}}, b\right)\right]$, é necessária a obtenção das derivadas parciais da soma entre os multiplicadores de *Lagrange* $(\alpha_{1i} + \alpha_{2i})$ e da extensão $\widetilde{\Psi}_i^2$ do vetor suporte $\underline{x}_i$. Das condições de KKT dadas pela equação (5.23), para θ_k igual a ε, é obtida a seguinte relação [183]:

$$\begin{bmatrix} \dfrac{\partial}{\partial\theta_k}\left(\underline{\alpha}_1-\underline{\alpha}_2\right) \\ \dfrac{\partial}{\partial\theta_k}b \end{bmatrix} = \left[\underline{\underline{\widetilde{K}}}^{-1}\right]\begin{bmatrix} \dfrac{\partial}{\partial\theta_k}\underline{g} \\ 0 \end{bmatrix} \tag{5.39}$$

Em (5.39), $\underline{g}\in\mathbb{R}^S$ é um vetor cuja definição e derivada em relação a θ_k são dadas por:

$$\underline{g} = \begin{cases} d_i-\varepsilon, \text{ se } \left(\alpha_{1i}-\alpha_{2i}\right)>0 \\ d_i+\varepsilon, \text{ se } \left(\alpha_{1i}-\alpha_{2i}\right)<0 \end{cases} \tag{5.40}$$

$$\frac{\partial}{\partial\theta_k}\underline{g} = \begin{cases} -1, \text{ se } \left(\alpha_{1i}-\alpha_{2i}\right)>0 \\ 1, \text{ se } \left(\alpha_{1i}-\alpha_{2i}\right)<0 \end{cases}$$

De forma análoga, para θ_k representando as demais variáveis são obtidas as seguintes expressões:

$$\begin{bmatrix} \dfrac{\partial}{\partial\theta_k}\left(\underline{\alpha}_1-\underline{\alpha}_2\right) \\ \dfrac{\partial}{\partial\theta_k}b \end{bmatrix} = \left[\underline{\underline{\widetilde{K}}}^{-1}\right]\begin{bmatrix} \dfrac{\partial}{\partial\theta_k}\underline{g} \\ 0 \end{bmatrix} \tag{5.41}$$

De posse de (5.40) e (5.41), a derivada parcial de $\left(\alpha_{1i}+\alpha_{2i}\right)$ em relação ao *k*-ésimo parâmetro θ_k pode ser obtida através da seguinte relação [183]:

$$\frac{\partial}{\partial\theta_k}\left(\alpha_{1i}+\alpha_{2i}\right) = \begin{cases} \dfrac{\partial}{\partial\theta_k}\left(\alpha_{1i}-\alpha_{2i}\right), \text{ se } \left(\alpha_{1i}-\alpha_{2i}\right)>0 \\ -\dfrac{\partial}{\partial\theta_k}\left(\alpha_{1i}-\alpha_{2i}\right), \text{ se } \left(\alpha_{1i}-\alpha_{2i}\right)<0 \end{cases} \tag{5.42}$$

Para obtenção das derivadas parciais da extensão do vetor suporte $\widetilde{\Psi}_i^2$ em relação a cada parâmetro θ_k, utilizando a equação (5.35),

$$\frac{\partial}{\partial\theta_k}\widetilde{\Psi}_i^2 = -\frac{1}{\left[\left(\underline{\underline{\widetilde{K}}}+\underline{\underline{D}}\right)^{-1}\right]_{ii}^2}\frac{\partial}{\partial\theta_k}\left[\left(\underline{\underline{\widetilde{K}}}+\underline{\underline{D}}\right)^{-1}\right]_{ii} + \frac{\eta}{\left(\alpha_{1i}+\alpha_{2i}\right)^2}\frac{\partial}{\partial\theta_k}\left(\alpha_{1i}+\alpha_{2i}\right) \tag{5.43}$$

O uso de propriedades matriciais permite calcular a derivada parcial de $\left[\left(\underline{\underline{\widetilde{K}}}+\underline{\underline{D}}\right)^{-1}\right]_{ii}$, dada por:

$$\frac{\partial}{\partial\theta_k}\left[\left(\underline{\underline{\widetilde{K}}}+\underline{\underline{D}}\right)^{-1}\right]_{ii}=\left\{\left(\underline{\underline{\widetilde{K}}}+\underline{\underline{D}}\right)^{-1}\left[\frac{\partial}{\partial\theta_k}\underline{\underline{\widetilde{K}}}+\frac{\partial}{\partial\theta_k}\underline{\underline{D}}\right]\left(\underline{\underline{\widetilde{K}}}+\underline{\underline{D}}\right)^{-1}\right\}_{ii} \tag{5.44}$$

A derivada em relação à matriz $\underline{\underline{D}}$ necessária em (5.44) é obtida utilizando a sua definição apresentada na equação (5.36), resultando na expressão:

$$\frac{\partial}{\partial\theta_k}\underline{\underline{D}}=-\eta\begin{bmatrix}\left(\alpha_{11}+\alpha_{21}\right)^{-2}\dfrac{\partial}{\partial\theta_k}\left(\alpha_{11}+\alpha_{21}\right) & 0 & \cdots & 0\\ 0 & \ddots & \cdots & 0\\ \vdots & \vdots & \left(\alpha_{1S}+\alpha_{2S}\right)^{-2}\dfrac{\partial}{\partial\theta_k}\left(\alpha_{1S}+\alpha_{2S}\right) & \vdots\\ 0 & 0 & \cdots & 0\end{bmatrix} \tag{5.45}$$

Resta definir a derivada em relação à matriz $\underline{\underline{\widetilde{K}}}$, relacionada com o produto interno entre os vetores suporte e dada pela equação (5.29). Para isto, é necessária a definição do tipo de *kernel* $K\left(\underline{x}_i,\underline{x}_j\right)$. Diversas funções podem ser utilizadas, como mostram os exemplos das expressões (5.19) a (5.21), com as condições dadas no teorema de *Mercer* devendo ser atendidas. Visando desenvolver um método automático para seleção de entradas, neste trabalho é utilizado um novo tipo de função, baseada no *kernel gaussiano* dado pela equação (5.20), porém com parâmetros independentes para cada entrada. Este *kernel* $K\left(\underline{x}_i,\underline{x}_j\right)$ modificado é dado por:

$$K\left(\underline{x}_i,\underline{x}_j\right)=e^{-\sum_{l=1}^{n}\sigma_l^2\left(x_{il}-x_{jl}\right)^2} \tag{5.46}$$

Nesta nova função, a contribuição de cada entrada no cálculo da saída é diretamente ponderada pelo parâmetro σ_l associado. Esta questão pode ser entendida se a equação (5.46) for escrita da seguinte forma:

$$K\left(\underline{x}_i,\underline{x}_j\right)=e^{-\sum_{l=1}^{n}\left(\sigma_l x_{il}-\sigma_l x_{jl}\right)^2} \tag{5.47}$$

Assim, tendo em mente a equação (5.22), é esperado que variáveis relevantes para o cálculo da saída $f\left(\underline{x},\underline{W},b\right)$ apresentem elevado σ_l, com aquelas menos determinantes para este cálculo possuindo valor reduzido para o respectivo ponderador. Desta forma, a análise dos valores otimizados para o conjunto de ponderadores $\underline{\sigma}=\left[\sigma_i \quad ... \quad \sigma_n\right]^t$ permite a definição de um procedimento automático de seleção de entradas, o qual será descrito posteriormente.

Assim, a derivada do *kernel gaussiano* modificado $K\left(\underline{x}_i,\underline{x}_j\right)$ em relação a cada ponderador σ_l é dada por:

$$\frac{\partial}{\partial\sigma_l}K\left(\underline{x}_i,\underline{x}_j\right)=-2\sigma_l\left(x_{il}-x_{jl}\right)^2 e^{-\sum_{l=1}^{n}\sigma_l^2\left(x_{il}-x_{jl}\right)^2}=-2\sigma_l\left(x_{il}-x_{jl}\right)^2 K\left(\underline{x}_i,\underline{x}_j\right) \tag{5.48}$$

Utilizando o *kernel* $K\left(\underline{x}_i,\underline{x}_j\right)$ dado por (5.47), a derivada da matriz $\underline{\underline{\widetilde{K}}}$, definida na equação (5.29), em relação ao *k*-ésimo parâmetro θ_k é dada por:

$$\frac{\partial}{\partial\theta_k}\underline{\underline{\widetilde{K}}}=\begin{bmatrix}\dfrac{\partial}{\partial\theta_k}\underline{\underline{K_{VS}}} & \underline{o}\\ \underline{o}^t & 0\end{bmatrix} \tag{5.49}$$

Nesta equação, $\underline{o}\in\mathbb{R}^S$ é um vetor nulo, com $\partial\underline{\underline{K_{VS}}}\big/\partial\theta_k$ representando a derivada de $K\left(\underline{x}_i,\underline{x}_j\right)$ calculada para todos os vetores suporte, ou seja, para todo $\underline{x}_i$, $\underline{x}_j\in\Omega$,

$$\frac{\partial K_{VS_{ij}}}{\partial\theta_k}=\frac{\partial K\left(\underline{x}_i,\underline{x}_j\right)}{\partial\theta_k} \tag{5.50}$$

Conforme mencionado anteriormente, ao utilizar a função quadrática de perda com tolerância ε, a constante de regularização C pode ser vista como mais um

parâmetro do *kernel* [179], como mostrado na equação (5.26). Desta forma, para θ_k representando C, a derivada parcial de $K_{VS_{ij}}$ é dada por:

$$\frac{\partial K_{VS_{ij}}}{\partial \theta_k} = -\frac{\delta_{ij}}{C^2} \tag{5.51}$$

Como a tolerância ε da função de perda não está relacionada com $K\left(\underline{x}_i, \underline{x}_j\right)$, a derivada de $\widetilde{K}_{ij}$ em relação a este parâmetro é nula para todo i e j.

De posse de (5.49), é possível calcular as componentes do gradiente $\nabla\widetilde{T}\left[f\left(\underline{x}, \underline{W}, b\right)\right]$. Seja $\underline{\theta}$ o vetor contendo $n+2$ elementos positivos, com os n primeiros relacionados com os parâmetros σ_i de $K\left(\underline{x}_i, \underline{x}_j\right)$, e os dois últimos respondendo pela constante de regularização C e pela tolerância ε, respectivamente. Portanto, utilizando a expressão (5.38) em conjunto com as derivadas parciais dadas nas equações (5.42) e (5.43), cada uma das $n+1$ primeiras componentes de $\nabla\widetilde{T}\left[f\left(\underline{x}, \underline{W}, b\right)\right]$ podem ser obtidas através da seguinte expressão:

$$\frac{\partial}{\partial \theta_k}\widetilde{T}\left[f\left(\underline{x}, \underline{W}, b\right)\right] = \sum_{i=1}^{S}\frac{\partial}{\partial \theta_k}\left(\alpha_{1i} + \alpha_{2i}\right)\widetilde{\Psi}_i^2 + \sum_{i=1}^{S}\left(\alpha_{1i} + \alpha_{2i}\right)\frac{\partial}{\partial \theta_k}\widetilde{\Psi}_i^2 \tag{5.52}$$

A estimativa para a última coordenada de $\nabla\widetilde{T}\left[f\left(\underline{x}, \underline{W}, b\right)\right]$, relacionada com ε, é dada por:

$$\frac{\partial}{\partial \theta_k}\widetilde{T}\left[f\left(\underline{x}, \underline{W}, b\right)\right] = \sum_{i=1}^{S}\frac{\partial}{\partial \theta_k}\left(\alpha_{1i} + \alpha_{2i}\right)\widetilde{\Psi}_i^2 + \sum_{i=1}^{S}\left(\alpha_{1i} + \alpha_{2i}\right)\frac{\partial}{\partial \theta_k}\widetilde{\Psi}_i^2 + N \tag{5.53}$$

As expressões (5.52) e (5.53) podem ser utilizadas em um algoritmo iterativo de descida em gradiente para minimização de $\widetilde{T}\left[f\left(\underline{x}, \underline{W}, b\right)\right]$. Entretanto, a aplicação direta destas equações apresenta alguns empecilhos. Primeiramente, todos os parâmetros a serem otimizados são números reais positivos. Para evitar a necessidade de

inserir esta restrição no problema de minimização de $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$, uma nova formulação pode ser escrita [181], [183], visando à otimização do logaritmo natural dos parâmetros que definem a SVM, ou seja, $\log\left(\underline{\sigma}\right)$, $\log\left(C\right)$ e $\log\left(\varepsilon\right)$. Desta forma, para o *k*-ésimo parâmetro θ_k, a derivada em relação ao $\log\left(\theta_k\right)$ é dada por:

$$\frac{\partial}{\partial \log\left(\theta_k\right)}\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]=\theta_k\frac{\partial}{\partial \theta_k}\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right] \tag{5.54}$$

Além da restrição de positividade, a derivação do limite $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ parte da premissa da existência de vetores suporte. Em outras palavras, este índice não está definido para modelos com todos os padrões situados dentro da banda especificada pela tolerância ε. Desta forma, além de positivo, o parâmetro ε não pode apresentar valores muito elevados, sob pena de dar origem a modelos sem vetores suporte, ou seja, com todos os padrões no interior da faixa definida por ε. Visando evitar esta situação e supondo que os padrões de saída sejam normalizados no intervalo $\left[a,b\right]\in\mathbb{R}$, o valor máximo $\varepsilon_{\max}$ admitido para este parâmetro é dado por:

$$\varepsilon_{\max}=c\frac{b-a}{2} \tag{5.55}$$

Para $\varepsilon=0.5\left(b-a\right)$, todos os padrões estarão dentro da banda definida por esta variável, o que não é desejado. Este fato explica o ponderador c apresentado em (5.55). Neste trabalho, este fator c é feito igual a 0.8.

O impacto diferenciado de cada parâmetro no comportamento de $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ compromete a otimização conjunta de todas as variáveis. Conforme mostra a equação (5.37), o limite $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$, além de estar diretamente relacionado com a tolerância ε, depende do conjunto de vetores suporte. Visto que estes vetores são definidos pelos padrões do conjunto de treinamento situados fora da banda especificada por ε e que a

constante C pondera somente os erros para padrões desta natureza, conforme apresentado na equação (5.10), variações nestes parâmetros podem conduzir a modificações no conjunto de vetores suporte. Esta é a razão da descontinuidade do limite dado pela equação (5.33). Desta forma, na otimização conjunta de $\log(\underline{\sigma})$, $\log(C)$ e $\log(\varepsilon)$, a variação nas duas últimas grandezas é mais crítica no sentido de ocasionar maiores variações em $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$. Esta questão foi verificada na prática através de testes realizados ao longo do desenvolvimento deste trabalho.

Como a análise dos valores otimizados de $\underline{\sigma}$ será utilizada para seleção de variáveis de entrada, a limitação da busca no espaço definido por estes parâmetros pode comprometer a técnica de avaliação da relevância dos sinais de entrada. Além disso, a extensão $\tilde{\Psi}_i^2$ é baseada na minimização da distância *euclidiana* entre pontos definidos no espaço de características induzido por $K\left(\underline{x}_i,\underline{x}_j\right)$. Este fato destaca a importância da escolha adequada do mapeamento $\underline{\phi}\left(\underline{x}_i\right)$ através da definição do *kernel*, realçando a necessidade da busca por valores ótimos para $\underline{\sigma}$. Desta forma, a minimização de $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ será desacoplada, sendo realizada em dois estágios baseados em descida em gradiente. No primeiro estágio serão otimizados somente os ponderadores $\underline{\sigma}$ utilizando as n primeiras componentes de $\nabla\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$, com os demais parâmetros mantidos constantes (parâmetro de regularização C e tolerância ε). Posteriormente, fazendo uso das duas últimas coordenadas de $\nabla\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ e utilizando os parâmetros $\underline{\sigma}$ previamente otimizados, é realizada a busca baseada em descida em gradiente de C e ε. Esse processo iterativo é repetido até o critério de convergência ser atingido.

As questões acima relatadas mostram a característica multimodal de $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$. Lembrando que este índice é oriundo de um limite descontínuo, os aspectos levantados acima evidenciam a existência de múltiplos mínimos locais nesta função. Este fato reforça a necessidade de adaptação do método de descida em gradiente, visto que a aplicação direta desta técnica pode conduzir a soluções de baixa qualidade em virtude da convergência prematura para mínimos locais.

Apesar do desacoplamento utilizado neste trabalho não garantir a convergência para pontos mais qualificados, este método de minimização mostrou ser mais efetivo ao longo dos testes efetuados. Esta questão pode ser explicada pela ampliação do espaço de busca em virtude da utilização de direções distintas em cada estágio, permitindo a avaliação de regiões do espaço que não seriam visitadas seguindo a direção estrita de $\nabla\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$.

Como todo método baseado em descida em gradiente, a técnica proposta também apresenta dependência em relação ao ponto inicial do processo iterativo. Desta forma, é desejada a escolha deste ponto em uma região onde esteja situado um mínimo local de considerável qualidade. Tendo este objetivo em mente, a constante de regularização C e a tolerância ε da função de perda são iniciadas utilizando as expressões recomendadas em [185] e dadas por:

$$C_0 = \max\left(\left|\overline{d} + 3\delta_d\right|, \left|\overline{d} - 3\delta_d\right|\right) \tag{5.56}$$

$$\varepsilon_0 = 3\sqrt{\frac{\ln N}{N\left(N-n\right)}\sum_{i=1}^{N}\left(d_i - \hat{y}_i\right)^2}$$

Na equação (5.56), $\overline{d}$ e δ_d são respectivamente a média e o desvio padrão amostral das saídas d_k, com $\hat{y}_i$ representando a saída gerada pelo modelo $f\left(\underline{x},\underline{W},b\right)$ quando alimentado pelo padrão $\underline{x}_i$. Visto que $f\left(\underline{x},\underline{W},b\right)$ não é definido *a priori*, neste ponto

$\hat{y}_i$ é estimado através de um modelo de regressão linear que utiliza as mesmas entradas apresentadas à $f\left(\underline{x},\underline{W},b\right)$. As estatísticas $\overline{d}$ e δ_d são dadas por:

$$\overline{d} = \frac{1}{N}\sum_{k=1}^{N} d_k \tag{5.57}$$

$$\delta_d = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}\left(d_k - \overline{d}\right)^2}$$

Para definição do ponto inicial para $\underline{\sigma}$, ainda não são disponíveis na literatura expressões com valores recomendados. Em [183], utilizando o *kernel gaussiano* tradicional dado pela equação (5.10), onde $\sigma_l = \sigma$ para todas as entradas, os autores partem de valores pequenos para $\underline{\sigma}$, especificamente 0.5, convergindo para valores que variam entre 0.01 a 0.27, dependendo do caso em estudo. A busca por estes valores é feita através da minimização de $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ via descida em gradiente. Na referência [184], é recomendada de forma empírica a utilização de um vetor unitário como ponto inicial para a busca por valores ótimos para $\underline{\sigma}$, sendo utilizado um *kernel gaussiano* $K\left(\underline{x}_i,\underline{x}_j\right)$ similar ao apresentado na equação (5.46). Contudo, essa função apresenta mais um parâmetro, podendo ser escrita da seguinte forma:

$$K\left(\underline{x}_i,\underline{x}_j\right) = e^{-\frac{1}{2\kappa^2}\sum_{l=1}^{n}\sigma_l^2\left(x_{il}-x_{jl}\right)^2} = e^{-\sum_{l=1}^{n}\left(\frac{\sigma_l}{\sqrt{2}\kappa}x_{il}-\frac{\sigma_l}{\sqrt{2}\kappa}x_{jl}\right)^2} \tag{5.58}$$

O parâmetro adicional κ na equação (5.58) é feito igual a 10. Portanto, ao utilizar o *kernel* $K\left(\underline{x}_i,\underline{x}_j\right)$ dado na equação (5.46), as componentes de $\underline{\sigma}$ devem ser feitas iguais a $0.1/\sqrt{2}$ segundo as recomendações de [184]. Este valor de pequena magnitude vai de encontro aos resultados obtidos em [183]. Portanto, apesar da ausência de expressões para $\underline{\sigma}$, a experiência mostra que a busca por valores ótimos para estes parâmetros está situada em regiões onde estes apresentam pequena magnitude. Seguindo tal

conhecimento, neste trabalho será definido como ponto inicial para $\underline{\sigma} \in \mathbb{R}^n$ um vetor com todas as suas componentes apresentando pequeno valor absoluto, mais especificamente igual a 0.1.

A minimização de $\tilde{T}\left[f\left(\underline{x}, \underline{W}, b\right)\right]$ busca determinar os parâmetros da SVM visando obter modelos com elevada capacidade de generalização. Além de C e ε, os parâmetros $\underline{\sigma}$ que definem o *kernel* também são otimizados. A análise dos ponderadores $\underline{\sigma}$ dá origem a um procedimento automático de mensuração da relevância de cada entrada, conforme apresentado na próxima seção.

5.1.2 Seleção de entradas de SVMs

O *kernel gaussiano* modificado $K\left(\underline{x}_i, \underline{x}_j\right)$ dado pela equação (5.46) é definido pelo vetor de parâmetros $\underline{\sigma} \in \mathbb{R}^n$. Conforme mostrado na equação (5.47), estes parâmetros podem ser vistos como ponderadores das entradas, cuja análise pode ser utilizada para mensuração da relevância de cada variável no cálculo da saída dada pela equação (5.22). Entretanto, de forma análoga à determinação automática de relevância (ARD) apresentada na seção 4.2, uma referência de irrelevância deve ser determinada. Em outras palavras, seguindo a terminologia definida na seção 2.1.2, é necessária a definição de um limiar σ_0 a partir do qual o sinal de entrada pode ser considerado irrelevante no cálculo da saída.

O método para definição empírica de limiares de relevância apresentado na seção 3.2 será utilizado também para SVMs. Neste contexto, direto da equação (2.20), a estatística $\tilde{T}\left[f\left(\underline{x}, \underline{W}, b\right)\right]$ representa a medida de desempenho, com $\underline{\sigma}$ respondendo pelos ponderadores reais de cada entrada. Especificamente, a minimização de $\tilde{T}\left[f\left(\underline{x}, \underline{W}, b\right)\right]$ é aplicada ao conjunto de treinamento estendido, criado a partir da

incorporação do sinal de prova ao conjunto de entradas original. Ao final do treinamento, o patamar de relevância σ_0 é determinado através do ponderador σ_l associado à variável de prova. As entradas do conjunto estendido são separadas segundo a sua natureza (contínua ou discreta), sendo então ordenadas de forma crescente segundo a magnitude dos parâmetros $\underline{\sigma}$ estimados. Após a ordenação, são descartadas as variáveis situadas abaixo do respectivo sinal de prova, ou seja, apresentando σ_l menor que σ_0. Neste contexto, são descartadas as variáveis que contribuem menos para o cálculo da saída do que o sinal de prova, guardando analogia com a determinação automática de relevância. Mantendo analogia com o ARD, a inserção de variáveis de prova deve seguir a natureza das entradas utilizadas. Desta forma, devem ser especificados dois patamares, um para as entradas contínuas, denominado σ_C, e outro para as variáveis discretas, denotado por σ_D.

A inserção de variáveis auxiliares para determinação empírica do limiar de irrelevância σ_0 permite a análise dos valores de $\underline{\sigma}$ visando à retirada de entradas irrelevantes do modelo final de previsão. A forma na qual esta variável é gerada demonstra a ausência de relação desta com a saída. Portanto, é esperado que os ponderadores σ_l associados aos sinais de prova apresentem pequena magnitude quando comparados aos ponderadores das demais entradas.

Este conhecimento pode ser utilizado para definição do ponto inicial do algoritmo de descida em gradiente para os ponderadores σ_l relacionados com os sinais auxiliares. Partindo do pressuposto que todas as entradas originalmente selecionadas pelo usuário são relevantes, os ponderadores destes sinais são igualados a 0.1 inicialmente. Portanto, para as variáveis de prova sabidamente não relacionadas com a

saída, estes parâmetros devem apresentar menor magnitude, sendo feitos iguais a 0.01 no início do algoritmo de otimização.

Definido o procedimento de seleção de entradas através da análise dos parâmetros $\underline{\sigma} \in \mathbb{R}^n$, o método automático de especificação e treinamento de SVMs pode ser resumido. Esta tarefa cabe à próxima seção, onde este procedimento é descrito na forma de um algoritmo, sendo listados todos os passos do procedimento.

5.1.3 Método automático de especificação e treinamento de SVMs

Após a descrição do procedimento para minimização de $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ juntamente com o método para seleção de entradas, o algoritmo de especificação e treinamento automático de SVMs pode ser resumido da maneira que segue:

1. Faça $l = 0$.
2. Insira variáveis de prova ao conjunto de entradas original seguindo as distribuições de probabilidade apresentadas nas equações (3.36) e (3.37). Se as variáveis de entrada forem somente contínuas, insira somente o sinal de prova desta natureza. Do contrário, insira também a variável de prova discreta.
3. Selecione o ponto inicial para os parâmetros $\log\underline{\sigma}(l)$, $\log C(l)$ e $\log\varepsilon(l)$ da SVM.
4. Utilizando $\underline{\sigma}(l)$, $C(l)$ e $\varepsilon(l)$, calcule os multiplicadores de *Lagrange* $\underline{\alpha}_1$, $\underline{\alpha}_2$ através da solução de (5.16).
5. Minimize $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ em relação a $\log\underline{\sigma}(l)$, utilizando descida em gradiente neste espaço de busca, obtendo $\log\underline{\sigma}(l+1)$. O gradiente neste espaço é dado pelas equações (5.52) e (5.54).

6. Minimize $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ em relação a $\log C(l)$ e $\log \varepsilon(l)$, utilizando descida em gradiente neste espaço de busca, obtendo $\log C(l+1)$ e $\log \varepsilon(l+1)$. O gradiente neste espaço é dado pelas equações (5.52), (5.53) e (5.54).
7. Verifique a restrição em $\varepsilon(l+1)$ dada por (5.55). Se $\varepsilon(l+1) > \varepsilon_{\max}$, faça $\varepsilon(l+1) = \varepsilon_{\max}$. Do contrário, mantenha $\varepsilon(l+1)$.
8. Se o critério de convergência for alcançado, vá para o passo 9. Senão, faça $l = l+1$ e retorne ao passo 4.
9. Agrupe os parâmetros σ_l associados a entradas contínuas e discretas em vetores distintos e ordene de forma crescente estes vetores.
10. Em cada vetor, selecione as variáveis de entrada situadas acima do respectivo limiar de relevância, ou seja, $\sigma_l > \sigma_C$ para variáveis contínuas e $\sigma_l > \sigma_D$ para as discretas.
11. Utilizando somente as entradas relevantes selecionadas no passo 10, juntamente com os respectivos parâmetros otimizados $\underline{\sigma}$, C e ε, obtenha o modelo final, calculando os multiplicadores de *Lagrange* $\underline{\alpha}_1$, $\underline{\alpha}_2$ através da solução de (5.16) e realize as previsões.

No algoritmo listado acima, os passos 5 e 6 utilizam descida em gradiente em diferentes espaços. Especificamente, seja $\nabla_{\underline{a}}\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ o vetor constituindo pelos componentes do gradiente $\nabla\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ associados ao vetor $\underline{a}$. Portanto, no passo 5, o processo de otimização pode ser representado pela seguinte equação:

$$\log \underline{\sigma}(l+1) = \log \underline{\sigma}(l) - \upsilon \nabla_{\log \underline{\sigma}} \tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]_{\underline{\sigma}=\underline{\sigma}(l),C=C(l),\varepsilon=\varepsilon(l)} \tag{5.59}$$

Em (5.59), υ representa o passo do algoritmo, definido por busca em linha e inicialmente feito igual a 0.001. A ordem de grandeza deste passo é reduzida até 10^{-7},

visando obter o passo máximo para o qual a estatística $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ é decrementada, ou seja, $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]_{\underline{\sigma}=\underline{\sigma}(l+1),C=C(l),\varepsilon=\varepsilon(l)} < \tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]_{\underline{\sigma}=\underline{\sigma}(l),C=C(l),\varepsilon=\varepsilon(l)}$. De forma análoga, para o passo 6, considerando $\psi(l)=\left[\log C(l) \quad \log\varepsilon(l)\right]^t$,

$$\psi(l+1)=\psi(l)-\vartheta\nabla_{\psi}\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]_{\underline{\sigma}=\underline{\sigma}(l+1),C=C(l),\varepsilon=\varepsilon(l)} \tag{5.60}$$

Na equação (5.60), ϑ representa o passo, iniciado em 10^{-6} e reduzido de forma análoga ao procedimento descrito para o passo 5.

Para convergência do algoritmo, são utilizados dois critérios. O primeiro é baseado no número máximo de iterações, feito igual a 100 e raramente atingido. O segundo está relacionado com a variação mínima na estatística $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ entre duas iterações consecutivas, especificada em 10^{-5}.

A minimização de $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ através de um algoritmo baseado em gradiente permite otimizar os parâmetros que definem a SVM, ou seja, a constante de regularização C, a tolerância ε da função de perda e os ponderadores $\underline{\sigma}\in\mathbb{R}^n$ do *kernel* modificado $K\left(\underline{x}_i,\underline{x}_j\right)$ dado pela equação (5.46). Através da otimização de propriedades geométricas dos mapeamentos induzidos por $K\left(\underline{x}_i,\underline{x}_j\right)$ mais complexas do que a margem ρ, esta busca visa à obtenção de modelos com elevada capacidade de generalização.

A análise dos parâmetros $\underline{\sigma}\in\mathbb{R}^n$ obtidos ao final do processo de otimização de $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ permite classificar as variáveis de entrada segundo a sua relevância para o cálculo da saída. Visando retirar aquelas que podem ser consideradas irrelevantes, sinais aleatórios de prova são inseridos para estimação empírica de limiares de irrelevância σ_0.

Diante da multimodalidade intrínseca do limite $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$, originário de uma estatística descontínua, o processo de otimização é desmembrado, visando a fuga de mínimos locais de baixa qualidade através da ampliação da busca em direções alternativas a cada estágio. Como todo método baseado em gradiente, a técnica utilizada depende das condições iniciais. A escolha do ponto de partida para o processo iterativo segue expressões recomendadas em [185] para os parâmetros C e ε. Para os ponderadores $\underline{\sigma} \in \mathbb{R}^n$, seguindo a recomendação da literatura, tais valores devem apresentar pequena magnitude. Para as variáveis originalmente definidas pelo usuário esses parâmetros são iniciados como 0.1. Os ponderadores relacionados às variáveis de prova são inicialmente feitos iguais a 0.01.

As dificuldades encontradas na especificação dos parâmetros que definem a SVM constituem um dos empecilhos na aplicação destes modelos. Conforme mencionado acima, a característica multi-modal da estatística $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ é um dos complicadores do processo de minimização. As máquinas de vetores relevantes (RVMs), também baseadas em *kernel*, possuem metodologias para estimação dos seus respectivos parâmetros. Esta questão motiva a utilização destes modelos no trabalho, sendo apresentados em detalhes na próxima seção.

5.2 Máquinas de vetores relevantes (RVMs)

As máquinas de vetor suporte apresentadas na seção anterior, apesar de populares principalmente no contexto de classificação, apresentam algumas limitações. Além da necessidade de técnicas para estimação dos diversos hiperparâmetros (ε, C e $\underline{\sigma}$), os *kernels* $K\left(\underline{x},\underline{y}\right)$ devem atender as condições do teorema de *Mercer*, restringindo a classe de funções que podem ser utilizadas. Por último, as previsões

pontuais geradas por esses modelos comprometem a definição de intervalos de confiança.

Propostas originalmente por TIPPING [63], as máquinas de vetores relevantes (RVMs) são modelos probabilísticos baseados em *kernel* que possuem algumas das vantajosas características das SVMs, como, por exemplo, representação esparsa. De forma análoga à SVM, somente alguns pontos do conjunto de treinamento contribuem para a estimação da superfície de regressão, vetores esses denominados relevantes. Esta nomenclatura foi adotada em função da similaridade da técnica com a determinação automática de relevância (ARD) apresentada na seção 4.2.

Dado um conjunto de dados $D = \left\{ \underline{\underline{X}}, \underline{Y} \right\}$, $\underline{\underline{X}} \in \mathbb{R}^N \times \mathbb{R}^n$, $\underline{\underline{X}} = \left[\underline{x}_1 \quad \underline{x}_2 \quad \ldots \quad \underline{x}_N \right]^t$, $\underline{x}_k \in \mathbb{R}^n$, $\underline{x}_k = \left[x_{k1}, ..., x_{kn} \right]^t$, $\underline{Y} \in \mathbb{R}^N$, $\underline{Y} = \left[d_1 \quad d_2 \quad \ldots \quad d_N \right]^t$, $d_k \in \mathbb{R}$, seja a formulação probabilística tradicional considerando ruído aditivo $\zeta_k \in \mathbb{R}$ na saída desejada, isto é, $d_k = F\left(\underline{x}_k\right) + \zeta_k$. Para modelar $F\left(\underline{x}\right) : \mathbb{R}^n \to \mathbb{R}$, seja a função aproximativa $f\left(\underline{x}, \underline{w}\right) : \mathbb{R}^n \to \mathbb{R}$ formada pela combinação linear de funções de base $\Phi\left(\underline{x}, \underline{z}\right) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ centradas em cada ponto do conjunto D, gerando a saída dada por:

$$f\left(\underline{x}, \underline{W}\right) = \sum_{i=1}^{N} w_i \Phi\left(\underline{x}, \underline{x}_i\right) + b = \left[\underline{\Phi}\left(\underline{x}\right) \right]^t \underline{W} \tag{5.61}$$

Em (5.61), $\underline{w} \in \mathbb{R}^N$, $\underline{w} = \left[w_1 \quad w_2 \quad \ldots \quad w_N \right]^t$, $b \in \mathbb{R}$, $\underline{W} \in \mathbb{R}^{N+1}$, $\underline{W} = \left[b \quad \underline{w}^t \right]^t$, com $\underline{\Phi}\left(\underline{x}\right) : \mathbb{R}^n \to \mathbb{R}^{N+1}$ representando as funções de base $\Phi\left(\underline{x}, \underline{x}_i\right) = \Phi_i\left(\underline{x}\right)$ avaliadas para o ponto $\underline{x}$ e centradas em cada ponto do conjunto de dados D, incluindo uma parcela constante responsável pelo bias, isto é,

$$\underline{\Phi}(\underline{x}) = \begin{bmatrix} 1 & \Phi(\underline{x},\underline{x}_1) & \dots & \Phi(\underline{x},\underline{x}_N) \end{bmatrix}^t = \begin{bmatrix} \Phi_0(\underline{x}) & \Phi_1(\underline{x}) & \dots & \Phi_N(\underline{x}) \end{bmatrix}^t \tag{5.62}$$

Para estimação do conjunto de parâmetros $\underline{W}$, inferência *bayesiana* de forma análoga à apresentada no capítulo 4 para MLPs pode ser aplicada. Observando a regra de *Bayes* dada pela equação (4.1), para obtenção da probabilidade a *posteriori* $p(\underline{W}|\underline{Y})$, é necessária a definição da verossimilhança $p(\underline{Y}|\underline{W})$ e da probabilidade a *priori* $p(\underline{W})$. Supondo que as amostras do ruído ζ_k sejam geradas de forma independente a partir de uma distribuição *gaussiana* com média nula e variância $\sigma^2 \in \mathbb{R}$, a função de verossimilhança $p(\underline{Y}|\underline{W})$ é dada por:

$$p(\underline{Y}|\underline{W},\sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left(-\frac{1}{2\sigma^2} \left\| \underline{Y} - \underline{\underline{\Phi}}\,\underline{W} \right\|^2 \right) \tag{5.63}$$

Em (5.63), $\underline{\underline{\Phi}} \in \mathbb{R}^N \times \mathbb{R}^{N+1}$ é a chamada matriz de modelagem, ou seja,

$$\underline{\underline{\Phi}} = \begin{bmatrix} \Phi_0(\underline{x}_1) & \Phi_1(\underline{x}_1) & \dots & \Phi_N(\underline{x}_1) \\ \Phi_0(\underline{x}_2) & \Phi_1(\underline{x}_2) & \dots & \Phi_N(\underline{x}_1) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_0(\underline{x}_N) & \Phi_1(\underline{x}_N) & \dots & \Phi_N(\underline{x}_1) \end{bmatrix} = \begin{bmatrix} \underline{\Phi}_0 & \underline{\Phi}_1 & \dots & \underline{\Phi}_N \end{bmatrix} \tag{5.64}$$

Em (5.64), $\underline{\Phi}_i \in \mathbb{R}^N$ é o vetor contendo a saída da *i*-ésima função de base para cada ponto do conjunto D. Para a probabilidade a *priori* $p(\underline{W})$, seja o produto de distribuições *gaussianas* dado por [63]:

$$p(\underline{W}|\underline{\alpha}) = \prod_{i=1}^{N+1} \frac{1}{\sqrt{2\pi\alpha_i^{-1}}} \exp\left(-\frac{1}{2\alpha_i^{-1}} W_i^2 \right) \tag{5.65}$$

Em (5.65), são consideradas distribuições *gaussianas* distintas, todas com média nula e variância dada pelo inverso dos respectivos hiperparâmetros $\alpha_i \in \mathbb{R}^+$. Conforme será apresentado a seguir, $\underline{\alpha} \in (\mathbb{R}^+)^{N+1}$, $\underline{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{N+1} \end{bmatrix}^t$ responde pelo conjunto de

hiperparâmetros que controlam a magnitude de cada parâmetro W_i analogamente ao desenvolvimento para mensuração de relevância das entradas apresentado na seção 4.2.

Diante do surgimento dos hiperparâmetros σ^2 e $\underline{\alpha}$, antes do cálculo da probabilidade a *posteriori* $p\left(\underline{W}\middle|\underline{Y}\right)$ é necessária a definição de probabilidades a *priori* para estas variáveis também desconhecidas. Seguindo a abordagem original apresentada em [63], especificações não-informativas serão consideradas para estes hiperparâmetros, por meio das distribuições dadas por:

$$p\left(\underline{\alpha}\right)=\prod_{i=1}^{N+1}G\left(\alpha_i^{-1}\middle|a,b\right) \tag{5.66}$$
$$p\left(\sigma^2\right)=G\left(\sigma^{-2}\middle|c,d\right)$$

Em (5.66) $G\left(x\middle|a,b\right):\mathbb{R}\to\mathbb{R}$ é a distribuição *gama* definida pela expressão:

$$G\left(x\middle|a,b\right)=\frac{b^a x^{a-1}e^{-bx}}{\int_0^\infty t^{a-1}e^{-t}dt} \tag{5.67}$$

Para que $p\left(\underline{\alpha}\right)$ e $p\left(\sigma^2\right)$ sejam não-informativas, as constantes a, b, c e d devem assumir valores de pequena magnitude. Para o caso em que estes parâmetros são nulos, $p\left(\underline{\alpha}\right)$ e $p\left(\sigma^2\right)$ convergem para distribuições uniformes na escala logarítmica, tornando as estimativas dadas em (5.61) independentes em relação a transformações lineares tanto das saídas desejadas quanto das saídas das funções de base $\Phi\left(\underline{x},\underline{z}\right)$. Em outras palavras, as previsões realizadas pelo modelo $f\left(\underline{x},\underline{w}\right)$ passam a ser independentes da escala na qual as saídas desejadas são medidas [63]. Por este motivo e por simplicidade de exposição, esta suposição será considerada neste trabalho. O desenvolvimento considerando valores quaisquer para os hiperparâmetros a, b, c e d pode ser encontrado em [63].

Definidas de forma hierárquica as diferentes distribuições *a priori*, a probabilidade *a posteriori* $p\left(\underline{W},\underline{\alpha},\sigma^2\middle|\underline{Y}\right)$ das grandezas desconhecidas $\underline{W}$, $\underline{\alpha}$ e σ^2 pode ser calculada pela regra de *Bayes*, dada por (4.1) e neste contexto expressa pela seguinte equação:

$$p\left(\underline{W},\underline{\alpha},\sigma^2\middle|\underline{Y}\right)=\frac{p\left(\underline{Y}\middle|\underline{W},\underline{\alpha},\sigma^2\right)p\left(\underline{W},\underline{\alpha},\sigma^2\right)}{p\left(\underline{Y}\right)} \tag{5.68}$$

Para realização de previsões para um novo ponto de teste $\underline{x}_{N+1}$, uma abordagem puramente *bayesiana* deve visar à estimativa da distribuição de probabilidade *a posteriori* $p\left(d_{N+1}\middle|\underline{x}_{N+1},\underline{Y},\underline{\underline{X}}\right)$ da saída a ser prevista d_{N+1} relacionada com o padrão de entrada $\underline{x}_{N+1}$. Analogamente ao discutido na seção 4.1, esta distribuição é obtida pela marginalização de $p\left(d_{N+1}\middle|\underline{x}_{N+1},\underline{\theta}\right)$ sobre todo o espaço de parâmetros desconhecidos $\underline{\theta}=\left[\underline{W}^t \quad \underline{\alpha}^t \quad \sigma^2\right]^t$ através da integral dada pela equação (4.3). Aqui, esta equação é dada por:

$$p\left(d_{N+1}\middle|\underline{Y}\right)=\iint\ldots\int p\left(d_{N+1}\middle|\underline{W},\underline{\alpha},\sigma^2\right)p\left(\underline{W},\underline{\alpha},\sigma^2\middle|\underline{Y}\right)dW_1\ldots dW_{N+1}d\alpha_1\ldots d\alpha_{N+1}d\sigma^2 \tag{5.69}$$

Em (5.69), as dependências em relação à $\underline{x}_{N+1}$ e X foram retiradas por simplicidade de notação, visto que a distribuição dos padrões de entrada não é modelada. Da mesma forma que para o treinamento de MLPs, neste contexto o cálculo analítico da equação (5.69) também é impraticável para quaisquer $p\left(d_{N+1}\middle|\underline{W},\underline{\alpha},\sigma^2\right)$ e $p\left(\underline{W},\underline{\alpha},\sigma^2\middle|\underline{Y}\right)$, sendo necessárias algumas aproximações.

A probabilidade a posteriori $p\left(\underline{W},\underline{\alpha},\sigma^2\middle|\underline{Y}\right)$ pode ser reescrita da forma que segue:

$$p\left(\underline{W},\underline{\alpha},\sigma^2\middle|\underline{Y}\right)=p\left(\underline{W}\middle|\underline{Y},\underline{\alpha},\sigma^2\right)p\left(\underline{\alpha},\sigma^2\middle|\underline{Y}\right) \tag{5.70}$$

A probabilidade $p\left(\underline{W}\middle|\underline{Y},\underline{\alpha},\sigma^2\right)$ de $\underline{W}$ dados os hiperparâmetros $\underline{\alpha}$ e σ^2, juntamente com o conjunto de saídas-alvo $\underline{Y}$, pode ser obtida pela regra de *Bayes* através da seguinte expressão:

$$p\left(\underline{W}\middle|\underline{Y},\underline{\alpha},\sigma^2\right)=\frac{p\left(\underline{Y}\middle|\underline{W},\underline{\alpha},\sigma^2\right)p\left(\underline{W}\middle|\underline{\alpha},\sigma^2\right)}{p\left(\underline{Y}\middle|\underline{\alpha},\sigma^2\right)}=\frac{p\left(\underline{Y}\middle|\underline{W},\sigma^2\right)p\left(\underline{W}\middle|\underline{\alpha}\right)}{p\left(\underline{Y}\middle|\underline{\alpha},\sigma^2\right)} \tag{5.71}$$

A segunda igualdade em (5.71) segue da independência entre a verossimilhança $p\left(\underline{Y}\middle|\underline{W},\sigma^2\right)$ e o hiperparâmetro $\underline{\alpha}$, relacionado somente com a probabilidade a *priori* $p\left(\underline{W}\middle|\underline{\alpha}\right)$ que por sua vez não está associada com σ^2. Em conjunto com as expressões (5.63) e (5.65), para o cálculo de (5.71) resta obter o fator de normalização $p\left(\underline{Y}\middle|\underline{\alpha},\sigma^2\right)$, através da integral dada por:

$$p\left(\underline{Y}\middle|\underline{\alpha},\sigma^2\right)=\iint\ldots\int p\left(\underline{Y}\middle|\underline{W},\sigma^2\right)p\left(\underline{W}\middle|\underline{\alpha}\right)dW_1\ldots dW_{N+1} \tag{5.72}$$

Como $p\left(\underline{Y}\middle|\underline{W},\sigma^2\right)$ e $p\left(\underline{W}\middle|\underline{\alpha}\right)$ são *gaussianas*, a distribuição $p\left(\underline{Y}\middle|\underline{\alpha},\sigma^2\right)$ pode ser estimada diretamente por meio da convolução entre estas distribuições, sendo dada por:

$$p\left(\underline{Y}\middle|\underline{\alpha},\sigma^2\right)=\frac{1}{\left(2\pi\right)^{\frac{N}{2}}\left|\underline{\underline{C}}\right|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}\underline{Y}^t\underline{\underline{C}}^{-1}\underline{Y}\right) \tag{5.73}$$

$$\underline{\underline{C}}=\sigma^2\underline{\underline{I}}+\underline{\underline{\Phi}}\,\underline{\underline{A}}^{-1}\underline{\underline{\Phi}}^t$$

Em (5.73), $\underline{\underline{I}}\in\mathbb{R}^N\times\mathbb{R}^N$ é a matriz identidade, com $\underline{\underline{A}}\in\mathbb{R}^{N+1}\times\mathbb{R}^{N+1}$ respondendo pela matriz dada por:

$$\underline{\underline{A}}=\begin{bmatrix}\alpha_1 & 0 & \ldots & 0\\ 0 & \alpha_2 & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \ldots & \alpha_{N+1}\end{bmatrix} \tag{5.74}$$

Assim, a distribuição também *gaussiana* $p\left(\underline{W}\middle|\underline{Y},\underline{\alpha},\sigma^2\right)$ é dada por:

$$p\left(\underline{W}\middle|\underline{Y},\underline{\alpha},\sigma^2\right)=\frac{1}{\left(2\pi\right)^{\frac{N+1}{2}}\left|\underline{\underline{\Sigma}}\right|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}\left(\underline{W}-\underline{\mu}\right)^t\underline{\underline{\Sigma}}^{-1}\left(\underline{W}-\underline{\mu}\right)\right] \tag{5.75}$$

Na expressão acima, a matriz de covariância $\underline{\underline{\Sigma}}\in\mathbb{R}^{N+1}\times\mathbb{R}^{N+1}$ e o vetor média $\underline{\mu}\in\mathbb{R}^{N+1}$ são estimados da seguinte forma:

$$\begin{aligned}\underline{\underline{\Sigma}}&=\left(\sigma^2\underline{\underline{\Phi}}^t\underline{\underline{\Phi}}+\underline{\underline{A}}\right)^{-1}\\ \underline{\mu}&=\sigma^{-2}\underline{\underline{\Sigma}}\,\underline{\underline{\Phi}}^t\underline{Y}\end{aligned} \tag{5.76}$$

Obtida a distribuição $p\left(\underline{W}\middle|\underline{Y},\underline{\alpha},\sigma^2\right)$, para o cálculo de (5.70) resta estimar a probabilidade $p\left(\underline{\alpha},\sigma^2\middle|\underline{Y}\right)$ dos hiperparâmetros $\underline{\alpha}$ e σ^2 dado o conjunto de saídas desejadas $\underline{Y}$. Mantendo a analogia com a maximização da evidência para MLPs apresentada na seção 4.1, TIPPING [63] parte do pressuposto que $p\left(\underline{\alpha},\sigma^2\middle|\underline{Y}\right)$ apresenta pouca dispersão em torno dos valores mais prováveis $\underline{\alpha}^{MP}$ e σ^{MP}. Uma abordagem alternativa, baseada em inferência variacional para aproximação de $p\left(\underline{W},\underline{\alpha},\sigma^2\middle|\underline{Y}\right)$, também pode ser utilizada, conforme proposto em [186]. Além de computacionalmente intensiva, esta metodologia na prática produz valores esperados para os hiperparâmetros iguais aos obtidos considerando a aproximação proposta em [63]. Assim neste trabalho será admitido que $p\left(\underline{\alpha},\sigma^2\middle|\underline{Y}\right)$ apresenta um pico em torno dos valores mais prováveis $\underline{\alpha}^{MP}$ e σ^{MP}, sendo necessária a estimação destes parâmetros.

Novamente, para obtenção da probabilidade *a posteriori* $p\left(\underline{\alpha},\sigma^2\middle|\underline{Y}\right)$ a regra de *Bayes* pode ser utilizada, resultando na seguinte expressão:

$$p\left(\underline{\alpha},\sigma^2\middle|\underline{Y}\right)=\frac{p\left(\underline{Y}\middle|\underline{\alpha},\sigma^2\right)p\left(\underline{\alpha},\sigma^2\right)}{p\left(\underline{Y}\right)}=\frac{p\left(\underline{Y}\middle|\underline{\alpha},\sigma^2\right)p\left(\underline{\alpha}\right)p\left(\sigma^2\right)}{p\left(\underline{Y}\right)} \tag{5.77}$$

Como $p\left(\underline{\alpha}\right)$ e $p\left(\sigma^2\right)$ são não-informativas e $p\left(\underline{Y}\right)$ é um fator de normalização, os valores mais prováveis $\underline{\alpha}^{MP}$ e σ^{MP} à luz dos dados podem ser estimados a partir da maximização de $p\left(\underline{Y}\middle|\underline{\alpha},\sigma^2\right)$, dada pela equação (5.73). Em modelagem *bayesiana*, a otimização desta quantidade, conhecida como verossimilhança marginalizada, é conhecida como método-II de maximização da verossimilhança, do inglês *type-II maximum likelihood*, sendo também denominada maximização da evidência no contexto de sistemas inteligentes.

O logaritmo natural de $p\left(\underline{Y}\middle|\underline{\alpha},\sigma^2\right)$ descrito em (5.73) é dado por:

$$\log p\left(\underline{Y}\middle|\underline{\alpha},\sigma^2\right)=L\left(\underline{\alpha},\sigma^2\right)=-\frac{1}{2}\left(N\log 2\pi+\log\left|\underline{\underline{C}}\right|+\underline{Y}^t\,\underline{\underline{C}}^{-1}\,\underline{Y}\right) \tag{5.78}$$

Seguindo a maximização da evidência proposta por MACKAY [54], a otimização de $L\left(\underline{\alpha},\sigma^2\right)$ é realizada através de um algoritmo iterativo, cujas expressões são obtidas a partir da anulação do gradiente de $L\left(\underline{\alpha},\sigma^2\right)$ e dadas por:

$$\gamma_i\left(l\right)=1-\alpha_i\left(l\right)\Sigma_{ii}\left(l\right) \tag{5.79}$$

$$\alpha_i\left(l+1\right)=\frac{\gamma_i\left(l\right)}{\mu_i^2\left(l\right)} \tag{5.80}$$

$$\sigma^2\left(l+1\right)=\frac{\left\|\underline{Y}-\underline{\underline{\Phi}}\,\underline{\mu}\left(l\right)\right\|^2}{N-\sum_{i=1}^{N+1}\gamma_i\left(l\right)} \tag{5.81}$$

Nas expressões acima, $\Sigma_{ii}\left(l\right)$ representa o *i*-ésimo elemento da diagonal da matriz de covariância $\underline{\underline{\Sigma}}\left(l\right)$ estimada na *l*-ésima iteração, com $\mu_i\left(l\right)$ respondendo pelo *i*-ésimo componente do vetor média $\underline{\mu}\left(l\right)$. A matriz $\underline{\underline{\Sigma}}\left(l\right)$ e o vetor $\underline{\mu}\left(l\right)$ são estimados

utilizando as equações (5.76) considerando os respectivos hiperparâmetros $\underline{\alpha}(l)$ e $\sigma^2(l)$. Além disso, como existe uma distribuição *a priori* para cada parâmetro W_i, $\gamma_i(l)$ pode ser entendido como uma medida da determinação de W_i à luz dos dados, análoga ao número efetivo de parâmetros obtido na seção 4.1.

As equações (5.79) a (5.81) em conjunto com as expressões (5.76) podem ser utilizadas em um algoritmo iterativo para estimação dos hiperparâmetros $\underline{\alpha}^{MP}$ e σ^{MP}, possibilitando a estimativa dos parâmetros mais prováveis *a posteriori* $\underline{\mu}^{MP}$ e $\underline{\underline{\Sigma}}^{MP}$. De posse destas quantidades, para realização de previsões, a probabilidade $p\left(d_{N+1}|\underline{Y},\underline{\alpha}^{MP},\sigma^{MP}\right)$ dada em (5.69), agora condicionada a $\underline{\alpha}^{MP}$ e σ^{MP}, passa a ser dada por:

$$p\left(d_{N+1}|\underline{Y},\underline{\alpha}^{MP},\sigma^{MP}\right)=\iint\ldots\int p\left(d_{N+1}|\underline{W},\sigma^{MP}\right)p\left(\underline{W}|\underline{Y},\underline{\alpha}^{MP},\sigma^{MP}\right)dW_1\ldots dW_{N+1} \tag{5.82}$$

Como $p\left(d_{N+1}|\underline{W},\sigma^{MP}\right)$, dado pela equação (5.63), e $p\left(\underline{W}|\underline{Y},\underline{\alpha}^{MP},\sigma^{MP}\right)$, expresso em (5.75), são distribuições *gaussianas*, $p\left(d_{N+1}|\underline{Y},\underline{\alpha}^{MP},\sigma^{MP}\right)$ também apresenta esta característica, sendo dada por:

$$p\left(d_{N+1}|\underline{Y},\underline{\alpha}^{MP},\sigma^{MP}\right)=\frac{1}{\left(2\pi\hat{\sigma}^2\right)^{\frac{1}{2}}}\exp\left[-\frac{1}{2\sigma_{N+1}^2}\left(d_{N+1}-\hat{d}_{n+1}\right)^2\right] \tag{5.83}$$

O valor esperado $\hat{d}_{N+1}$ e a variância $\hat{\sigma}^2$ da estimativa da saída desejada d_{N+1} associada a um novo ponto de teste $\underline{x}_{N+1}$ são obtidos através das expressões:

$$\begin{aligned}\hat{d}_{N+1}&=f\left(\underline{x}_{N+1},\underline{\mu}^{MP}\right)=\left[\underline{\Phi}\left(\underline{x}_{N+1}\right)\right]^t\underline{\mu}^{MP}\\ \hat{\sigma}^2&=\left(\sigma^{MP}\right)^2+\left[\underline{\Phi}\left(\underline{x}_{N+1}\right)\right]^t\underline{\underline{\Sigma}}^{MP}\underline{\Phi}\left(\underline{x}_{N+1}\right)\end{aligned} \tag{5.84}$$

Em (5.84), $\underline{\mu}^{MP}$ e $\underline{\underline{\Sigma}}^{MP}$ são obtidos a partir da avaliação das expressões (5.76) para $\underline{\alpha}^{MP}$ e σ^{MP}. Desta maneira, a saída estimada pelo modelo é dada pela expressão (5.61) avaliada em $\underline{x}_{N+1}$, considerando o conjunto mais provável de parâmetros $\underline{\mu}^{MP}$. A variância $\hat{\sigma}^2$, utilizada para definição de intervalos de confiança, apresenta duas componentes, uma relacionada com a estimativa da variância do ruído presente nos dados e outra associada com a incerteza no cálculo de $\underline{\mu}^{MP}$.

Desta forma, o algoritmo de estimação de máquinas de vetores relevantes baseado na maximização da evidência pode ser resumido da maneira que segue

1. Faça $l = 0$ e inicialize o conjunto de hiperparâmetros, ou seja, $\sigma(l)$ e $\underline{\alpha}(l) = \left[\alpha_1(l) \quad \ldots \quad \alpha_{N+1}(l)\right]^t$.
2. Calcule os parâmetros $\underline{\mu}(l) = \left[\mu_1(l) \quad \ldots \quad \mu_{N+1}(l)\right]^t$ e $\underline{\underline{\Sigma}}(l)$ utilizando (5.76).
3. Faça $l = l + 1$ e atualize os hiperparâmetros $\underline{\alpha}(l)$ e $\sigma(l)$ utilizando as equações (5.79) a (5.81).
4. Se o critério de parada for atendido, encerre o algoritmo. Do contrário, retorne ao passo 2.

Considerando as distribuições *a priori* não-informativas dadas pelas equações (5.66) com parâmetros a, b, c e d nulos, a utilização iterativa das expressões (5.76) e (5.80) conduz a soluções apresentando hiperparâmetros α_i^{MP} de elevada magnitude, tendendo a infinito. Isto significa que a distribuição *a posteriori* $p\left(\underline{W}\middle|\underline{Y}, \underline{\alpha}, \sigma^2\right)$ dada em (5.75) possui um pico em $W_i = 0$, evidenciando a baixa relevância da função de base $\Phi\left(\underline{x}, \underline{x}_i\right)$ no cálculo da saída. As funções que apresentam $\alpha_i^{MP} = \infty$ podem ser retiradas do modelo final, gerando representações esparsas semelhantes às obtidas para as

máquinas de vetor suporte. Os vetores $\underline{x}_i$ associados às funções de base $\Phi\left(\underline{x},\underline{x}_i\right)$ remanescentes no modelo final são denominados vetores relevantes.

A análise do funcional $L\left(\underline{\alpha},\sigma^2\right)$ dado pela equação (5.78) permite justificar de forma analítica a representação esparsa característica das RVMs [187]. Para tal, a matriz $\underline{\underline{C}}$ dada em (5.73) deve ser escrita de forma alternativa, visando obter a influência específica de um dado hiperparâmetro α_i em $L\left(\underline{\alpha},\sigma^2\right)$, passando a ser dada por:

$$\underline{\underline{C}} = \sigma^2 \underline{\underline{I}} + \sum_{j=1}^{N+1} \alpha_j^{-1} \underline{\Phi}_j \underline{\Phi}_j^t = \sigma^2 \underline{\underline{I}} + \sum_{j=1, j\neq i}^{N+1} \alpha_j^{-1} \underline{\Phi}_j \underline{\Phi}_j^t + \alpha_i^{-1} \underline{\Phi}_i \underline{\Phi}_i^t \tag{5.85}$$

$$\underline{\underline{C}} = \underline{\underline{C}}_{-i} + \alpha_i^{-1} \underline{\Phi}_i \underline{\Phi}_i^t$$

Na expressão acima, $\underline{\underline{C}}_{-i} \in \mathbb{R}^N \times \mathbb{R}^N$ é a matriz de covariância com a influência da *i*-ésima função de base removida, com $\underline{\Phi}_i$ definido em (5.64) respondendo pela avaliação da *i*-ésima função de base em cada ponto do conjunto de dados D, ou seja,

$$\underline{\underline{C}}_{-i} = \sigma^2 \underline{\underline{I}} + \sum_{j=1, j\neq i}^{N+1} \alpha_j^{-1} \underline{\Phi}_j \underline{\Phi}_j^t \tag{5.86}$$

Assim, $L\left(\underline{\alpha},\sigma^2\right)$ pode ser decomposto da forma que segue:

$$L\left(\underline{\alpha},\sigma^2\right) = L\left(\underline{\alpha}_{-i},\sigma^2\right) + l\left(\alpha_i,\sigma^2\right) \tag{5.87}$$

$$L\left(\underline{\alpha}_{-i},\sigma^2\right) = -\frac{1}{2}\left[N \log 2\pi + \log\left|\underline{\underline{C}}_{-i}\right| + \underline{Y}^t \underline{\underline{C}}_{-i}^{-1} \underline{Y}\right]$$

$$l\left(\alpha_i,\sigma^2\right) = \frac{1}{2}\left[\log \alpha_i - \log\left(\alpha_i + \underline{\Phi}_i^t \underline{\underline{C}}_{-i}^{-1} \underline{\Phi}_i\right) + \frac{\left(\underline{\Phi}_i^t \underline{\underline{C}}_{-i}^{-1} \underline{Y}\right)^2}{\alpha_i + \underline{\Phi}_i^t \underline{\underline{C}}_{-i}^{-1} \underline{\Phi}_i}\right]$$

Em (5.87), $L\left(\underline{\alpha}_{-i},\sigma^2\right)$ é o logaritmo da verossimilhança para o modelo reduzido desconsiderando a função de base $\underline{\Phi}_i$, ou seja, $\alpha_i^{MP} = \infty$ e consequentemente $W_i = 0$. Desta forma, a contribuição de $\underline{\Phi}_i$ para o cálculo de $L\left(\underline{\alpha},\sigma^2\right)$ é avaliada pela expressão $l\left(\alpha_i,\sigma^2\right)$.

Desmembrada a influência no cálculo de $L\left(\underline{\alpha},\sigma^2\right)$, o gradiente em relação a α_i passa a ser dado por [187]:

$$\frac{\partial}{\partial\alpha_i}L\left(\underline{\alpha},\sigma^2\right)=\frac{\partial}{\partial\alpha_i}l\left(\alpha_i,\sigma^2\right)=\frac{\alpha_i^{-1}s_i^2-\left(q_i^2-s_i\right)}{2\left(\alpha_i+s_i\right)^2} \tag{5.88}$$

Em (5.88), $s_i\in\mathbb{R}^+$ e $q_i\in\mathbb{R}$ são definidas pelas expressões:

$$\begin{aligned} s_i &= \underline{\Phi}_i^t\underline{\underline{C}}_{-i}^{-1}\underline{\Phi}_i \\ q_i &= \underline{\Phi}_i^t\underline{\underline{C}}_{-i}^{-1}\underline{Y} \end{aligned} \tag{5.89}$$

Os fatores s_i e q_i são quantidades relacionadas com a esparsidade da solução e com o ajuste do modelo aos dados. Enquanto s_i pode ser visto como uma medida do grau de sobreposição de $\underline{\Phi}_i$ em relação às funções de base correntemente utilizadas, q_i representa o nível de alinhamento entre $\underline{\Phi}_i$ e o erro cometido pelo modelo desconsiderando esta função de base, já que q_i também é dado por:

$$q_i=\sigma^{-2}\underline{\Phi}_i^t\left(\underline{Y}-\underline{\hat{Y}}_{-i}\right) \tag{5.90}$$

Em (5.90), $\underline{\hat{Y}}_{-i}$ é a saída gerada pelo modelo desconsiderando a função de base $\underline{\Phi}_i$.

Anulando a expressão dada em (5.88), pode ser verificada a ocorrência de dois pontos estacionários para $L\left(\underline{\alpha},\sigma^2\right)$. O primeiro dado em $\alpha_i^*=\infty$ e o segundo no ponto estimado pela relação:

$$\alpha_i^{**}=\frac{s_i^2}{q_i^2-s_i} \tag{5.91}$$

Como $\alpha_i\in\mathbb{R}^+$, na equação acima $q_i^2>s_i$. Para qualificação dos pontos estacionários, é necessário o estudo da segunda derivada de $L\left(\underline{\alpha},\sigma^2\right)$ em relação ao hiperparâmetro α_i, dada por:

$$\frac{\partial^2}{\partial \alpha_i^2} L\left(\underline{\alpha}, \sigma^2\right) = \frac{-\alpha_i^{-2} s_i^2 \left(\alpha_i + s_i\right)^2 - 2\left(\alpha_i + s_i\right)\left[\alpha_i^{-1} s_i^2 - q_i^2 + s_i\right]}{2\left(\alpha_i + s_i\right)^4} \tag{5.92}$$

Avaliada para α_i^{**} finito dado pela equação (5.91), a segunda derivada de $L\left(\underline{\alpha}, \sigma^2\right)$ é obtida pela seguinte expressão:

$$\left.\frac{\partial^2}{\partial \alpha_i^2} L\left(\underline{\alpha}, \sigma^2\right)\right|_{\alpha_i^{**}} = \frac{-s_i^2}{2\left(\alpha_i^{**}\right)^2 \left(\alpha_i^{**} + s_i\right)^2} \tag{5.93}$$

Como (5.93) é negativa para qualquer α_i^{**} e s_i, α_i^{**} é o único ponto de máximo de $L\left(\underline{\alpha}, \sigma^2\right)$ em relação ao hiperparâmetro α_i, desde que $q_i^2 > s_i$ [187].

Para o ponto estacionário $\alpha_i^* = \infty$, FAUL e TIPPING [187] mostram que a segunda derivada dada em (5.92) tende a zero à medida que α_i tende a infinito. Entretanto, o sinal do gradiente dado em (5.88) está relacionado com a diferença entre q_i^2 e s_i. Se $q_i^2 > s_i$, o gradiente para α_i^* é negativo, indicando a necessidade de decréscimo em α_i em direção ao único ponto de máximo α_i^{**} dado por (5.91). Portanto, $\alpha_i^* = \infty$ representa um ponto de mínimo. Por outro lado, se $q_i^2 < s_i$, $\alpha_i^* = \infty$ corresponde ao único ponto de máximo. Por último, se $q_i^2 = s_i$, os máximos $\alpha_i^* = \infty$ e α_i^{**} dado por (5.91) coincidem. FAUL e TIPPING [187] também mostram que a matriz *hessiana* de $L\left(\underline{\alpha}, \sigma^2\right)$ avaliada no ponto $\underline{\alpha}^{MP} \in \mathbb{R}^{N+1}$ formado pelos respectivos pontos de máximo α_i^* ou α_i^{**} é semi-definida negativa. Desta forma, o vetor $\underline{\alpha}^{MP}$ formado pelos respectivos α_i^{MP} iguais a α_i^* ou α_i^{**} corresponde a um ponto de máximo de $L\left(\underline{\alpha}, \sigma^2\right)$.

O critério baseado na diferença entre q_i^2 e s_i pode ser usado para definição dos vetores relevantes, obtendo assim as funções de base a serem utilizadas visto que $\alpha_i^{MP} = \infty$ corresponde a $W_i = 0$, ou seja, retirada da respectiva função de base do

modelo. Estas quantidades podem ser calculadas para todas as $N+1$ funções de base através das equações:

$$s_i = \frac{\alpha_i S_i}{\alpha_i - S_i} \qquad (5.94)$$
$$q_i = \frac{\alpha_i Q_i}{\alpha_i - S_i}$$

onde

$$S_i = \sigma^{-2} \underline{\Phi}_i^t \underline{\Phi}_i - \sigma^{-4} \underline{\Phi}_i^t \underline{\underline{\Phi}} \underline{\underline{\Sigma}} \underline{\underline{\Phi}}^t \underline{\Phi}_i \qquad (5.95)$$
$$Q_i = \sigma^{-2} \underline{\Phi}_i^t \underline{Y} - \sigma^{-4} \underline{\Phi}_i^t \underline{\underline{\Phi}} \underline{\underline{\Sigma}} \underline{\underline{\Phi}}^t \underline{Y}_i$$

Em (5.95), a matriz de modelagem $\underline{\underline{\Phi}}$ definida em (5.64) e a matriz de covariância $\underline{\underline{\Sigma}}$ estimada por (5.76) são calculadas considerando somente as funções de base correntemente utilizadas pelo modelo, ou seja, aquelas com $\alpha_i^{MP} \neq \infty$. Além disso, em (5.94), para funções de base não incluídas no modelo ($\alpha_i^{MP} = \infty$), $s_i = S_i$ e $q_i = Q_i$.

A diferença entre q_i^2 e s_i pode ser utilizada como critério de seleção de funções de base em um algoritmo construtivo. Partindo de um modelo simples, a cada iteração uma função de base pode ser escolhida, com a sua inclusão no modelo sendo definida através do cálculo de $\Omega_i = q_i^2 - s_i$. Se Ω_i for positivo, a função de base $\Phi_i(\underline{x})$ deve ser incluída, com α_i^{MP} sendo estimado por (5.91). Se $\Phi_i(\underline{x})$ já faz parte do modelo, o respectivo hiperparâmetro α_i^{MP} deve ser atualizado por (5.91). Do contrário, $\alpha_i^{MP} = \infty$ e $\Phi_i(\underline{x})$ deve ser excluída. Desta forma, para definição do algoritmo resta estabelecer um critério para definição da função de base a ser analisada a cada iteração.

Considerando σ^2 constante, o que é válido entre iterações, em [64] os autores desenvolvem expressões para avaliação do impacto da inclusão ou retirada de funções de base no logaritmo $L(\underline{\alpha}, \sigma^2)$ da verossimilhança, como também da re-estimação de

um dado hiperparâmetro α_i relacionado com uma função já incluída no modelo. Desta forma, pode ser selecionada a cada iteração a função que causar maior acréscimo em $L\left(\underline{\alpha},\sigma^2\right)$. Supondo que na *l*-ésima iteração o modelo apresente m funções de base e que a função $\Phi_i\left(\underline{x}\right)$ ainda não faça parte do modelo, ou seja, $\alpha_i^{MP}\left(l\right)=\infty$. Se $\Omega_i\left(l\right)$ for positivo, a variação $\Delta L\left[\underline{\alpha}\left(l\right),\sigma^2\left(l\right)\right]$ entre iterações, obtida em virtude da inserção de $\Phi_i\left(\underline{x}\right)$ a partir da próxima iteração, é dada por:

$$\Delta L\left[\underline{\alpha}\left(l\right),\sigma^2\left(l\right)\right]=\frac{1}{2}\left[\frac{Q_i^2\left(l\right)-S_i\left(l\right)}{S_i\left(l\right)}+\log\frac{S_i\left(l\right)}{Q_i^2\left(l\right)}\right] \tag{5.96}$$

Se por outro lado a função $\Phi_i\left(\underline{x}\right)$ pertence ao modelo e deve ser retirada, isto é, $\Omega_i\left(l\right)\le 0$ com $\alpha_i^{MP}\left(l+1\right)=\infty$, o impacto no logaritmo da verossimilhança é estimado por:

$$\Delta L\left[\underline{\alpha}\left(l\right),\sigma^2\left(l\right)\right]=\frac{1}{2}\left\{\frac{Q_i^2\left(l\right)}{S_i\left(l\right)-\alpha_i\left(l\right)}-\log\left[1-\frac{S_i\left(l\right)}{\alpha_i\left(l\right)}\right]\right\} \tag{5.97}$$

Por último, se $\Phi_i\left(\underline{x}\right)$ já pertence ao modelo e $\alpha_i^{MP}\left(l\right)$ deve ser re-estimado ($\Omega_i\left(l\right)>0$) segundo (5.91) obtendo o novo hiperparâmetro $\alpha_i^{MP}\left(l+1\right)$, $\Delta L\left[\underline{\alpha}\left(l\right),\sigma^2\left(l\right)\right]$ é dada pela expressão:

$$\Delta L\left[\underline{\alpha}\left(l\right),\sigma^2\left(l\right)\right]=\frac{1}{2}\frac{Q_i^2\left(l\right)}{S_i\left(l\right)+\left[\frac{1}{\alpha_i^{MP}\left(l+1\right)}-\frac{1}{\alpha_i^{MP}\left(l\right)}\right]^{-1}}-\log\left\{1+S_i\left[\frac{1}{\alpha_i^{MP}\left(l+1\right)}-\frac{1}{\alpha_i^{MP}\left(l\right)}\right]\right\} \tag{5.98}$$

Os respectivos impactos $\Delta L\left[\underline{\alpha}\left(l\right),\sigma^2\left(l\right)\right]$ dados nas expressões (5.96), (5.97) e (5.98) podem ser utilizados para definição da atitude a ser tomada na iteração corrente.

Calculado $\underline{\Omega}(l)=\left[\Omega_1(l) \quad \Omega_2(l) \quad \ldots \quad \Omega_{N+1}(l)\right]^t$, $\Omega_i(l)=q_i^2(l)-s_i(l)$, deve ser selecionada a função de base que produzir o maior impacto na verossimilhança, ou seja, maior $\Delta L\left[\underline{\alpha}(l),\sigma^2(l)\right]$. Definida a função, a atualização do respectivo hiperparâmetro $\alpha_i^{MP}(l+1)$ está relacionada com o respectivo $\Omega_i(l)$. Assim, o algoritmo construtivo para estimação de uma máquina de vetores relevantes pode ser resumido da seguinte forma:

1. Faça $l=0$ e $\underline{\alpha}(l)=\left[\alpha_1(l) \quad \ldots \quad \alpha_{N+1}(l)\right]^t=\left[\infty \quad \ldots \quad \infty\right]^t$ e $\underline{\mu}(l)=\left[0 \quad \ldots \quad 0\right]^t$.
2. Inicialize o hiperparâmetro $\sigma(l)$.
3. Selecione a primeira função de base a integrar o modelo.
4. Atualize o hiperparâmetro $\alpha_i^{MP}(l+1)$ da função de base selecionada utilizando a equação (5.91).
5. Considerando somente as funções de base integrantes do modelo na *l*-ésima iteração, atualize os parâmetros $\underline{\mu}(l+1)$ e $\underline{\underline{\Sigma}}(l+1)$ segundo (5.76).
6. Atualize o hiperparâmetro $\sigma(l+1)$ utilizando as expressões (5.79) e (5.81).
7. Se o critério de parada for atendido, encerre o algoritmo. Do contrário, siga para o próximo passo.
8. Faça $l=l+1$ e calcule os respectivos incrementos $\Delta L\left[\underline{\alpha}(l),\sigma^2(l)\right]$, conforme as expressões (5.96) a (5.98) em conjunto com as equações (5.94) e (5.95).
9. Escolha a função de base que produzir o maior incremento $\Delta L\left[\underline{\alpha}(l),\sigma^2(l)\right]$ e retorne ao passo 4.

O algoritmo acima requer a escolha da primeira função de base a integrar o modelo. TIPPING e FAUL [64] sugerem a escolha do bias $\underline{\Phi}_0$ ou da função de base

mais alinhada com as saídas desejadas $\underline{Y}$, ou seja, aquela que apresentar a maior projeção normalizada p_i dada por:

$$p_i = \frac{\left\| \underline{\underline{\Phi}}_i^t \underline{Y} \right\|^2}{\left\| \underline{\underline{\Phi}}_i \right\|^2} \tag{5.99}$$

O algoritmo construtivo para desenvolvimento de máquinas de vetores relevantes, além de calcular analiticamente os parâmetros e hiperparâmetros envolvidos, fornece mecanismos para seleção das funções de base a serem utilizadas. A maximização de $L\left(\underline{\alpha}, \sigma^2\right)$ - logaritmo da verossimilhança $p\left(\underline{Y} \middle| \underline{\alpha}, \sigma^2\right)$ - possui preocupação com a complexidade do modelo, visto que $L\left(\underline{\alpha}, \sigma^2\right)$ dado em (5.78) pode ser escrito da forma que segue [63]:

$$L\left(\underline{\alpha}, \sigma^2\right) = -\frac{1}{2}\left[N \log 2\pi - \log\left|\underline{\underline{\Sigma}}\right| + N \log \sigma^2 - \log\left|\underline{\underline{A}}\right| + \underline{\mu}^t \underline{\underline{A}}\, \underline{\mu} + \frac{1}{\sigma^2}\left\| \underline{Y} - \underline{\underline{\Phi}}\, \underline{\mu} \right\|^2 \right] \tag{5.100}$$

Desconsiderando a constante $N \log 2\pi$ em (5.100), $L\left(\underline{\alpha}, \sigma^2\right)$ apresenta parcelas relacionadas tanto com o controle de complexidade da estrutura estimada ($-\log\left|\underline{\underline{\Sigma}}\right| + N \log \sigma^2 - \log\left|\underline{\underline{A}}\right| + \underline{\mu}^t \underline{\underline{A}}\, \underline{\mu}$) quanto com o ajuste do modelo aos dados, representado pelo erro de modelagem ($\sigma^{-2}\left\| \underline{Y} - \underline{\underline{\Phi}}\, \underline{\mu} \right\|^2$), evidenciando a preocupação com a regularização do modelo estimado.

Ao abordar a questão da regularização da estrutura diretamente na função objetivo, a estimação de RVMs abdica do uso de métodos de validação cruzada para estimação dos hiperparâmetros. Entretanto, a definição da função de base $\Phi\left(\underline{x}, \underline{z}\right)$, incluindo seus hiperparâmetros, constitui uma questão tão importante quanto o problema de estimação dos parâmetros e hiperparâmetros das RVMs. Assim como para as SVMs, esta tarefa comumente é realizada através do uso de um conjunto específico

de dados para seleção do conjunto de hiperparâmetros da função de base escolhida pelo usuário. Diante da explosão combinatorial oriunda do aumento do espaço de busca, esta abordagem compromete a utilização de funções mais flexíveis que incluam múltiplos hiperparâmetros. Visando permitir o uso de tais funções, TIPPING [63] sugere o uso de um método iterativo baseado em gradiente para estimação dos diversos hiperparâmetros da função de base escolhida. Considerando uma função de base *gaussiana* similar à utilizada para SVMs, esta técnica permite o desenvolvimento de um método de seleção de entradas similar ao apresentado na seção 5.1.2, conforme será apresentado a seguir.

5.2.1 Determinação automática de relevância para RVMs

As metodologias apresentadas na seção anterior consideram a relação entre o logaritmo da verossimilhança $L\left(\underline{\alpha},\sigma^2\right)$ e os diversos hiperparâmetros, especificado o tipo de função de base $\Phi\left(\underline{x},\underline{z}\right)$ utilizada, para estimação de RVMs. Entretanto, os hiperparâmetros de $\Phi\left(\underline{x},\underline{z}\right)$ também influenciam o comportamento de $L\left(\underline{\alpha},\sigma^2\right)$, sendo necessários métodos para estimação dos mesmos.

Considere a função de base $\Phi\left(\underline{x},\underline{z}\right):\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}$ dada por:

$$\Phi\left(\underline{x},\underline{z}\right)=\Phi\left(\sqrt{\eta_1}x_1,\sqrt{\eta_2}x_2,\ldots,\sqrt{\eta_n}x_n,\sqrt{\eta_1}z_1,\sqrt{\eta_2}z_2,\ldots,\sqrt{\eta_n}z_n\right) \tag{5.101}$$

Em (5.101), $\underline{\eta}=\begin{bmatrix}\eta_1 & \ldots & \eta_n\end{bmatrix}^t$ representa o conjunto de hiperparâmetros de $\Phi\left(\underline{x},\underline{z}\right)$, que neste caso corresponde aos ponderadores de cada entrada. O gradiente de $L\left(\underline{\alpha},\sigma^2,\underline{\eta}\right)$ em relação ao *k*-ésimo ponderador é dado por:

$$\frac{\partial}{\partial\eta_k}L\left(\underline{\alpha},\sigma^2,\underline{\eta}\right)=\sum_{n=1}^{N}\sum_{m=2}^{N+1}\frac{\partial}{\partial\Phi_{nm}}L\left(\underline{\alpha},\sigma^2,\underline{\eta}\right)\frac{\partial\Phi_{nm}}{\partial\eta_k} \tag{5.102}$$

Na expressão acima, Φ_{nm} representa o elemento da *n*-ésima linha da *m*-ésima coluna da matriz de modelagem $\underline{\underline{\Phi}}\in\mathbb{R}^N\times\mathbb{R}^{N+1}$ dada pela equação (5.64). Como a primeira coluna

desta matriz responde pelo bias, os elementos desta coluna não são incluídos em (5.102). O gradiente de $L\left(\underline{\alpha},\sigma^2,\underline{\eta}\right)$ em relação a Φ_{nm} pode ser representado em uma matriz $\underline{\underline{D}} \in \mathbb{R}^N \times \mathbb{R}^N$, $D_{nm} = \partial L\left(\underline{\alpha},\sigma^2,\underline{\eta}\right) \big/ \partial \Phi_{nm+1}$, dada por [63]:

$$\underline{\underline{D}} = \left(\underline{\underline{C}}^{-1}\underline{Y}\underline{Y}^t\underline{\underline{C}}^{-1} - \underline{\underline{C}}^{-1}\right)\underline{\underline{\Phi}}\,\underline{\underline{A}}^{-1} \tag{5.103}$$

$$\underline{\underline{D}} = \frac{1}{\sigma^2}\left[\left(\underline{Y} - \underline{\underline{\Phi}}\,\underline{\mu}\right)\underline{\mu}^t - \underline{\underline{\Phi}}\,\underline{\underline{\Sigma}}\right]$$

Para o cálculo de (5.102) resta obter o gradiente de Φ_{nm} em relação ao ponderador η_k, sendo necessária a especificação da função de base utilizada. Seja uma função *gaussiana* $\Phi\left(\underline{x},\underline{z}\right): \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ similar à utilizada na seção 5.1.2 e neste contexto dada por:

$$\Phi\left(\underline{x},\underline{z}\right) = e^{-\sum_{k=1}^{n}\left(\sqrt{\eta_k}x_k - \sqrt{\eta_k}z_k\right)^2} = e^{-\sum_{k=1}^{n}\eta_k\left(x_k - z_k\right)^2} \tag{5.104}$$

Considerando esta função de base, o gradiente de $L\left(\underline{\alpha},\sigma^2,\underline{\eta}\right)$ em relação ao *k*-ésimo ponderador é dado por:

$$\frac{\partial}{\partial \eta_k} L\left(\underline{\alpha},\sigma^2,\underline{\eta}\right) = -\sum_{n=1}^{N}\sum_{m=2}^{N+1} D_{nm-1}\Phi_{nm}\left(x_{mk} - x_{nk}\right)^2 \tag{5.105}$$

A equação (5.105) pode ser utilizada em um algoritmo de subida em gradiente para maximização de $L\left(\underline{\alpha},\sigma^2,\underline{\eta}\right)$. Entretanto, a forma na qual esta expressão deve ser incluída no processo de estimação dos hiperparâmetros $\underline{\alpha}$ e σ^2, como também dos parâmetros $\underline{\mu}$ e $\underline{\underline{\Sigma}}$, ainda não está consolidada na literatura. TIPPING [63] sugere um algoritmo iterativo incluindo ciclos de atualização independentes para $\underline{\alpha}$ e σ^2, utilizando as equações (5.79) a (5.81) baseadas na maximização da evidência, e $\underline{\eta}$, através de subida em gradiente baseada na expressão (5.105). Neste trabalho, TIPPING enfatiza as dificuldades encontradas para combinar de maneira adequada e efetiva a

otimização dos hiperparâmetros $\underline{\alpha}$ e σ^2 com a otimização de η_k, destacando os empecilhos para definir o número de ciclos de atualização de cada conjunto de hiperparâmetros e indicando que tal escolha esteja diretamente ligada ao problema em estudo.

Visando automatizar o processo de estimação dos hiperparâmetros no que tange à combinação entre as distintas atualizações supracitadas, a informação sobre o gradiente de $L\left(\underline{\alpha},\sigma^2,\underline{\eta}\right)$ em relação a η_k será combinada com o algoritmo construtivo de estimação de RVMs apresentado na seção 5.2. Especificamente, após a estimação dos respectivos hiperparâmetros e parâmetros em virtude da função de base escolhida para análise, os hiperparâmetros η_k serão atualizados utilizando uma expressão simples de subida em gradiente. Além disso, visto que $\eta_k \geq 0$ para todo k, de forma análoga à desenvolvida para SVMs, a busca será realizada no espaço logarítmico, ou seja, $\nu_k = \log \eta_k$. O gradiente neste espaço passa a ser dado por:

$$\frac{\partial}{\partial \log\left(\eta_k\right)} L\left(\underline{\alpha},\sigma^2,\underline{\eta}\right) = \eta_k \frac{\partial}{\partial \eta_k} L\left(\underline{\alpha},\sigma^2,\underline{\eta}\right) \tag{5.106}$$

Mantendo a analogia com o método de seleção de entradas de SVMs apresentado na seção 5.1.2, a análise dos hiperparâmetros η_k obtidos ao final do algoritmo permite mensurar a relevância de cada entrada no cálculo da saída $\hat{d}$ dada em (5.84). Observando a equação (5.101), entradas com elevados valores de η_k apresentam maior contribuição para o cálculo da saída, contrastando com aquelas que possuem η_k de pequena magnitude. Diante da menor relevância destas últimas no cálculo da saída, algumas destas variáveis podem ser retiradas do modelo final, sendo necessária a definição de limiares de relevância. O método empírico de definição de limiares de relevância apresentado na seção 3.2 será utilizado para especificação de tais patamares,

a partir dos quais as entradas de RVMs podem ser descartadas. Desta forma, para sinais contínuos o hiperparâmetro η_C associado à variável de prova contínua representará o patamar de relevância para entradas desta natureza, com o mesmo valendo para o ponderador η_D relacionado com a variável de referência discreta. Desta forma, serão descartadas do modelo final entradas com hiperparâmetros η_k menores do que o respectivo patamar de referência.

5.2.2 Método automático de especificação de RVMs

O gradiente de $L\left(\underline{\alpha},\sigma^2,\underline{\eta}\right)$ em relação aos hiperparâmetros da função de base $\Phi\left(\underline{x},\underline{z}\right)$ dada em (5.104) pode ser incorporado ao método construtivo de estimação de RVMs, fornecendo uma metodologia para seleção de entradas deste tipo de modelo. Desta forma, o método automático de especificação de RVMs pode ser resumido da forma que segue:

1. Faça $l=0$.
2. Insira variáveis de prova ao conjunto de entradas original seguindo as distribuições de probabilidade apresentadas nas equações (3.36) e (3.37). Se as variáveis de entrada forem somente contínuas, insira somente o sinal de prova desta natureza. Do contrário, insira também a variável de prova discreta.
3. Inicialize os hiperparâmetros $\underline{\eta}(l)=\left[\eta_1(l) \quad \ldots \quad \eta_n(l)\right]^t$ de $\Phi\left(\underline{x},\underline{z}\right)$.
4. Faça $\underline{\alpha}(l)=\left[\alpha_1(l) \quad \ldots \quad \alpha_{N+1}(l)\right]^t=\left[\infty \quad \ldots \quad \infty\right]^t$ e $\underline{\mu}(l)=\left[0 \quad \ldots \quad 0\right]^t$.
5. Inicialize o hiperparâmetro $\sigma(l)$.
6. Selecione a primeira função de base a integrar o modelo.
7. Atualize o hiperparâmetro $\alpha_i^{MP}(l+1)$ da função de base selecionada utilizando a equação (5.91).

8. Considerando somente as funções de base integrantes do modelo na *l*-ésima iteração, atualize os parâmetros $\underline{\mu}(l+1)$ e $\underline{\underline{\Sigma}}(l+1)$ segundo (5.76).

9. Atualize o hiperparâmetro $\sigma(l+1)$ utilizando as expressões (5.79) e (5.81).

10. Atualize o conjunto de hiperparâmetros $\underline{\eta}(l+1)$ utilizando subida em gradiente através da seguinte expressão:

$$\log\left[\underline{\eta}(l+1)\right] = \log\left[\underline{\eta}(l)\right] + \rho \frac{\partial}{\partial \log(\eta_k)} L\left(\underline{\alpha}, \sigma^2, \underline{\eta}\right)\Big|_{\underline{\alpha}(l+1), \sigma^2(l+1), \underline{\eta}(l)} \tag{5.107}$$

11. Se o critério de parada for atendido, vá para o passo 14. Do contrário, siga para o próximo passo.

12. Faça $l = l+1$ e calcule os respectivos incrementos $\Delta L_i\left[\underline{\alpha}(l), \sigma^2(l)\right]$, conforme as expressões (5.96) a (5.98) em conjunto com as equações (5.94) e (5.95).

13. Selecione a *i*-ésima função de base que produzir o maior incremento $\Delta L_i\left[\underline{\alpha}(l), \sigma^2(l)\right]$. Se $\Delta L_i\left[\underline{\alpha}(l), \sigma^2(l)\right] < tol_L$ para todo $i = 0,1,\ldots,N$, vá para o passo 14. Do contrário retorne ao passo 7.

14. Agrupe os hiperparâmetros $\underline{\eta}$ associados a entradas contínuas e discretas em vetores distintos e ordene de forma crescente estes vetores.

15. Em cada vetor, selecione as variáveis de entrada situadas acima do respectivo limiar de relevância, ou seja, $\eta_k > \eta_C$ para variáveis contínuas e $\eta_k > \eta_D$ para as discretas.

16. Utilizando somente as entradas relevantes selecionadas no passo anterior, repita os passos 4 a 13, obtendo o modelo final e possibilitando a realização de previsões.

No algoritmo resumido acima, o passo ρ do estágio de subida em gradiente responsável pela atualização de $\underline{\eta}$ deve ser especificado pelo usuário, sendo feito

constante e igual a 10^{-2}. Fazendo menção ao sugerido em [63], será considerado um único ciclo de atualização de $\underline{\eta}$ a cada atualização dos demais parâmetros e hiperparâmetros. Outra questão que deve ser mencionada diz respeito ao critério de parada citado no passo 11 e à tolerância tol_L. A tolerância tol_L para a variação $\Delta L_i\left[\underline{\alpha}(l), \sigma^2(l)\right]$ máxima é considerada igual a 10^{-2}. Para o critério de parada, são consideradas duas possibilidades: número máximo de iterações, feito igual a $3N$ e raramente atingido; e máxima variação em $\underline{\mu}$ menor que a tolerância especificada, isto é, $\max\left|\underline{\mu}(l+1) - \underline{\mu}(l)\right| < tol_\mu$, com tol_μ igual a 10^{-3}.

Além dos parâmetros relacionados com o processo de otimização, o algoritmo acima requer a inicialização dos hiperparâmetros $\underline{\eta}(l)$ e $\sigma(l)$. Analogamente ao apresentado para SVMs, os hiperparâmetros $\underline{\eta}(l)$ inicialmente são feitos iguais a 10^{-1}, com os respectivos $\eta_i(l) = \eta_C(l)$ e $\eta_j(l) = \eta_D(l)$ iguais a 10^{-2}, visto que estão associados às variáveis sabidamente irrelevantes. Seguindo a recomendação de [64], o valor inicial do hiperparâmetro $\sigma(l)$ é feito igual a $0.1\delta_d^2$, onde δ_d representa o desvio padrão das saídas desejadas $\underline{Y}$, calculado através da equação (5.57).

O algoritmo listado acima apresenta um método automático para estimação de RVMs, incluindo métodos analíticos para seleção de funções de base e de entradas relevantes. Utilizando inferência *bayesiana* de forma análoga à desenvolvida por MACKAY [54] para MLPs, esta técnica produz modelos baseados em *kernel* com representação esparsa similar à obtida pelas SVMs, possuindo, porém, menos parâmetros a serem especificados pelo usuário. Além das tolerâncias requeridas pelo método de otimização e do conjunto inicial de entradas, o algoritmo acima requer exclusivamente a escolha da função de base $\Phi\left(\underline{x}, \underline{z}\right)$, o que corresponde à escolha da

função de ativação dos neurônios da camada oculta de MLPs e ao *kernel* utilizado pelas SVMs. Para o caso do algoritmo de especificação automática de MLPs apresentado na seção 4.4, por exemplo, é necessária ainda a especificação do intervalo de variação do número de neurônios na camada oculta. Em relação ao algoritmo para SVMs discutido na seção 5.1.3 e baseado em descida em gradiente, a escolha do ponto inicial do algoritmo é crucial para o desempenho do método em termos de erro de previsão. Combinando inferência *bayesiana* e representação esparsa baseada em *kernel*, as RVMs produzem modelos que requerem menor intervenção do usuário para sua especificação e estimação, motivando a sua utilização na busca por modelos autônomos de previsão.

5.3 Resumo e Discussão

Este capítulo apresentou as máquinas baseadas em *kernel* enfatisando os modelos *feedforward* utilizados neste trabalho, respectivamente as máquinas de vetor suporte (SVMs) e as máquinas de vetores relevantes (RVMs). De uma maneira geral, os populares perceptrons de múltiplas camadas (MLPs) estudados no capítulo 4 podem ser vistos como modelos baseados em *kernel*. Especificamente, já que as funções sigmoidais atendem às condições de *Mercer* para valores específicos dos parâmetros β_0 e β_1 na equação (5.21), os MLPs podem também ser entendidos como SVMs. Para o caso das RVMs, a analogia é direta visto que quaisquer funções de base $\Phi\left(\underline{x},\underline{z}\right)$ podem ser utilizadas. Desta forma, os neurônios da camada oculta de MLPs com uma única camada escondida desempenham o papel do *kernel* $K\left(\underline{x},\underline{x}_k\right)$ em (5.22) ou da função de base $\Phi\left(\underline{x},\underline{x}_i\right)$ em (5.61). Entretanto, esta é talvez a única semelhança entre o MLP e as máquinas baseadas em *kernel* estudadas neste trabalho.

A primeira diferença entre estes dois paradigmas de modelagem reside na definição da estrutura. Enquanto que para MLPs o número de camadas escondidas e de

neurônios por camada deve ser definido *a priori*, a estrutura das SVMs e RVMs é obtida como um subproduto do algoritmo de treinamento. Especificado o tipo de *kernel* $K\left(\underline{x},\underline{x}_k\right)$ e os parâmetros que o definem, assim como as constantes C e ε, os vetores suporte obtidos ao final da estimação de SVMs definem a estrutura. Analogamente, dada a função de base $\Phi\left(\underline{x},\underline{z}\right)$, o conjunto de vetores relevantes determinado pelo algoritmo de treinamento de RVMs produz automaticamente a estrutura a ser utilizada.

Em relação às SVMs, outra questão que merece destaque diz respeito à superfície a ser otimizada ao longo do treinamento. Enquanto que para MLPs esta superfície é extremamente não-convexa, repleta de mínimos locais, em SVMs tal superfície é quadrática, apresentando um único ponto de máximo. Assim, para um mesmo conjunto de dados, o treinamento de MLPs apresenta elevada dependência do ponto inicial do algoritmo, com o treinamento de SVMs resultando em uma única solução, mantidos constantes o tipo de *kernel* e seus parâmetros, e as constantes C e ε. Visto que a estimação destes parâmetros não é trivial, sendo necessária a busca em superfícies multimodais representando limites como $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$, o problema de mínimos locais também é um empecilho para SVMs, porém em um nível de inferência distinto.

Além da superfície a ser otimizada, outra diferença entre SVMs e MLPs reside no objetivo do treinamento de cada modelo. Se por um lado MLPs tem por base o princípio da otimização do risco empírico (minimização única e exclusiva do erro para o conjunto de dados disponível), o treinamento de SVMs encontra fundamento no princípio da otimização do risco estrutural, que busca a minimização do limite superior do erro de generalização. Este princípio parte do pressuposto que o erro para um conjunto independente de dados é limitado superiormente pela soma entre o erro para o conjunto de treinamento e uma parcela relacionada com a dimensão VC. A busca pelo

ajuste dos dados em conjunto com a maximização da margem de separação visa à minimização implícita de um limite superior para a dimensão VC, gerando um modelo intrinsecamente regularizado. Guardando analogia biológica, é esperado que a máquina de aprendizagem "aprenda" o mapeamento entrada-saída, e não "decore" tal função. Intuitivamente, a minimização do risco estrutural objetiva o "aprendizado", visto que minimiza o limite superior do erro para novos padrões. O mesmo não pode ser dito da minimização do risco empírico.

Apesar das desejáveis características teóricas das SVMs, estes modelos apresentam como principal deficiência a dificuldade no ajuste dos seus hiperparâmetros, comumente especificados por validação cruzada. Apesar de popular, esta abordagem compromete o uso de *kernels* com diversos hiperparâmetros como os utilizados neste trabalho, em virtude da explosão combinatorial resultante da busca em espaços de dimensionalidade elevada. Sobrepor esta deficiência é um dos fatores que motivam o desenvolvimento das RVMs, que através da aplicação de inferência *bayesiana* dão origem a modelos esparsos com poucos hiperparâmetros a serem especificados pelo usuário. A inserção de conhecimento prévio na forma de distribuições de probabilidade *a priori*, além de permitir o desenvolvimento de um funcional a ser otimizado que conjugue tanto o ajuste dos dados quanto o controle de complexidade, produz um método automático de seleção de estrutura e representação esparsa, conforme apresentado na seção 5.1.

A aplicabilidade dos métodos propostos é apresentada no próximo capítulo. Para comparação e verificação dos procedimentos, algumas técnicas comumente utilizadas na literatura também são testadas. Todos estes procedimentos são aplicados a três bases de dados de domínio público, visando à reprodutibilidade dos resultados. A descrição das demais técnicas testadas, como também das bases de dados, é feita a seguir.

6 Testes e resultados

Este capítulo apresentará uma descrição das bases de dados estudadas, sendo discutidas as séries temporais disponíveis, suas características e eventuais processamentos efetuados. Além destas questões, serão discutidos os métodos utilizados para criação dos conjuntos de treinamento, sendo definidas as entradas utilizadas, os horizontes de previsão desejados e os períodos nos quais os modelos serão testados. Visando a reprodutibilidade dos resultados em conjunto com comparações com metodologias previamente propostas, são analisadas três bases de dados de domínio público, utilizadas em competições entre modelos de previsão de carga e em outros trabalhos disponíveis na literatura.

Além da comparação com modelos propostos para abordagem específica de cada base de dados, as metodologias automáticas para desenvolvimento de modelos neurais apresentadas nos capítulos 4 e 5 são comparadas com técnicas comumente utilizadas em previsão de carga. A especificação destas técnicas, juntamente com alguns aspectos referentes a estas, são detalhados juntamente com os resultados obtidos. Antes, porém, é necessária a apresentação das bases de dados, motivando o início da próxima seção.

6.1 Bases de dados

Nesta seção serão apresentados os três conjuntos de dados de domínio público tratados neste trabalho. A primeira base de dados, utilizada em uma competição entre modelos de previsão de carga horária promovida em 1991, possui séries de carga e temperatura horária referentes à *Puget Sound Power and Light Company*, uma empresa norte-americana de energia. A segunda, estudada em uma competição promovida no ano de 2001, é constituída de séries de carga, verificada a cada meia-hora, e temperatura média diária, visando a modelagem do pico de carga diário da empresa eslovaca de energia *East-Slovakia Power Distribution Company*. O último conjunto de dados

apresenta informações de carga, temperatura e preço da energia, medidas a cada 30 minutos e disponibilizadas pelo *National Electricity Market Management Company* (NEMMCO), operador do mercado de energia australiano. Apesar de não ser utilizada em competições, esta base de dados é mais atual do que as anteriores, sendo também utilizada na literatura para avaliação de modelos de previsão de carga horária.

Além da metodologia automática para seleção de entradas apresentada no capítulo 3 e detalhada nas próximas seções, também foram realizados testes com conjuntos de entradas selecionados pelo usuário. Tais espaços de entrada, especificados para cada base de dados, são utilizados por todos os modelos estudados nesta tese, assim como eventuais partições do conjunto de treinamento. Vale destacar que a escolha da cardinalidade destes espaços é feita arbitrariamente grande visando verificar a eficiência das técnicas propostas de seleção de entradas.

Definido o conjunto de entradas e saídas, todas as variáveis contínuas são padronizadas, utilizando uma transformação linear que dá origem a sinais apresentando média nula e variância unitária. Considerando a série $S(k)$, o sinal padronizado $Z(k)$ associado a este histórico é obtido através da seguinte relação:

$$Z(k) = \frac{S(k) - \overline{S}}{\delta_S} \tag{6.1}$$

Na equação (6.1), $\overline{S}$ e δ_S representam a média amostral e o desvio padrão de $S(k)$, respectivamente dados por:

$$\overline{S} = \frac{1}{N}\sum_{k=1}^{N} S(k) \tag{6.2}$$

$$\delta_S = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}\left[S(k) - \overline{S}\right]}$$

6.1.1 *Puget Sound Power and Light Company*

A primeira base de dados utilizada neste trabalho apresenta dados horários de carga e temperatura disponibilizados pela *Puget Sound Power and Light Company*, empresa norte-americana de energia. Utilizada em uma competição entre modelos de previsão de carga realizada em 1991, este conjunto de dados pode ser encontrado em *www.ee.washington.edu/class/555/el-sharkawi/index_files/Page3404.html*. Esta base de dados apresenta informações horárias de carga, em [MWh/h], e temperatura, em [°F], para o período de 1º. de janeiro de 1985 a 12 de outubro de 1992, totalizando 68208 dados de carga e temperatura.

A competição realizada em 1997 visou o desenvolvimento de modelos de previsão da curva de carga diária seguindo os padrões especificados pela empresa geradora dos dados. Seguindo este padrão, para dias úteis, a previsão da curva de carga do próximo dia, em base horária, deve ser entregue às 9 horas da manhã do dia atual. Para fins de semana, às 9 horas da manhã de sexta-feira devem ser fornecidas as previsões das curvas de carga para sábado, domingo e segunda-feira. Desta forma, para estimação da curva de carga referente as terças, quartas, quintas e sextas-feiras, devem ser realizadas previsões de carga horária de 16 a 40 passos à frente. Para os demais dias da semana, ou seja, para previsão simultânea da curva de carga para sábado, domingo e segunda-feira, devem ser realizadas estimativas de 16 a 88 horas à frente. Utilizando dados referentes ao período de 1º. de janeiro de 1985 a 31 de outubro de 1990 (51120 dados horários de carga e temperatura) para especificação e estimação dos modelos iniciais, os sistemas desenvolvidos devem realizar previsões na forma apresentada acima para o período de 1º. de novembro de 1990 a 31 de março de 1991. Vale destacar que, a medida em que novos dados são aquisitados, ou seja, o período de teste vai sendo

efetivamente verificado, o estágio de estimação dos modelos de previsão pode ser repetido, incorporando estes novos dados ao conjunto original de treinamento.

Vários métodos foram utilizados ao longo desta competição, incluindo regressão múltipla, redes neurais recorrentes, MLP tradicional, *splines* variantes no tempo e as previsões realizadas por especialistas da própria *Puget Sound Power and Light Company*. O melhor modelo em termos de erro percentual absoluto médio para o período de teste foi proposto por RAMANATHAN *et. al.* [8]. Este método, vencedor da competição, divide a base de dados em 168 agrupamentos, visando o desenvolvimento de um modelo específico para cada hora da semana. Esta segmentação das séries encontra explicação na sazonalidade diária e semanal presente nas curvas de carga horária, ilustradas na Figura 6.1, onde são mostradas as curvas para as duas últimas semanas da base de dados disponível no início da competição. Estas curvas evidenciam a influência da hora do dia e do dia da semana na dinâmica da carga horária, visto que o comportamento da carga para duas semanas consecutivas é bastante similar.

Cada um dos 168 modelos desenvolvidos pela proposta vencedora apresenta estrutura simples baseada em regressão múltipla, incluindo uma parcela dinâmica de correção das previsões utilizando os erros cometidos para as últimas horas. Toda a base de dados é utilizada para estimação dos modelos, através de um algoritmo baseado em mínimos quadrados. Para estimação das parcelas de correção relacionadas aos erros de previsão, um processo iterativo também baseado na minimização do erro quadrático é aplicado. Maiores detalhes podem ser encontrados em [8].

Além da metodologia baseada na teoria do caos para seleção do espaço de entradas, visando verificar a eficácia das técnicas de seleção de entradas desenvolvidas, um conjunto extenso de variáveis inicialmente é utilizado para alimentar os modelos. Este conjunto inicial é escolhido tomando por base as entradas selecionadas por

modelos encontrados na literatura que tratam esta base de dados específica, como [8], [14], [30], [107]. Para facilitar a exposição, seja $\underline{I}_{a-b}(k) \in \mathbb{R}^{b-a+1}$ o vetor contendo os $(b-a+1)$ atrasos consecutivos da série $I(k)$, definido por:

$$\underline{I}_{a-b}(k) = \left[I(k-a) \quad I(k-a-1) \quad \ldots \quad I(k-b) \right]^t \tag{6.3}$$

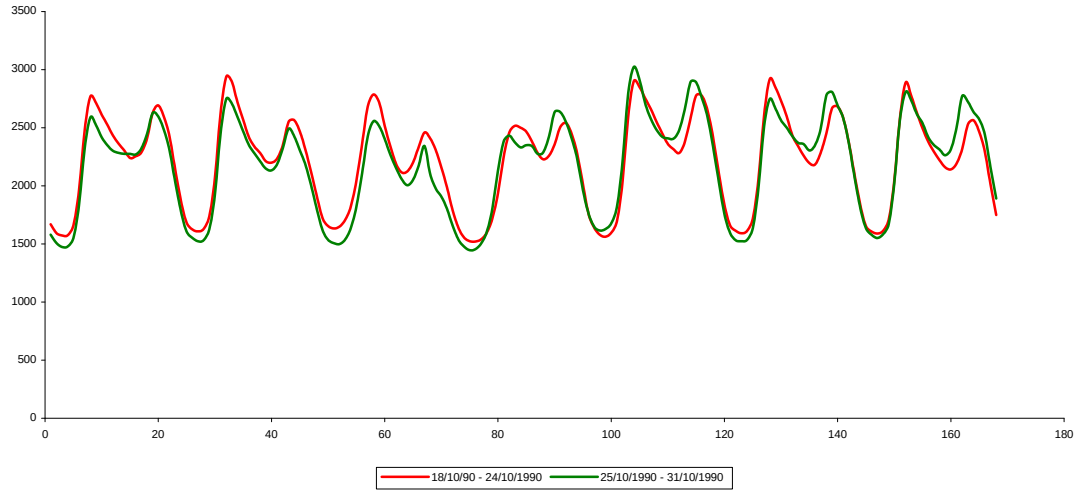

Figura 6.1 – Ilustração da sazonalidade diária e semanal da série de carga discutida na seção 6.1.1

Desta forma, considerando $L(k)$ como o valor da série de carga verificado no instante k, o vetor $\underline{IL}(k)$ representando os valores atrasados desta série utilizados como entradas é dado por:

$$\underline{IL}(k) = \left[\underline{IL}^t_{1-6}(k) \quad \underline{IL}^t_{24-29}(k) \quad \underline{IL}^t_{168-173}(k) \right]^t \tag{6.4}$$

Em (6.4), $\underline{IL}^t_{a-b}(k)$ representa o transposto do vetor definido em (6.3) contendo valores atrasados consecutivos de $L(k)$. Além das referências citadas, a escolha deste conjunto específico de atrasos encontra justificativa nas sazonalidades diária e semanal existentes nas séries de carga horária.

Para a série de temperatura, são utilizados os valores medidos nos mesmos instantes selecionados para a série de carga, além da previsão de temperatura para a hora a ser prevista. Matematicamente, o vetor $\underline{IT}(k)$ representando os valores atrasados da série de temperatura utilizados como entradas é dado por:

$$\underline{IT}(k) = \left[T(k) \quad \underline{IT}^t_{1-6}(k) \quad \underline{IT}^t_{24-29}(k) \quad \underline{IT}^t_{168-173}(k) \right]^t \tag{6.5}$$

Analogamente à série de carga, $\underline{IT}^t_{a-b}(k)$ é um vetor constituído de valores atrasos consecutivamente da série de temperatura, definido pela equação (6.3).

Transformações da série de temperatura também são utilizadas. Seja $TS(k)$ a série dada pelo quadrado $T^2(k)$ do valor da temperatura medida no instante k. Assim, o vetor $\underline{ITS}(k)$ contendo os valores atrasados de $TS(k)$ utilizados como entradas é definido da forma que segue:

$$\underline{ITS}(k) = \left[T^2(k) \quad \underline{ITS}^t_{1-6}(k) \quad \underline{ITS}^t_{24-29}(k) \quad \underline{ITS}^t_{168-173}(k) \right]^t \tag{6.6}$$

Mantendo a analogia, $\underline{ITS}^t_{1-6}(k)$ é dado pela expressão (6.3). Além da transformação quadrática, a temperatura máxima diária também é utilizada como entrada dos modelos. Representando esta nova série por $T_{\max}(i)$, com i indexando o dia da medição, os valores desta série utilizados como entradas podem ser agrupados no vetor $\underline{IT}_{\max}(k)$:

$$\underline{IT}_{\max}(k) = \left[T_{\max}(d) \quad T_{\max}(d-1) \right]^t \tag{6.7}$$

Na definição de $\underline{IT}_{\max}(k)$ dado pela equação (6.7), d representa o dia cuja hora k deve ser prevista. Valores atrasados da série obtida através do quadrado da temperatura máxima diária, $T^2_{\max}(i)$, também são utilizados como entradas dos modelos, sendo representados no vetor $\underline{ITS}_{\max}(k)$ dado por:

$$\underline{ITS}_{\max}(k) = \left[T^2_{\max}(d) \quad T^2_{\max}(d-1) \right]^t \tag{6.8}$$

As entradas de temperatura listadas acima utilizam informações sobre o instante k e o dia d a ser previsto, a saber, $T(k)$ e $T^2(k)$, $T_{\max}(d)$ e $T^2_{\max}(d)$, respectivamente. Considerando que serviços de meteorologia possam fornecer previsões precisas para estas variáveis, os modelos de previsão utilizam os próprios valores medidos destas grandezas. Desta forma, as entradas contínuas utilizadas para previsão da carga $L(k)$ no instante k podem ser agrupadas no vetor $\underline{IC}(k)$ dado por:

$$\underline{IC}(k) = \left[\underline{IL}^t(k) \quad \underline{IT}^t(k) \quad \underline{ITS}^t(k) \quad \underline{IT}^t_{\max}(k) \quad \underline{ITS}^t_{\max}(k) \right]^t \tag{6.9}$$

Das definições apresentadas nas equações (6.4) a (6.8), é visto que $\underline{IC}(k)$ apresenta um total de 60 componentes, significando que os modelos utilizados possuem 60 entradas contínuas.

Além de variáveis de natureza contínua, sinais discretos também são empregados. Conforme mostra a Figura 6.1, a dinâmica horária da carga está diretamente relacionada com a hora do dia. Como esta interdependência não apresenta relação de ordem, a representação 1 de *n* é a forma mais adequada de codificar esta informação. Seja $\underline{D}(k) \in \{0,1\}^{24}$ um vetor com todas as suas componentes nulas, com exceção da *j*-ésima coordenada, feita igual a 1. Supondo que o valor inicial $L(0)$ da série de carga tenha sido verificado na primeira hora do dia, o índice j associado à componente unitária é determinado pela seguinte relação:

$$j = \Gamma\left(\frac{k}{24}\right) \tag{6.10}$$

Na equação (6.10) $\Gamma(a/b) : \mathbb{N}^2 \rightarrow \mathbb{N}$ representa a função que retorna o resto da divisão entre dois números naturais a e b. Assim, seguindo a notação introduzida no capítulo

2, para previsão da saída desejada $d_k = L(k)$, o conjunto inicial de entradas $\underline{x}_k$ apresentado aos modelos de previsão utilizados neste trabalho é dado por:

$$\underline{x}_k = \left[\underline{IC}^t(k) \quad \underline{D}^t(k) \right]^t \tag{6.11}$$

O conjunto de entradas $\underline{x}_k$, possuindo um total de 84 variáveis, não apresenta sinais responsáveis pela representação direta da sazonalidade semanal evidenciada na Figura 6.1. De maneira análoga à utilizada em [8], esta característica será tratada através da segmentação da base de dados, sendo desenvolvidos sete modelos, um para cada dia da semana. Especificamente, os pares $(\underline{x}_k, d_k)$ associados a cada dia da semana são agrupados em sete subconjuntos distintos, sendo treinado um modelo específico para cada conjunto de dados.

A utilização de toda a base de dados disponível, cobrindo o período de 1º. de janeiro de 1985 a 31 de outubro de 1990, permite a utilização de cerca de 7300 padrões para treinamento de cada um dos sete modelos. Apesar de desejável, esta abundância de dados eleva decisivamente os custos computacionais dos algoritmos de treinamento. Além disso, a utilização indevida de dados históricos muito antigos pode comprometer a capacidade de generalização do modelo, visto que tais dados podem representar dinâmicas distintas da atual. Desta forma, para previsão da curva de carga em um dado mês, são utilizados para treinamento os últimos padrões verificados nesse mesmo mês juntamente com os pares $(\underline{x}_k, d_k)$ relacionados aos últimos dois meses. Os padrões verificados neste mesmo período do ano anterior são também incorporados ao conjunto de treinamento, perfazendo cerca de 650 pares $(\underline{x}_k, d_k)$ disponíveis para estimação do modelo. Esta segmentação da base de dados também pode ser explicada pela sazonalidade mensal presente nas série de carga horária, conforme ilustrado na Figura 6.2. Esta Figura mostra a curva de carga em base horária para o período de 25, segunda-

feira, a 31 de outubro de 1989, terça-feira. Para verificação da presença da sazonalidade, nesta Figura também é apresentado o último intervalo de sete dias começando numa segunda-feira para outubro de 1990, ou seja, o período de 24 a 30 de outubro deste ano. O padrão cíclico ilustrado pela similaridade entre as duas curvas está relacionado com as estações do ano, justificando a escolha segmentada do conjunto de treinamento utilizada neste trabalho.

Utilizando as entradas listadas acima juntamente com as respectivas partições do conjunto de dados, as previsões requeridas pela competição são realizadas de forma recursiva. Exemplificando, para previsão da curva de carga para terça-feira, o modelo de previsão estimado com dados referentes à segunda-feira efetua previsões de 1 a 16 passos à frente, começando pela nona hora da segunda-feira e terminando à meia-noite deste mesmo dia. Alimentado por estes resultados, o modelo obtido utilizando dados associados às terças-feiras estima a curva de carga para este dia, realizando previsões horárias de 1 a 24 horas à frente. Assim, partindo da nona hora do dia anterior, o sistema proposto realiza previsões horárias de 1 a 40 passos à frente para terças, quartas, quintas e sextas-feiras. De maneira análoga, para previsão conjunta das curvas de carga para sábado, domingo e segunda-feira, a metodologia desenvolvida estima cargas horárias de 1 a 88 passos à frente. Visando adaptar os modelos a medida em que novos dados são adquiridos, cada modelo é treinado uma vez por semana.

6.1.2 *East-Slovakia Power Distribution Company*

A base de dados relacionada a este empresa européia de energia apresenta dados de carga, em [MWh/h], verificados a cada meia-hora, e de temperatura média diária, em [°C], abrangendo o período de 1º. de janeiro de 1997 a 31 de dezembro de 1998. Este conjunto de dados, encontrado em *http://neuron.tuke.sk/competition*, foi utilizado na competição promovida em 2001 pelo *European Network on Intelligent Technologies for*

Smart Adaptive Systems, popularmente conhecido pela sigla EUNITE. Nesta competição, a tarefa dos modelos residiu na previsão do pico de carga diário para todo o mês de janeiro de 1999, sendo eleito o melhor aquele que apresentar menor erro absoluto percentual médio em conjunto com reduzido erro absoluto máximo.

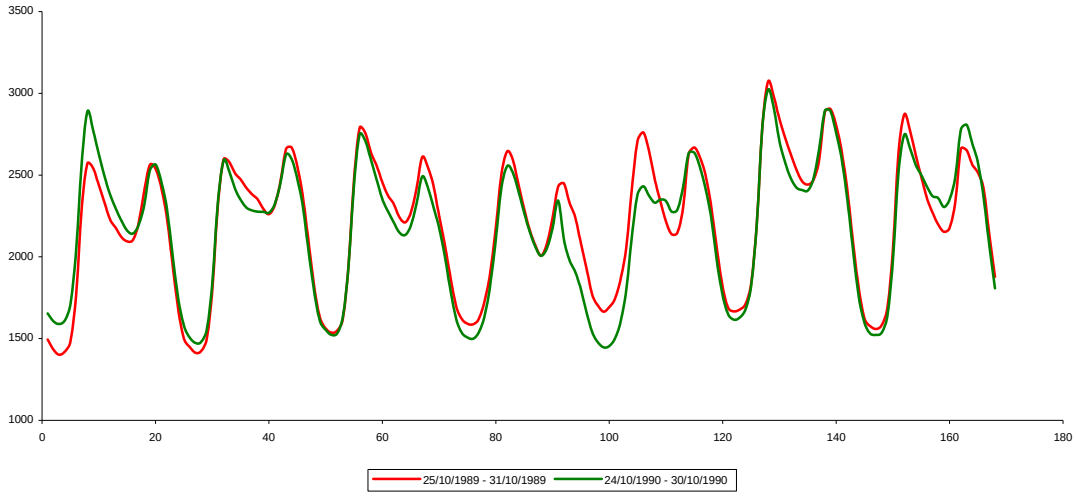

Figura 6.2 – Ilustração da sazonalidade mensal da série de carga discutida na seção 6.1.1

Dentre as diversas metodologias concorrentes, o modelo proposto por [29], baseado em SVMs com função linear de perda com tolerância ε dada pela equação (5.4) e com parâmetros estimados via validação cruzada, foi aclamado vencedor da competição. Visto que o conjunto de dados disponibilizado pela competição não contemplava previsões de temperatura para o período desejado, a utilização desta informação ficou comprometida, diante da necessidade de desenvolvimento de um modelo de previsão para esta série. Assim, a proposta vencedora não utilizou dados de temperatura no seu conjunto de entradas. Por outro lado, esta informação foi implicitamente valiosa, visto que o sistema proposto é treinado utilizando somente padrões associados à estação do ano em que são realizadas as previsões, no caso o inverno europeu.

Além de discussões referentes à competição, em [29] também são mostrados resultados obtidos por modelos apresentando em suas entradas variáveis de temperatura. Surpreendentemente, o modelo utilizando os valores verificados como previsões de temperatura para janeiro de 1999 apresentou desempenho de previsão inferior em relação ao modelo vencedor, que prescindiu de tal informação. Este resultado sinaliza a ausência de relação entre carga e temperatura para o período de janeiro de 1999, justificando a ausência desta grandeza nas entradas do modelo vencedor. Vale destacar que esta conclusão, apresentada em [29], não é esperada diante do forte relacionamento sazonal, relacionado com as estações do ano, existente entre carga e temperatura.

Diante dos resultados apresentados em [29], a inserção de entradas diretamente relacionadas com a série de temperatura pode contribuir para a avaliação das técnicas de seleção de entrada. De outra forma, visto que aparentemente estas variáveis contribuem de forma negativa para o desempenho de previsão, é esperado que os métodos propostos efetivamente retirem estas variáveis do modelo final. Assim, além das entradas utilizadas pelo modelo vencedor [29] serão adicionadas variáveis relacionadas com a série de temperatura.

Seguindo a abordagem utilizada em [29], a série com o pico de carga diário $L(k)$ será obtida a partir dos dados de carga $C(i)$ verificados a cada 30 minutos. Este novo histórico $L(k)$ será gerado através da obtenção do valor máximo medido entre os 48 $C(i)$s armazenados por dia e que podem ser encontrados na série de carga originalmente disponibilizada. Portanto, no processo de modelagem será utilizada a série de carga máxima diária $L(k)$ em conjunto com os dados de temperatura média diária $T(k)$ para o período de 1º. de janeiro de 1997 a 31 de dezembro de 1998, totalizando 730 medições para cada histórico.

Visando evitar os elevados níveis de recursão utilizados para abordagem da base de dados apresentada na seção 6.1.1, para este caso será desenvolvido um modelo para cada passo à frente. Para a base de dados anterior, esta abordagem necessitaria do desenvolvimento de até 80 modelos, elevando os requisitos computacionais para especificação e estimação do sistema de previsão como um todo. Além disso, enquanto o primeiro caso trata da estimação da curva de carga diária, este aborda a previsão da curva de carga mensal. Na prática, o tempo computacional disponível para geração das previsões para este caso é maior do que para o primeiro, possibilitando assim o desenvolvimento de um número maior de modelos. Como regra geral, o uso de recursão deve ser evitado, principalmente para elevados horizontes de previsão, visto que a incerteza inerente às previsões é realimentada no próprio modelo.

Assim, seguindo a notação utilizada na seção 6.1.1, para previsão da carga s passos à frente, ou seja, $L(k+s)$, o vetor $\underline{IL}(k+s)$ contendo os valores atrasados da série de carga utilizados como entradas é dado por:

$$\underline{IL}(k+s) = \underline{IL}_{0-6}(k) \tag{6.12}$$

Na equação (6.12), $\underline{IL}_{0-6}(k)$ é definido segundo a equação (6.3). Considerando os dados de temperatura referentes a janeiro de 1999 como previsões fornecidas por algum instituto de meteorologia, os valores atrasados da série de temperatura utilizados como entradas podem ser encapsulados em $\underline{IT}(k+s)$ da forma que segue:

$$\underline{IT}(k+s) = \underline{IT}_{(-s)-6}(k) \tag{6.13}$$

Em (6.13) $\underline{IT}_{(-s)-6}(k)$ é obtido pela equação (6.3). Assim, o conjunto de variáveis contínuas $\underline{IC}(k+s)$ utilizadas como entradas do modelo de previsão de carga s passos a frente pode ser definido:

$$\underline{IC}(k+s)=\left[\underline{IL}^{t}(k+s)\quad \underline{IT}^{t}(k+s)\right]^{t} \tag{6.14}$$

Portanto, para previsão de $L(k+s)$ são utilizadas $(s+14)$ variáveis contínuas.

De maneira semelhante às curvas de carga horária, a dinâmica do pico de carga diário também apresenta padrões sazonais. De maneira menos eloqüente do que para a curva de carga diária, estas questões podem ser identificadas na Figura 6.3 e na Figura 6.4. Na primeira, a relação existente entre o pico de carga diário e o dia da semana é exemplificada pelas curvas semanais apresentadas para duas semanas consecutivas de janeiro de 1998. Este padrão explica o uso do conjunto de entradas contínuas representado por $\underline{IC}(k+s)$ e dado por (6.14). A interdependência entre a carga máxima diária e o mês é ilustrada na Figura 6.4, que mostra a evolução desta grandeza ao longo dos anos de 1997 e 1998. Na realidade, esta componente sazonal está relacionada com as estações do ano, conforme realçado nesta Figura. Entretanto, a transição entre as estações é de difícil modelagem, dificultando esta forma de representação. Além disso, devido à escassez de dados disponibilizados, a segmentação do conjunto de treinamento de maneira análoga à utilizada para a base de dados descrita na seção 6.1.1 não é recomendável. Desta forma, as duas parcelas sazonais identificadas serão codificadas nas entradas dos modelos, através de variáveis binárias seguindo a representação 1 de *n* utilizada anteriormente.

Para codificação da sazonalidade semanal, seja $\underline{S}(k)\in\{0,1\}^{7}$ um vetor nulo a menos da sua *j*-ésima componente, a qual é feita igual a 1. O índice j relacionado à componente unitária é dado por:

$$j=\Gamma\left(\frac{k}{7}\right) \tag{6.15}$$

Em (6.15) $\Gamma(a/b):\mathbb{N}^2 \to \mathbb{N}$ representa a função que retorna o resto da divisão entre dois números naturais a e b. Analogamente, o vetor nulo $\underline{M}(k) \in \{0,1\}^{12}$ apresentando a *j*-ésima coordenada unitária pode ser utilizado para codificação da parcela sazonal relacionada ao mês. Neste caso, *j* é dado pelo mês associado ao instante k no qual o respectivo pico de carga $L(k)$ é verificado. Assim, o conjunto de variáveis discretas utilizado para previsão do valor máximo de carga *s* passos à frente, $L(k+s)$, pode ser aglutinado no vetor $\underline{D}(k+s)$ representado por:

$$\underline{D}(k+s) = \left[\underline{S}^t(k+s) \quad \underline{M}^t(k+s) \right]^t \tag{6.16}$$

Seguindo a notação utilizada, o vetor $\underline{x}_{k+s}$ representando o conjunto de entradas utilizadas para modelagem da saída desejada $d_{k+s} = L(k+s)$ é dado por:

$$\underline{x}_{k+s} = \left[\underline{IC}^t(k+s) \quad \underline{D}^t(k+s) \right]^t \tag{6.17}$$

Para o modelo de previsão de carga s passos à frente, o espaço de entrada apresenta cardinalidade igual a $(33+s)$.

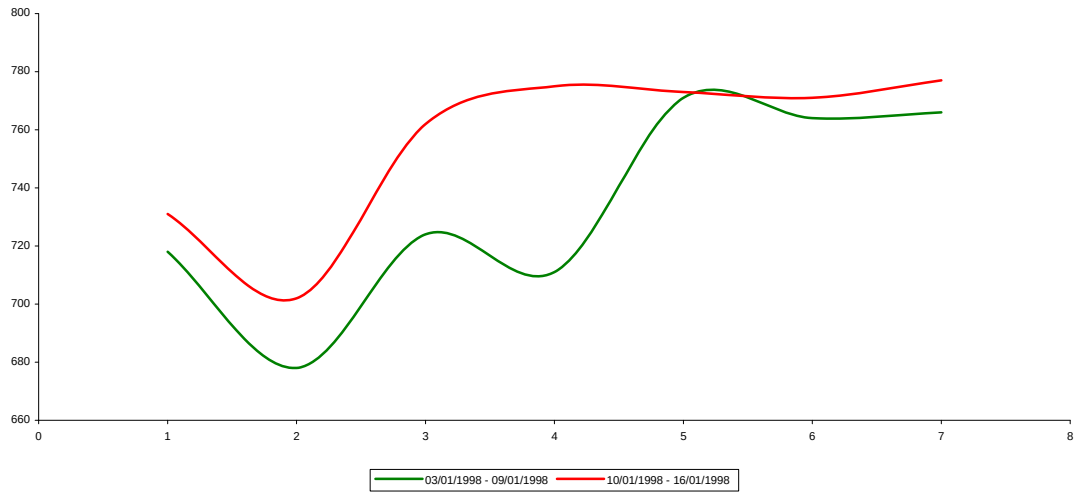

Figura 6.3 – Ilustração da sazonalidade diária presente na série de carga descrita na seção 6.1.2

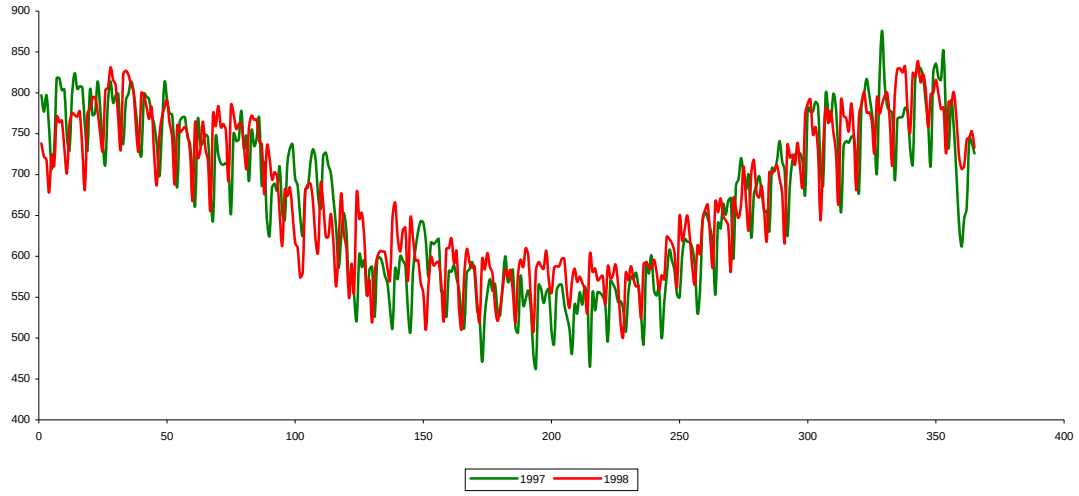

Figura 6.4 – Ilustração da sazonalidade mensal presente na série de carga descrita na seção 6.1.2

Utilizando todos os dados disponíveis para treinamento e o respectivo conjunto de entradas $\underline{x}_{k+s}$ dado por (6.4), são treinados 31 modelos, ou seja, s variando de 1 a 31, sendo então estimada a curva de carga máxima para o mês de janeiro de 1999.

6.1.3 *National Electricity Market Management Company*

O operador do mercado de energia australiano, conhecido pela sigla NEMMCO, disponibiliza em seu site (www.nemmco.com.au) bases de dados com informações referentes a diversas regiões da Austrália. Especificamente, podem ser encontrados históricos de carga e preço da energia, verificados a cada trinta minutos e relacionados a seis subsistemas, a saber: *Queensland, New South Wales, the Australian Capital Territory, Victoria, South Austrália* e *Tasmânia.* Neste trabalho, são utilizados dados de carga e preço da região de *Victoria*, abrangendo o período de 4 de dezembro de 2001 a 31 de dezembro de 2003, totalizando 36384 valores para cada série. Diante da característica influência das condições climáticas na dinâmica de curto prazo da carga, dados de temperatura, verificados também a cada 30 minutos, para esta mesma região da Austrália e cobrindo o mesmo período também são utilizados. Estas informações

podem ser encontradas no site do Departamento Australiano de Meteorologia (*Australian Bureau of Meteorology*), localizado em *www.bom.gov.au*.

O conjunto de dados sucintamente descrito acima foi utilizado em [31], [69] e [70]. Estas três referências utilizam o mesmo método de previsão, distinguindo entre si basicamente em relação às semanas utilizadas para avaliação dos modelos. Visando comparar as técnicas propostas com as melhores práticas encontradas na literatura, os resultados obtidos neste trabalho serão comparados com os apresentados na referência mais recente.

Utilizando uma técnica não supervisionada para segmentação da base de dados baseada em informações de temperatura, em [70] são desenvolvidos MLPs para previsão de carga horária de 1 a 6 passos à frente. O desempenho destes modelos neurais é avaliado para o período de 1 a 6 de setembro de 2003. O processo de modelagem utiliza somente séries em base horária, que são obtidas dos históricos medidos a cada trinta minutos através da média aritmética entre pares consecutivos. Exemplificando, a carga relacionada à segunda hora (2:00) é considerada como a média entre as cargas verificadas à 1:30 e às 2:00, respectivamente. Esta transformação é aplicada a todas as séries, com os modelos sendo alimentados somente com dados horários de carga, temperatura e preço.

A segmentação da base de dados utilizada em [70] visa tratar a variação da carga em virtude de mudanças climáticas através do agrupamento de padrões similares. Com base em informações de temperatura, são selecionados para treinamento somente os padrões semelhantes ao dia a ser previsto. O nível de similaridade é medido pela distância euclidiana entre os vetores com informações meteorológicas associados a cada padrão, sendo escolhidos aqueles que apresentarem menor distância em relação ao vetor representante do dia a ser previsto. Portanto, para cada passo à frente, um MLP é

treinado por dia utilizando somente padrões similares em termos de condições climáticas.

Tendo em mente a reprodutibilidade e a comparação fidedigna entre os modelos, a transformação efetuada em [70] para obtenção dos históricos horários de carga, temperatura e preço é também aplicada neste trabalho. Assim, com dados horários para o período de 4 de dezembro de 2001 a 31 de agosto de 2003, devem ser realizadas previsões de uma a seis horas à frente para toda a primeira semana de setembro de 2003.

Novamente, visando verificar a eficácia das técnicas de seleção de entradas, o conjunto de variáveis que inicialmente irão alimentar os modelos de previsão será definido seguindo as recomendações de [31], [69], [70]. Além disso, visando evitar o uso de recursão, analogamente a [31], [69], [70], serão desenvolvidos diferentes modelos para cada passo à frente. Seguindo a notação utilizada na seção 6.1.1, para previsão da carga s horas à frente, ou seja, $L(k+s)$, o vetor $\underline{IL}(k+s)$ contendo os valores atrasados da série de carga utilizados como entradas é dado por:

$$\underline{IL}(k+s)=\left[\underline{IL}^{t}_{0-s}(k)\quad \underline{IL}^{t}_{24-29}(k)\quad \underline{IL}^{t}_{168-173}(k)\right]^{t} \tag{6.18}$$

Na equação (6.18), $\underline{IL}_{0-s}(k)$, $\underline{IL}^{t}_{24-29}(k)$ e $\underline{IL}^{t}_{168-173}(k)$ são definidos segundo a equação (6.3). Supondo que os dados de temperatura referentes à primeira semana de setembro de 2003 possam ser fornecidos por algum instituto de previsão climática, os valores atrasados da série de temperatura utilizados como entradas podem ser encapsulados em $\underline{IT}(k+s)$ da forma que segue:

$$\underline{IT}(k+s)=\left[\underline{IT}^{t}_{(-s)-(6-s)}(k)\quad \underline{IT}^{t}_{24-29}(k)\quad \underline{IT}^{t}_{168-173}(k)\right]^{t} \tag{6.19}$$

De forma análoga, $\underline{IT}^{t}_{(-s)-(6-s)}(k)$, $\underline{IT}^{t}_{24-29}(k)$ e $\underline{IT}^{t}_{168-173}(k)$ são obtidos através da equação (6.3). A utilização de entradas relacionadas com previsões de preço da energia

necessita do desenvolvimento de um modelo para esta variável, o que foge ao escopo deste trabalho inicial. Desta forma, os valores atrasados da série $P(k)$ representando o preço da energia em base horária a serem utilizados como entradas podem ser aglutinados em $\underline{IP}(k+s)$ dado por:

$$\underline{IP}(k+s) = \left[\underline{IP}^t_{0-s}(k) \quad \underline{IP}^t_{24-29}(k) \quad \underline{IP}^t_{168-173}(k) \right]^t \tag{6.20}$$

Na equação (6.20), $\underline{IP}^t_{0-s}(k)$, $\underline{IP}^t_{24-29}(k)$ e $\underline{IP}^t_{168-173}(k)$ seguem a equação (6.3). Portanto, o conjunto de variáveis contínuas $\underline{IC}(k+s)$ utilizadas como entradas do modelo de previsão s horas à frente é dado por:

$$\underline{IC}(k+s) = \left[\underline{IL}^t(k+s) \quad \underline{IT}^t(k+s) \quad \underline{IP}(k+s) \right]^t \tag{6.21}$$

Desta maneira, para modelagem de $L(k+s)$ são utilizadas $(57-2s)$ variáveis contínuas.

Os padrões sazonais verificados nas curvas de carga horária discutidas na seção 6.1.1 podem também ser identificados na base australiana de dados. A Figura 6.5 ilustra a sazonalidade semanal, com a Figura 6.6 exemplificando a presença da componente relacionada com as estações do ano. Nesta última Figura, de maneira análoga às curvas apresentadas na Figura 6.2, a última semana de agosto de 2002, começando no domingo dia 25 e terminando no dia 31, sábado, é apresentada juntamente com o último período de sete dias começando em domingo para agosto de 2003, ou seja, o intervalo do dia 24 ao dia 30.

A componente sazonal ilustrada na Figura 6.5, associada com a relação entre dinâmica da carga horária e fatores de calendário como hora do dia e dia da semana, fornece mais uma justificativa para a escolha das variáveis contínuas representadas na equação (6.21). Além destas variáveis, analogamente à abordagem da série de carga

horária discutida na seção 6.1.1, a influência da hora do dia no comportamento da carga a curto prazo será incluída na entrada dos modelos, utilizando representação 1 de *n*. Esta codificação é realizada através do vetor $\underline{D}(k+s) \in \{0,1\}^{24}$ apresentando todas as suas componentes nulas, com exceção da *j*-ésima coordenada, a qual é feita igual a 1. Supondo que o valor inicial $L(0)$ da série de carga tenha sido verificado na primeira hora do dia, o índice j associado à componente unitária é determinado pela relação dada na equação (6.10).

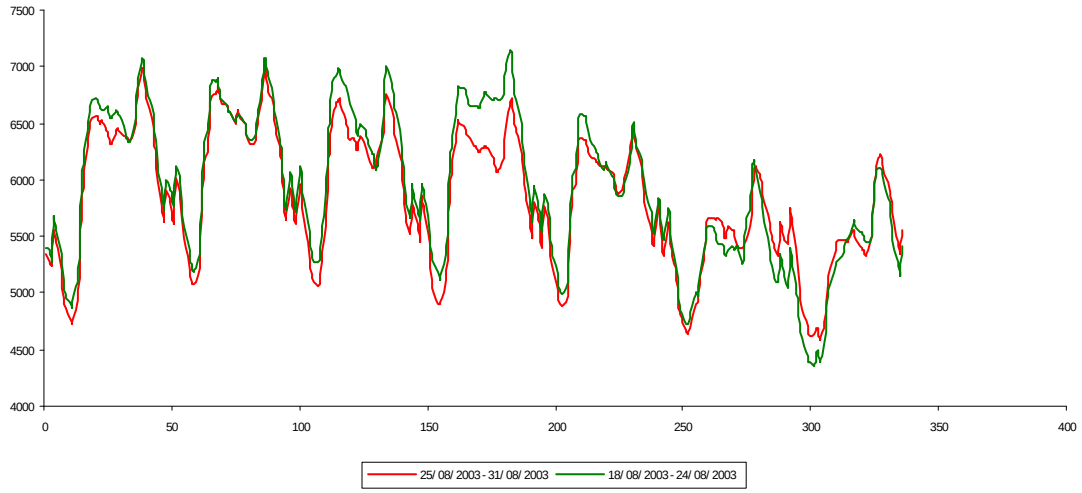

Figura 6.5 – Ilustração da sazonalidade semanal presente na base de dados australiana

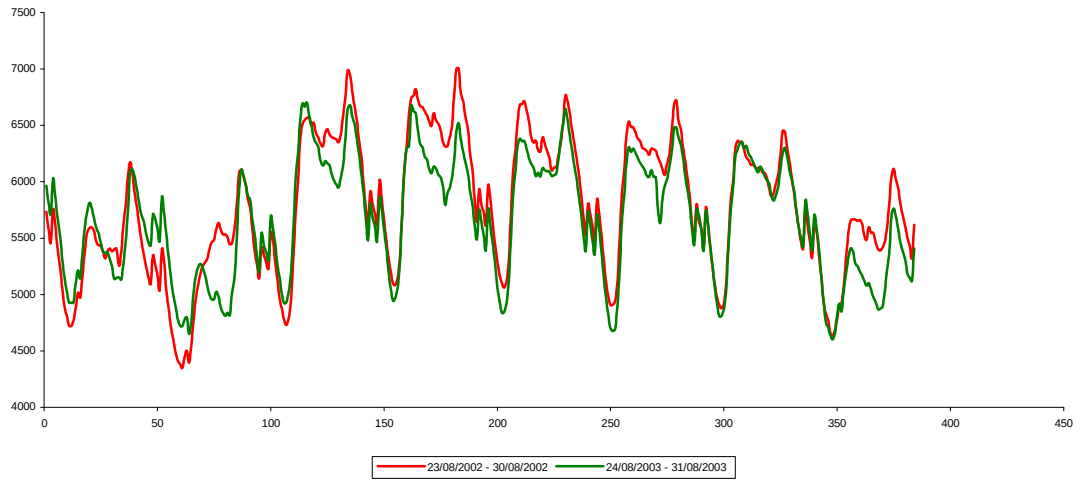

Figura 6.6 – Ilustração da sazonalidade mensal presente na base de dados australiana

Desta forma, o vetor $\underline{x}_{k+s}$ contendo as variáveis de entrada utilizadas para modelagem da saída desejada $d_{k+s} = L(k+s)$ é dado por:

$$\underline{x}_{k+s} = \left[\underline{IC}^t(k+s) \quad \underline{D}^t(k+s) \right]^t \tag{6.22}$$

Para esta base de dados, o modelo de previsão de carga horária s passos à frente possui espaço de entrada contendo $(81-2s)$ sinais contínuos e discretos.

Mantendo a analogia com o procedimento discutido na seção 6.1.1, os padrões sazonais relacionados com o dia da semana e com as estações do ano serão tratados através da segmentação da base de dados. Para o *k*-ésimo passo à frente, são desenvolvidos sete modelos, um para cada dia da semana, com os respectivos conjuntos de treinamento sendo obtidos através do agrupamento dos pares $(\underline{x}_{k+s}, d_{k+s})$ associados a cada dia da semana específico. Dentro de cada subconjunto, são escolhidos para treinamento somente os padrões verificados no mês para o qual devem ser realizadas as previsões, juntamente com os dados referentes aos dois meses imediatamente anteriores e com os pares verificados neste mesmo período no ano anterior. Desta forma, o conjunto de treinamento de cada modelo apresenta cerca de 530 padrões.

6.2 Métodos Utilizados

As bases de dados descritas anteriormente foram utilizadas para teste das metodologias propostas neste trabalho. Fazendo uso dos conjuntos de entrada especificados na seção anterior, procedimentos populares para desenvolvimento de modelos neurais comumente encontrados na literatura também foram testados. Juntamente com o algoritmo *bayesiano* de desenvolvimento de MLPs (BMLP) descrito na seção 4.4 e com a técnica automática de especificação de L2-SVMs (AL2-SVM) apresentada na seção 5.1.3, algumas técnicas de seleção de entradas e controle de complexidade de modelos neurais também foram aplicadas. Especificamente, o método

de seleção de entradas utilizando índices de correlação linear (CL) apresentado na seção 2.1.1, baseado na identificação de níveis de dependência entre entrada e saída e de redundância entre sinais de entrada, é combinado com algumas das técnicas de controle de complexidade descritas na seção 2.2, a saber: estabilização de estrutura (ES), parada antecipada do treinamento (PAT) e escalonamento do ganho da função de ativação (EGFA). Assim, os procedimentos testados para modelos neurais foram os seguintes:

- Número de neurônios na camada oculta do MLP especificado pelo usuário, com o modelo sendo treinado via retropropagação do erro tradicional, ou seja, minimização do risco empírico dado pela equação (2.22). Este modelo será identificado pela sigla (RP).
- Escalonamento do ganho da função de ativação aplicado a MLPs com número de neurônios na camada oculta especificado pelo usuário. Método identificado pela sigla (EGFA).
- Estabilização de estrutura através da determinação do número de neurônios na camada oculta baseada no desempenho para um conjunto de validação, com os modelos treinados através do algoritmo de retropropagação do erro original seguido pela heurística de escalonamento do ganho da função de ativação. A sigla (ES-EGFA) será utilizada para identificação deste procedimento.
- Seleção de entradas baseada na análise de índices de correlação linear e estabilização de estrutura através da especificação do número de neurônios com base no desempenho para um conjunto de validação, como todos os modelos sendo treinados utilizando o algoritmo de retropropagação do erro seguido pelo escalonamento do ganho da função de ativação. A sigla (CL-ES-EGFA) identifica esta metodologia.

- Parada antecipada do treinamento aplicada a MLPs com número de neurônios na camada oculta especificado pelo usuário. Esta heurística será associada à sigla (PAT).
- Estabilização de estrutura através da determinação do número de neurônios na camada oculta tomando por base o desempenho para um conjunto de validação, com os modelos treinados através da parada antecipada do algoritmo de retropropagação do erro. Este método será denominado pela sigla (ES-PAT).
- Seleção de entradas baseada na análise de índices de correlação linear e estabilização de estrutura através da especificação do número de neurônios com base no desempenho para um conjunto de validação, como todos os modelos treinados utilizando parada antecipada do treinamento. A sigla (CL-ES-PAT) rotula este método.
- Inferência *bayesiana* aplicada ao desenvolvimento de MLPs, denotada pela sigla (BMLP).
- Parâmetros que definem a L2-SVM especificados pelo usuário, ou seja, as constantes C e ε, e o parâmetro do *kernel gaussiano* $K\left(\underline{x}_i, \underline{x}_j\right)$ dado pela equação (5.20). Este método será associado à sigla (L2-SVM).
- Parâmetros que definem a L2-SVM definidos através da análise do desempenho para um conjunto de validação. Este método será associado à sigla (VCL2-SVM).
- Seleção de entradas baseada no estudo de índices de correlação linear e parâmetros que definem a SVM especificados através da análise do desempenho para um conjunto de validação, procedimento identificado pela sigla (CL-VCL2-SVM).
- Método automático de especificação de L2-SVMs (AL2-SVM).

– Método automático de especificação de RVMs (A-RVM).

Além das estruturas neurais desenvolvidas também é utilizado um modelo linear simples. Estimado através do método de mínimos quadrados tradicional, este modelo utiliza inicialmente as mesmas entradas listadas na seção 6.1, mais um parâmetro de intercepto. Para seleção automática de entradas, o teste dos multiplicadores de *Lagrange* [161], [162], ao nível de significância de 99 %, é utilizado para avaliação da significância dos parâmetros e conseqüente eliminação de variáveis. Visto que a inclusão de variáveis binárias tem por objetivo a representação dos padrões sazonais verificados nas séries em estudo, este teste é aplicado somente às entradas de natureza contínua, com as discretas sendo mantidas no modelo final. Os resultados referentes a este método serão listados sob a sigla LINEAR.

Alguns métodos descritos acima necessitam da especificação de certos parâmetros por parte do usuário. Para minimização do risco empírico $E_S\left[f\left(\underline{x},\underline{w}\right)\right]$ dado pela equação (2.22), juntamente com a otimização de $S\left(\underline{w}\right)$ dado pela equação (4.23), é utilizado o algoritmo de retropropagação do erro baseado em gradiente conjugado. Resumidamente apresentado no APÊNDICE B, este algoritmo pode ser encontrado em [188]. Visto que são dois funcionais distintos, os critérios de parada utilizados também diferem. Para minimização de (2.22), o algoritmo baseado em gradiente conjugado evolui até atingir um valor mínimo para este funcional, especificado neste trabalho em 10^{-9}. Portanto, $E_S\left[f\left(\underline{x},\underline{w}\right)\right]$ é minimizado enquanto a condição $E_S\left[f\left(\underline{x},\underline{w}\right)\right]\Big|_{\underline{w}=\underline{w}(l)} \leq 10^{-9}$ não for satisfeita. No caso do funcional $S\left(\underline{w}\right)$ definido na equação (4.23), o processo iterativo é interrompido a partir da identificação da sua convergência, verificada pela variação máxima nas componentes de $\underline{w}$ entre duas iterações consecutivas. Esta tolerância é feita igual a 10^{-4}, ou seja, a otimização de

$S(\underline{w})$ continua enquanto $\max\left|\underline{w}(l)-\underline{w}(l+1)\right|>10^{-4}$. Para estimação dos multiplicadores de *Lagrange* associados com as SVMs através da minimização de (5.16), é utilizado um algoritmo baseado na otimização sequencial mínima, do inglês *sequential minimal optimization* (SMO), e disponibilizado em [189]. Maiores detalhes sobre este algoritmo, como também sobre SMO podem ser encontrados em [190].

Além dos algoritmos, outros parâmetros devem ser definidos pelo usuário. No caso da estabilização de estrutura de MLPs através da definição do número m de neurônios na camada oculta, valores limites $[m_{\min}, m_{\max}] \in \mathbb{N}$ para esta variável devem ser especificados. Analogamente aos limites definidos para a inferência *bayesiana* aplicada a MLPs, para estabilização de estrutura são testados modelos com número de neurônios na camada escondida variando entre $m_{\min}=1$ e $m_{\max}=10$. Apesar de escolhido de forma heurística, este número máximo foi definido tendo em mente o número de padrões disponíveis para treinamento e a quantidade de parâmetros a serem estimados. Em outras palavras, para os conjuntos de treinamento escolhidos dentre as bases de dados em estudo, seriam necessários mais padrões para estimação adequada de modelos mais parametrizados do que aquele gerado a partir do número máximo de neurônios especificado anteriormente, sendo esperado que um modelo adequado para as séries em estudo apresente número de neurônios na camada oculta dentro da faixa citada. Os resultados da Tabela 6.5 comprovam e ilustram esta questão, mostrando que na média os modelos selecionados apresentam estrutura dentro da faixa especificada e não no seu limite superior, indicando a necessidade de modelos mais parametrizados. Vale lembrar que, para os MLPs utilizandos neste trabalho, estes neurônios possuem função de ativação sigmoidal, dada pela equação (2.23), com o ganho a constante e igual a 1. Este parâmetro é modificado somente na técnica de escalonamento do ganho da função de ativação. Para este método, um intervalo de variação para o parâmetro

σ_{ruido} também deve ser definido. O conjunto de possíveis valores de σ_{ruido} utilizado neste trabalho está limitado no intervalo $[0,3\delta_d] \in \mathbb{R}$, com δ_d respondendo pelo desvio padrão amostral das saídas d_k. Discretizado em passos de $0.01\delta_d$, este intervalo dá origem desde modelos sem escalonamento dos ganhos ($\sigma_{ruido} = 0$), até estruturas treinadas puramente com ruído ($\sigma_{ruido} = 3\delta_d$). No caso da parada antecipada do treinamento, enquanto não é atingida a convergência do algoritmo de retropropagação com base no erro para o conjunto de treinamento, ao final de cada época é verificado o desempenho para o conjunto de validação, sendo armazenado ao final do treinamento o modelo que apresentar o menor erro para este subconjunto. Para as SVMs especificadas por validação cruzada, os intervalos de busca para as constantes que definem estes modelos também devem ser definidos. Desta forma, o parâmetro de regularização C varia no intervalo $[C_0, 1000C_0] \in \mathbb{R}$ e a tolerância ε na faixa $[0, 2\varepsilon_0] \in \mathbb{R}$, com C_0 e ε_0 dados pelas equações (5.56) e (5.58), respectivamente. Na ausência de valores recomendados para o parâmetro σ do *kernel* $K(\underline{x}_i, \underline{x}_j)$ dado pela equação (5.20), neste trabalho serão buscados valores para esta variável no intervalo $[0.001, 1] \in \mathbb{R}$. Para cada variável, serão testados 10 valores dentro de cada intervalo, sendo avaliado um total de 1000 modelos. Esta explosão combinatória explica a utilização do *kernel gaussiano* $K(\underline{x}_i, \underline{x}_j)$ dado pela equação (5.20) em lugar da sua versão modificada dada pela equação (5.46) para as SVMs especificadas por validação cruzada, visto que neste caso seria necessária a estimação de 10^{n+2} modelos, onde n representa o número de entradas.

Conforme mencionado intensivamente ao longo do texto, a definição do conjunto de validação é puramente heurística, sendo extremamente dependente do

problema em estudo. Neste trabalho, este conjunto será especificado de forma simples, partindo das bases de dados de treinamento descritas na seção 6.1, as quais são segmentadas em dois subconjuntos, um para treinamento e outro para validação. Sorteados de forma aleatória, 2/3 dos padrões disponíveis são separados para o primeiro grupo, com os dados restantes sendo dedicados para avaliação dos modelos.

As metodologias listadas acima foram testadas considerando o conjunto inicial de entradas definido pelo usuário. Tendo em mente o desenvolvimento de modelos neurais autônomos, a metodologia para seleção do conjunto inicial de entradas baseada na teoria do caos e resumida no algoritmo listado na seção 3.1.5 foi aplicada em dois modelos, a saber:

- Inferência *bayesiana* aplicada ao desenvolvimento de MLPs, denotada pela sigla (CHAOS-BMLP) neste caso;
- Método automático de especificação de máquinas de vetores relevantes, denotada pela sigla (CHAOS-RVM) neste caso.

Além dos BMLPs terem apresentado o melhor desempenho em termos de precisão das previsões para os testes iniciais considerando o conjunto de entradas definido pelo usuário, as RVMs fundem a representação esparsa dos modelos baseados em *kernel* com a inferência *bayesiana* para estimação dos parâmetros e principalmente dos hiperparâmetros. Diante dos empecilhos verificados na otimização dos hiperparâmetros das SVMs, estes modelos baseados em *kernel* foram escolhidos para teste da metodologia de seleção de entradas.

O método de seleção de entradas baseado na teoria do caos também necessita da definição de alguns parâmetros. Ao utilizar o primeiro mínimo da função de informação mútua $I_X(k)$ como estimativa para o atraso τ da imersão, é necessário definir um método para estimação das diversas probabilidades ou funções de densidade de

probabilidade envolvidas. O APÊNDICE A apresenta dois métodos automáticos, um baseado em histogramas para cálculo de probabilidades e outro utilizando estimadores de *Parzen* para estimação de densidades de probabilidade. Em que pese a suposição de distribuição *gaussiana* para obtenção de estimativas para o intervalo de discretização h_{hist} no caso de histogramas e do comprimento do *kernel h* nos estimadores de *Parzen*, o uso de expressões analíticas para estes parâmetros é de suma importância tendo em mente o desenvolvimento de modelos neurais autônomos. Existem na literatura metodologias sofisticadas para estimação destes parâmetros [191], [192], porém o elevado custo computacional inviabiliza o uso destas técnicas em modelos neurais para previsão de carga, cujo esforço computacional já é razoavelmente elevado conforme apresentado na Tabela 6.7. Esta restrição acerca do requisito computacional do modelo de previsão como um todo norteou a escolha do método baseado em histogramas para o cálculo de $I_X(k)$, após testes iniciais verificarem a similaridade entre os resultados obtidos pelas estimativas obtidas através de histogramas e de estimadores de *Parzen*. Definido o atraso da imersão, a dimensão d foi estimada utilizando o método baseado em falsos vizinhos mais próximos apresentado na seção 3.1.1. Para definição automática do ponto de saturação da estatística $\delta(d)$, foi considerado um nível de significância α de 99%, com a dimensão máxima $d_{\max}$ igual a 30.

6.3 Resultados

As metodologias citadas na seção 6.2 foram aplicadas aos conjuntos de dados descritos e definidos na seção 6.1. Para melhor compreensão e apresentação dos resultados, referências à base de dados da *Puget Sound Power and Light Company* apresentada na seção 6.1.1 serão denominadas como caso 1. Os resultados relacionados ao conjunto de dados da *East-Slovakia Power Distribution Company* listado na seção

6.1.2 serão rotulados como caso 2. Por último, as estatísticas para os modelos abordando os históricos da *National Electricity Market Management Company* (NEMMCO) discutidos na seção 6.1.3 serão identificadas como caso 3.

Considerando o conjunto de entradas inicial e a partição do conjunto de treinamento definido pelo usuário, é apresentado na Tabela 6.1 o erro absoluto percentual médio, conhecido pela sigla MAPE (*mean absolute percentage error*), gerado pelas metodologias utilizadas para as diversas bases de dados. Para o caso 3, esta estatística é mostrada para cada um dos seis passos à frente. As duas últimas linhas desta Tabela apresentam, respectivamente, esta medida de desempenho para os modelos encontrados na literatura desenvolvidos especificamente para cada base de dados (*benchmark*), e os eventuais ganhos promovidos pelas técnicas testadas. Vale lembrar que as referências contendo os melhores resultados para cada base de dados são [8], [29] e [70].

Os resultados apresentados na Tabela 6.1 mostram o desempenho de previsão superior obtido pela inferência *bayesiana* aplicada ao treinamento de MLPs (BMLP). Com exceção da base de dados norte-americana (caso 1), este método mostrou o menor MAPE para todos demais casos estudados. Mesmo para o caso 1, a diferença para os resultados obtidos em [8] é mínima, evidenciando a eficiência do método automático de previsão. Vale destacar que a metodologia proposta em [8], apesar de simples, requer intervenção dedicada de especialistas no processo de modelagem, no que tange tanto à seleção de entradas e aplicação de eventuais transformações a estas, incluindo interações entre variáveis, quanto à definição da própria estrutura do modelo. No caso da inferência *bayesiana* aplicada a MLPs, dado um conjunto de entradas, as mais relevantes em termos de capacidade de previsão são selecionadas automaticamente, com a intervenção de especialistas requisitada somente no estágio de seleção do conjunto

inicial. Para definição da estrutura do modelo, o usuário necessita definir somente o número mínimo e máximo de neurônios na camada oculta, com a estrutura mais adequada sendo escolhida também de forma automática.

Tabela 6.1 – Desempenho dos métodos para os diferentes casos (MAPE)

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
RP	10,43	5,05	0,97	1,33	1,49	1,57	1,80	1,72
EGFA	14,18	4,87	1,53	1,60	1,86	1,97	2,09	2,44
ES-EGFA	13,76	2,19	1,53	1,68	1,94	1,81	2,26	2,50
CL-ES-EGFA	17,80	2,77	2,41	3,58	3,51	3,03	3,24	3,38
PAT	8,07	1,95	2,04	1,93	2,09	2,60	2,00	2,35
ES-PAT	7,11	2,13	1,61	1,44	1,49	1,57	1,78	1,46
CL-ES-PAT	11,41	2,87	2,14	2,26	2,39	2,27	2,27	2,22
BMLP	4,89	1,75	0,49	0,72	0,82	0,94	0,99	1,07
L2-SVM	6,58	3,05	1,56	1,64	1,69	1,71	1,73	1,76
CVL2-SVM	4,88	3,52	0,81	0,93	1,06	1,15	1,20	1,36
CL-CVL2-SVM	10,54	2,87	1,57	2,15	2,15	2,15	2,24	2,24
AL2-SVM	8,72	2,07	0,88	0,84	1,01	1,20	1,56	1,20
A-RVM	8,46	2,76	0,60	1,23	0,99	1,40	1,11	1,18
LINEAR	4,97	2,23	0,56	0,84	1,09	1,23	1,35	1,77
Benchmark	4,73	1,98	0,56	0,83	1,00	1,15	1,20	1,30
Ganho (%)	-3,09	11,72	11,73	13,40	18,17	17,99	17,65	17,62

A técnica automática para especificação de L2-SVMs (AL2-SVMs) apresentou resultados razoáveis, próximos aos obtidos pelas referências a menos para o caso 1. Quando comparado com a escolha do modelo por validação cruzada (CVL2-SVM), este método mostrou melhor desempenho para metade dos casos, com o CVL2-SVM apresentando resultado realmente superior somente para o caso 1. Curiosamente, este caso foi o único para o qual o método automático apresentou pior desempenho do que o modelo especificado pelo usuário (L2-SVM). Os parâmetros que definem este modelo são o ponto inicial do algoritmo de descida em gradiente utilizado pelo AL2-SVM. Este resultado mostra que, ao contrário dos demais históricos, para o caso 1 a minimização de $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$ não produziu melhoria na capacidade de generalização. Este fato indica que a comparação entre modelos tomando por base os respectivos limites superiores do erro de generalização nem sempre conduzirá à escolha daquele com melhor capacidade de generalização. Intuitivamente, é esperado que modelos mais

complexos apresentem limite superior maior do que o estimado para modelos mais simples, em virtude do maior grau de flexibilidade disponibilizado. Porém, se a série em estudo apresentar elevada variabilidade, a capacidade de generalização efetivamente verificada para o modelo mais complexo pode ser eventualmente menor do que a obtida para o modelo mais simples. Diante do elevado número de recursões requerido pelo caso 1, esta questão pode explicar o resultado insatisfatório apresentado pela técnica automática de especificação de L2-SVMs para esta base de dados. As dificuldades encontradas na minimização do limite superior do erro de generalização visando estimar valores ótimos para os hiperparâmetros de L2-SVMs (multimodalidade, sensibilidade no ajuste do passo da descida em gradiente) também justificam o desempenho insatisfatório de previsão verificado por este modelo.

Os empecilhos identificados acima podem ser superados pelo uso de inferência bayesiana para estimação dos parâmetros e hiperparâmetros de modelos esparsos baseados em *kernel* como as máquinas de vetores relevantes (A-RVM), conforme mostram os resultados apresentados na Tabela 6.1. Quando comparado com o método automático para especificação de L2-SVMs, a A-RVM apresentou erros absolutos percentuais médios inferiores para cinco dos oito casos estudados (incluindo os seis passos à frente do caso 3). Apesar de apresentarem desempenho de previsão inferior em relação ao BMLP, o resultado superior da A-RVM quando comparadas às L2-SVMs motiva o aprofundamento do desenvolvimento desta metodologia, justificando a aplicação das técnicas baseadas em teoria do caos para seleção de entradas a estes modelos, cujos resultados serão apresentados ainda nesta seção.

Os demais resultados apresentados na Tabela 6.1 mostram a importância da seleção de entradas e do controle de complexidade de modelos neurais de forma sistemática e analítica. A menos do caso 1, as técnicas automáticas propostas neste

trabalho superaram todas as metodologias comumente encontradas na literatura para abordagem destes problemas. Em conjunto com os resultados superiores obtidos pelo modelo linear simples (LINEAR), esta questão evidencia a queda de desempenho de previsão dos modelos neurais quando a questão do controle de complexidade e da seleção de entradas não é tratada de forma adequada, com exceção para o caso 2 onde a parada antecipada do treinamento (PAT) apresentou resultado residualmente inferior quando comparado ao BMLP. Além disso, a técnica de seleção de entradas baseada em análise de índices de correlação linear (CL) deteriorou o desempenho de todos os modelos, com exceção da L2-SVM especificada por validação cruzada para o caso 2. Esta questão já era esperada, visto que esta técnica captura somente dependências lineares entre variáveis. Sinais representando grandezas como temperatura e preço estão sabidamente relacionados de forma não-linear com a carga, podendo, desta forma, ser descartados do modelo final segundo este método. Estas questões reafirmam a necessidade de utilização de técnicas analíticas adequadas para escolha do espaço de entrada e regularização de modelos neurais de previsão de carga, ao contrário da prática comum encontrada na literatura.

Na Tabela 6.2 são mostrados os erros absolutos máximos, índice conhecido pela sigla MAE (*maximum absolute error*), obtidos pelos diferentes métodos aplicados às bases de dados analisadas. Para comparação com as referências encontradas na literatura, para o caso 2 esta estatística está em [MW], representando realmente o erro absoluto máximo. No caso 3, a referência [70] apresenta este erro na forma percentual, sendo chamado neste trabalho de erro absoluto percentual máximo, denotado pelo símbolo MAE%. Como não são disponibilizadas informações sobre o erro máximo em [8], para o caso 1 os resultados são apresentados em termos do MAE%. De forma análoga à Tabela 6.1, as duas últimas linhas desta Tabela mostram os valores obtidos

pelas referências encontradas na literatura para cada base de dados, juntamente com os eventuais ganhos de desempenho.

Tabela 6.2 – Desempenho dos métodos para os diferentes casos (MAE e MAE%)

	Caso 1 (%)	Caso 2 (MW)	Caso 3 (%)					
			1 passos	2 passos	3 passos	4 passos	5 passos	6 passos
RP	93,12	118,89	4,99	5,96	4,50	5,85	6,73	8,02
EGFA	66,54	137,78	7,61	10,66	9,10	11,22	10,82	11,48
ES-EGFA	87,50	55,95	6,89	9,16	14,87	7,46	11,88	10,21
CL-ES-EGFA	112,89	70,99	11,61	20,48	23,85	11,36	12,77	11,22
PAT	43,98	40,28	7,84	10,38	7,91	15,13	7,02	12,95
ES-PAT	46,07	50,90	5,97	6,79	5,56	6,18	7,16	5,73
CL-ES-PAT	54,03	71,26	7,32	9,43	8,66	8,34	8,52	8,72
BMLP	41,57	55,64	1,97	2,65	3,89	4,62	4,86	5,46
L2-SVM	32,83	58,86	5,21	5,50	5,61	5,88	5,97	6,02
CVL2-SVM	38,06	60,39	4,00	3,51	4,53	4,62	5,45	5,95
CL-CVL2-SVM	60,06	67,17	5,90	6,19	6,18	6,17	6,48	6,48
AL2-SVM	46,70	59,78	3,48	4,05	5,12	5,87	6,14	5,59
A-RVM	55,42	47,21	2,77	8,32	4,37	7,81	5,61	5,99
LINEAR	39,86	65,17	2,10	3,13	4,70	6,20	7,00	6,39
Benchmark	-	51,42	3,24	3,43	4,11	3,87	5,57	5,20
Ganho (%)	-	21,66	39,09	22,64	5,44	-19,26	12,69	-4,93

A Tabela 6.2 confirma o desempenho satisfatório do desenvolvimento automático de MLPs com base em inferência *bayesiana* (BMLP). Em quatro dos oito casos testados, esta técnica apresentou menor erro absoluto máximo. Para o caso 2, a parada antecipada do treinamento apresentou um excelente resultado, superior que o obtido pelo modelo vencedor da competição promovida em 2001 [29]. Surpreendentemente, a L2-SVM definida pelo usuário e utilizada como ponto de partida para o método automático de especificação destes modelos (AL2-SVM) foi o método que mostrou melhor desempenho com base nesta estatística para o caso 1. Além de corroborar o resultado apresentado na Tabela 6.1, este resultado mostra que, juntamente com as questões previamente levantadas sobre os limites superiores, o ponto de partida do algoritmo de descida em gradiente já apresenta um potencial considerável em termos de magnitude de $\tilde{T}\left[f\left(\underline{x},\underline{W},b\right)\right]$. Em termos de erro absoluto máximo, a metodologia automática para especificação de RVMs (A-RVM) não apresentou nenhum resultado expressivo, superando as AL2-SVMs em quatro dos oito casos testados. Por fim, o

desempenho mostrado na Tabela 6.2 pelos modelos obtidos seguindo os procedimentos comumente encontrados na literatura confirma as conclusões tiradas a partir da Tabela 6.1. O caráter heurístico da parada antecipada do treinamento e do ganho da função de ativação, em conjunto com a inadequação a modelos não-lineares da técnica de seleção de entradas baseada em índices de correlação, prejudica a aplicação destes métodos visando o desenvolvimento de modelos com elevada capacidade de generalização.

Para ilustrar ainda mais os resultados obtidos, as curvas das Figura 6.7 à Figura 6.14 exemplificam as previsões realizadas pelos modelos testados. Para facilitar a visualização, são mostradas somente as estimativas geradas pelos métodos propostos neste trabalho e pelas metodologias autônomas encontradas na literatura. Assim, além dos métodos desenvolvidos nesta tese, nestas Figuras são mostradas as previsões realizadas por MLPs treinados através do algoritmo de retropropagação utilizando as entradas especificadas pelo usuário (RP); MLPs com estabilização de estrutura e treinados através do escalonamento do ganho da função de ativação, com as entradas originais sendo filtradas através da análise dos índices de correlação (CL-ES-EGFA); MLPs com entradas selecionadas desta mesma forma e estabilização de estrutura, porém estimados com parada antecipada do treinamento (CL-ES-PAT); L2-SVMs especificadas por validação cruzada, com as entradas filtradas utilizando análise dos índices de correlação linear (CL-CVL2- SVM); e o modelo linear (LINEAR). Estas Figuras confirmam o desempenho satisfatório do BMLP, contrastando com os resultados desanimadores obtidos pelo escalonamento do ganho da função de ativação. Esta técnica é atrativa em virtude da sua simplicidade e do requisito computacional mínimo, conforme mostra a equação (2.24). Apesar destas características desejáveis, os testes mostram a baixa efetividade deste método no desenvolvimento de modelos com considerável desempenho de previsão.

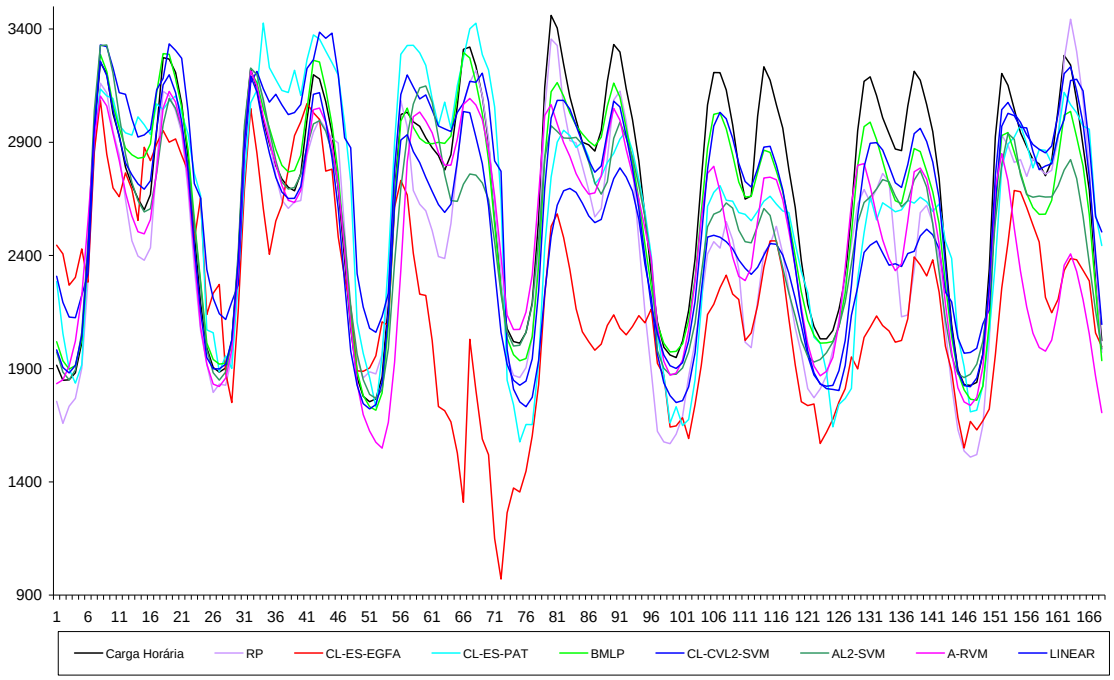

Figura 6.7 – Previsões de carga horária realizadas para o caso 1, cobrindo a semana de 27/11/1990 a 3/12/1990

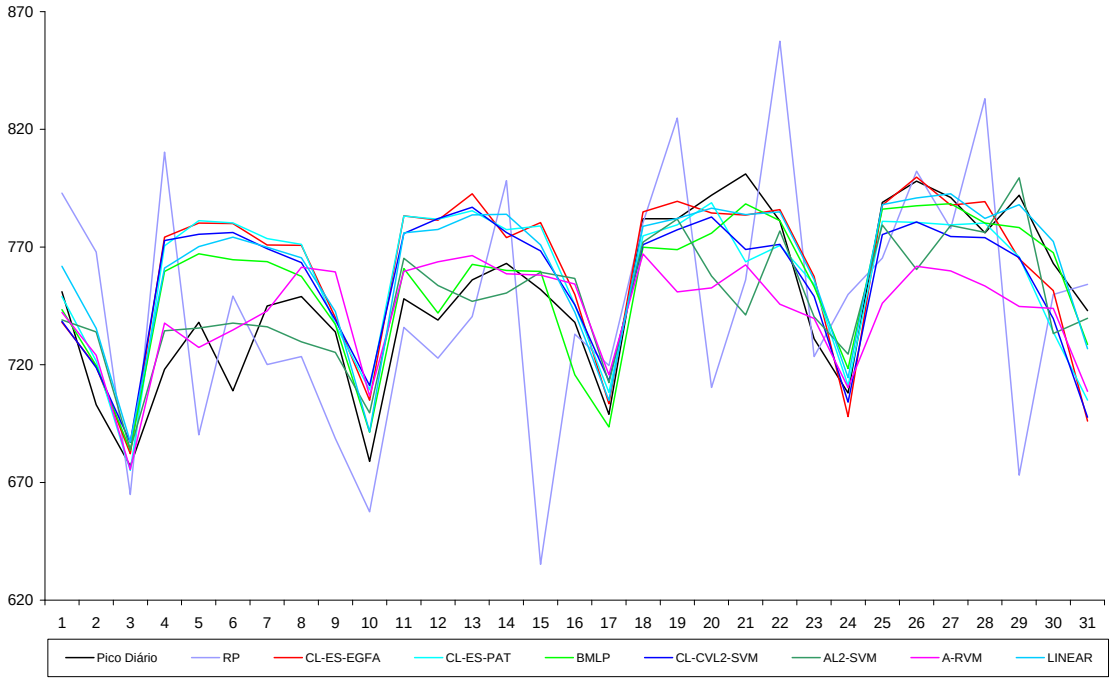

Figura 6.8 – Previsões de pico de carga diário realizadas para o caso 2, cobrindo o período de 1/1/1999 a 31/1/1999

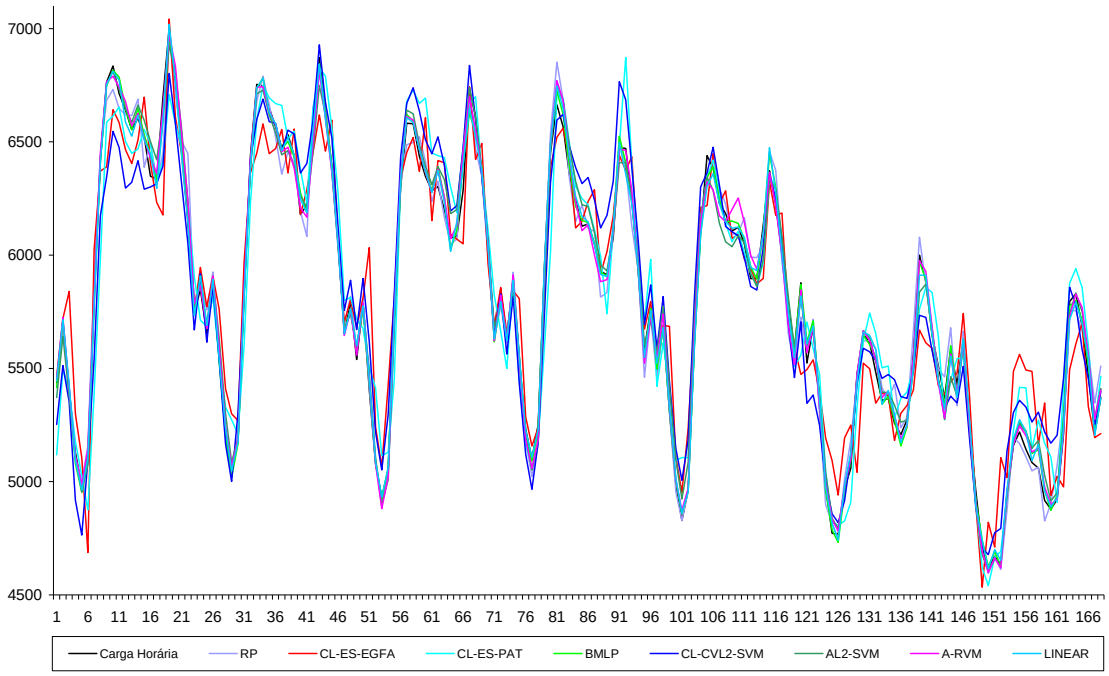

Figura 6.9 – Previsões de carga horária realizadas 1 passo à frente para o caso 3, cobrindo o período de 1/9/2003 a 7/9/2003

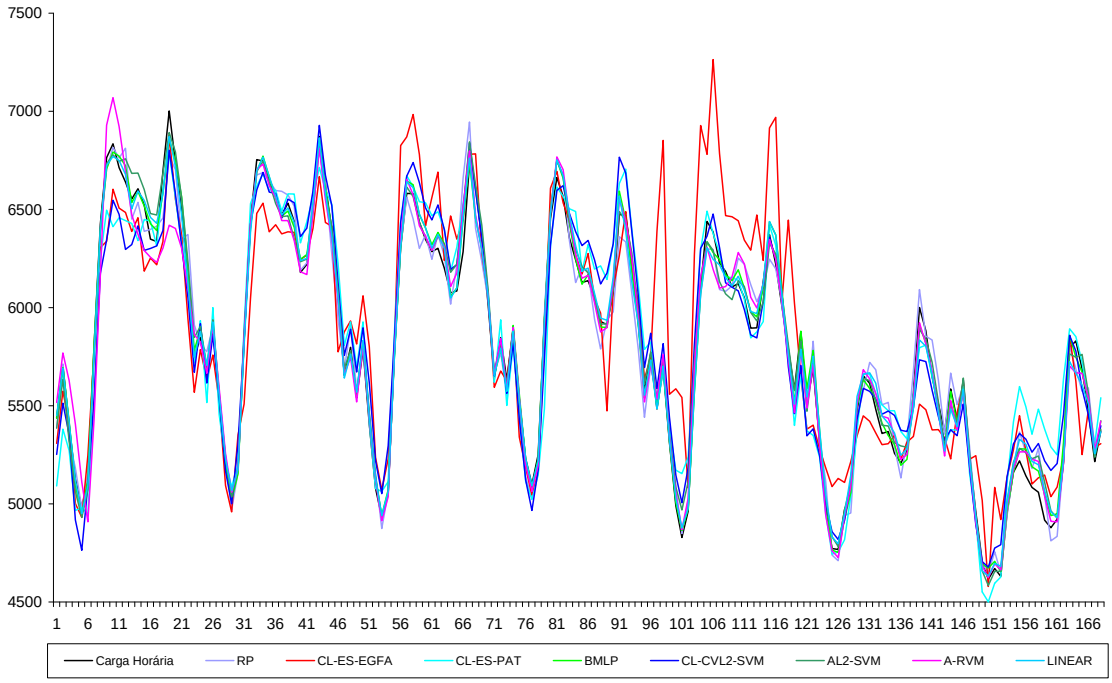

Figura 6.10 – Previsões de carga horária realizadas 2 passos à frente para o caso 3, cobrindo o período de 1/9/2003 a 7/9/2003

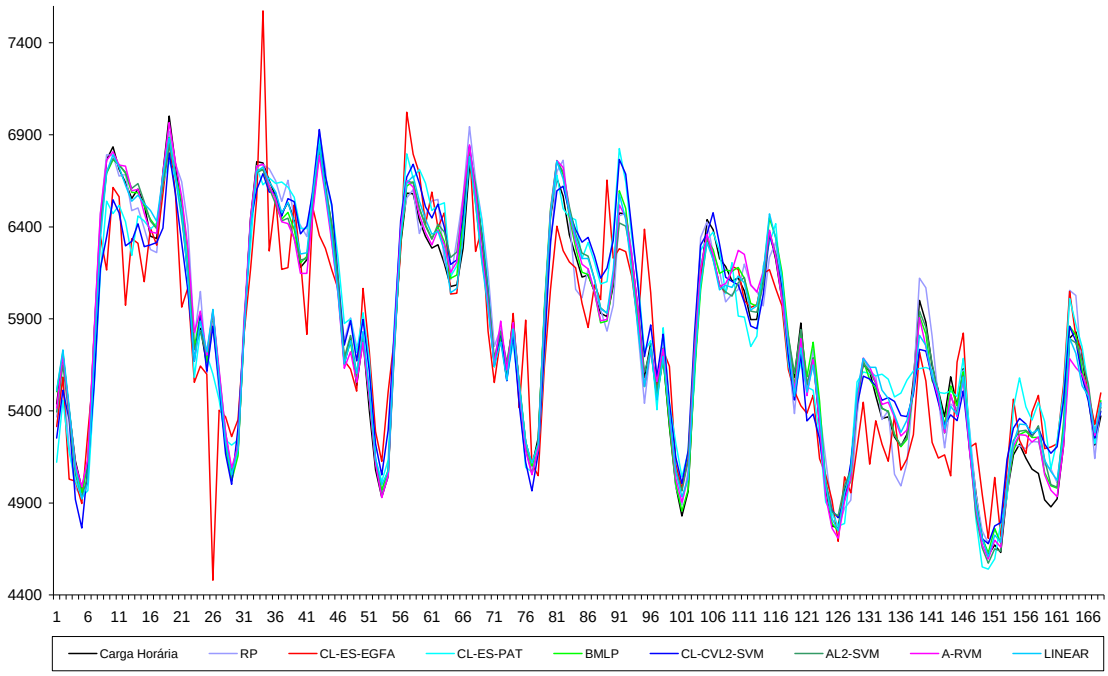

Figura 6.11 – Previsões de carga horária realizadas 3 passos à frente para o caso 3, cobrindo o período de 1/9/2003 a 7/9/2003

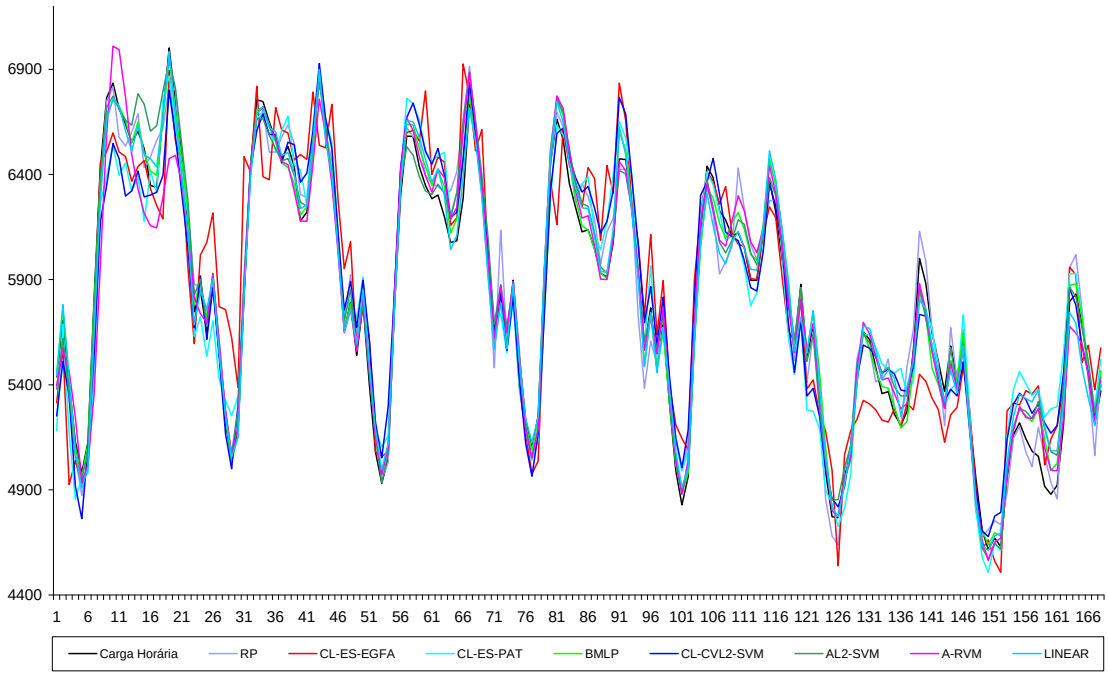

Figura 6.12 – Previsões de carga horária realizadas 4 passos à frente para o caso 3, cobrindo o período de 1/9/2003 a 7/9/2003

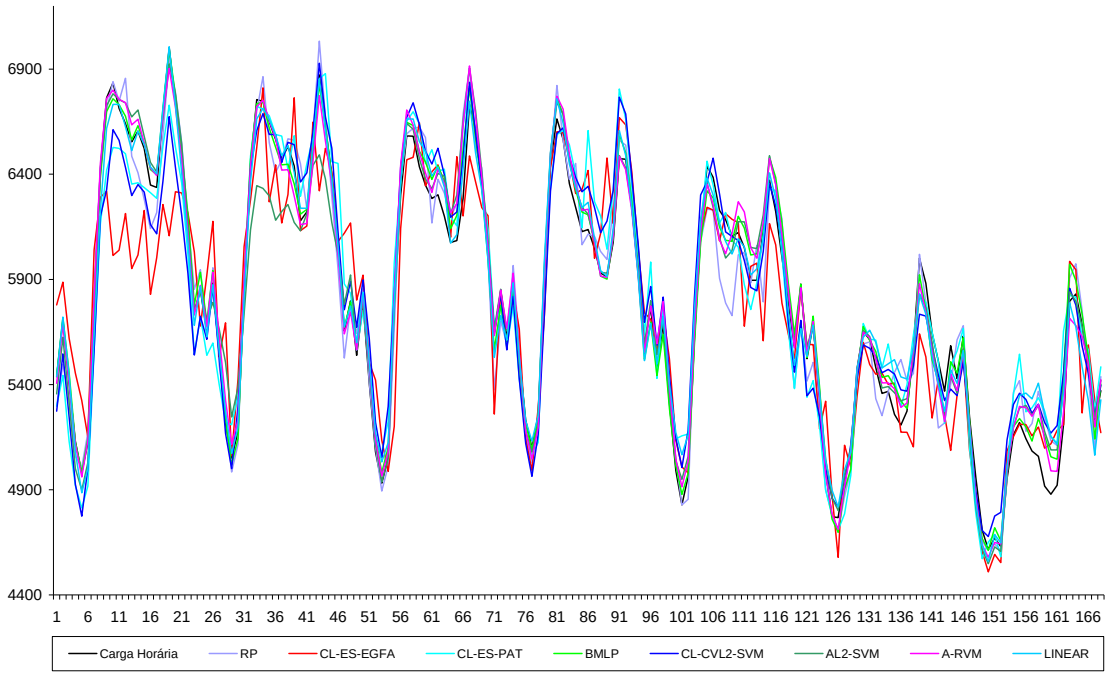

Figura 6.13 – Previsões de carga horária realizadas 5 passos à frente para o caso 3, cobrindo o período de 1/9/2003 a 7/9/2003

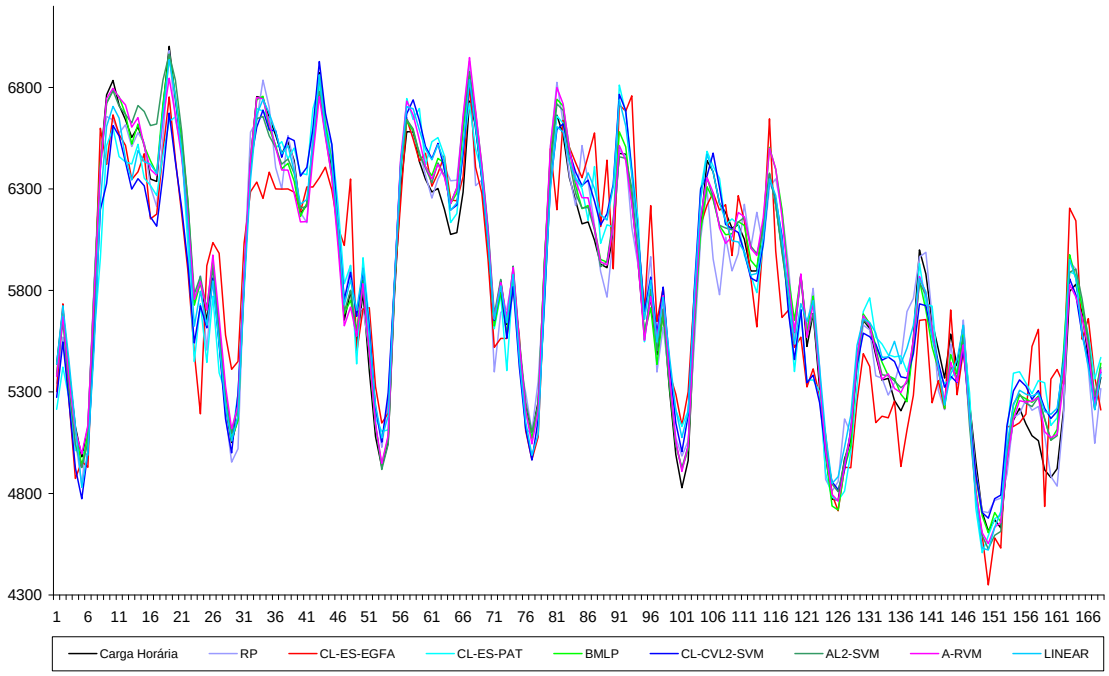

Figura 6.14 – Previsões de carga horária realizadas 6 passos à frente para o caso 3, cobrindo o período de 1/9/2003 a 7/9/2003

A eficiência das técnicas de seleção de entradas pode ser avaliada através da análise da Tabela 6.3. Nesta Tabela é apresentado o número médio de entradas utilizadas por cada modelo. De forma mais clara, todos os métodos são alimentados originalmente pelas entradas especificadas na seção 6.1. Visto que algumas técnicas possuem procedimentos de seleção de variáveis de entrada, a Tabela 6.3 mostra o número médio de sinais selecionados. O cálculo da média é necessário devido à utilização de diversos modelos para cada caso, além de no caso 1 serem realizadas previsões para várias semanas. Para ilustrar a variação no número de entradas selecionadas, a Tabela 6.4 apresenta o desvio padrão do número de entradas selecionado para cada metodologia.

Tabela 6.3 – Número médio de entradas utilizadas por cada método

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
RP	84	49	79	77	75	73	71	69
EGFA	84	49	79	77	75	73	71	69
ES-EGFA	84	49	79	77	75	73	71	69
CL-ES-EGFA	26	20	27	26	26	26	26	26
PAT	84	49	79	77	75	73	71	69
ES-PAT	84	49	79	77	75	73	71	69
CL-ES-PAT	26	20	27	26	26	26	26	26
BMLP	70	40	66	67	63	51	60	56
L2-SVM	84	49	79	77	75	73	71	69
CVL2-SVM	84	49	79	77	75	73	71	69
CL-CVL2-SVM	26	20	27	26	26	26	26	26
AL2-SVM	76	45	73	71	71	61	60	65
A-RVM	84	49	79	77	75	73	71	69
LINEAR	33	20	32	30	30	29	29	27
Redução (%)	68,55	58,99	66,37	66,79	65,90	64,97	63,78	62,73

O estudo da Tabela 6.3 mostra que a inferência *bayesiana* aplicada ao desenvolvimento de MLPs (BMLP) apresenta maior capacidade de redução da dimensionalidade do espaço de entrada em relação às técnicas automáticas de especificação de L2-SVMs (AL2-SVM) e de RVMs (A-RVM). Especificamente para a A-RVM, a otimização dos hiperparâmetros η_k que ponderam diretamente cada entrada não permitiu a detecção de sinais irrelevantes para nenhum dos casos estudados. Este

fato vai de encontro ao exposto por TIPPING [63], que destaca a dificuldade em combinar de forma efetiva a otimização dos hiperparâmetros $\underline{\alpha}$ e σ^2 e do hiperparâmetro η_k do *kernel*. Em outras palavras, o uso de uma única iteração para atualização de η_k via subida em gradiente combinada com a otimização de $\underline{\alpha}$ e σ^2 utilizando as expressões analíticas dadas em (5.79) a (5.81) não conduziu a alterações significativas em η_k que possibilitassem a identificação de variáveis irrelevantes por meio da comparação dos ponderadors otimizados η_k com aqueles obtidos para as variáveis de prova.

Tabela 6.4 – Desvio padrão do número de entradas utilizadas por cada método

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
RP	0	0	0	0	0	0	0	0
EGFA	0	0	0	0	0	0	0	0
ES-EGFA	0	0	0	0	0	0	0	0
CL-ES-EGFA	1	1	1	1	1	1	1	1
PAT	0	0	0	0	0	0	0	0
ES-PAT	0	0	0	0	0	0	0	0
CL-ES-PAT	1	1	1	1	1	1	1	1
BMLP	12	9	9	6	5	15	10	9
L2-SVM	0	0	0	0	0	0	0	0
CVL2-SVM	0	0	0	0	0	0	0	0
CL-CVL2-SVM	1	1	1	1	1	1	1	1
AL2-SVM	11	8	1	3	4	12	16	5
A-RVM	0	0	0	0	0	0	0	0
LINEAR	3	1	1	1	1	1	1	1

Quando comparadas com o método baseado em análise de índices de correlação linear (CL), as técnicas propostas na tese são bem menos efetivas no que diz respeito à redução da dimensionalidade do espaço de entrada. Entretanto, este poder de redução traz consigo a degradação do desempenho de previsão, como mostra a Tabela 6.1. Este fato encontra explicação na abdicação do modelo de previsão no processo de seleção de entradas utilizado pela análise de índices de correlação. Ao considerar o relacionamento somente entre pares de sinais, interdependências de ordem superior podem ser desconsideradas. Em outras palavras, sinais que individualmente são descorrelacionados

com a saída podem ser vitais para explicação desta em conjunto com outros. De maneira análoga, a utilização de sinais redundantes pode contribuir de maneira mais decisiva para a modelagem da saída do que o uso de um deles somente. Lembrando que variáveis discretas são desconsideradas nesta análise e que o número de entradas deste tipo utilizadas pelos modelos são respectivamente iguais a 24, 19 e 24 para os casos 1, 2 e 3, a Tabela 6.3 mostra que, em média, no máximo 3 entradas contínuas foram selecionadas (caso 3, 1 passo a frente). Observando as variáveis desta natureza especificadas na seção 6.1, níveis de redundância linear são claramente esperados. Tomando por exemplo as entradas relacionadas à carga, se $L(k)$ está correlacionado com $L(k-1)$, todos os atrasos consecutivos irão apresentar redundância, sendo retirados do modelo final. Apesar de redundantes, estes sinais podem contribuir de forma conjunta para estimação de $L(k)$, e efetivamente contribuem como mostra a Tabela 6.1. Esta possível dependência conjunta é considerada nos métodos acoplados de seleção de entradas, como os utilizados pelo BMLP, pela AL2-SVM e pela A-RVM, em função do uso de todo o espaço de representação disponibilizado em conjunto com o modelo. Assim, são descartadas somente entradas que contribuam de maneira ínfima para a estimação da saída, com o nível de irrelevância sendo definido pelas variáveis de prova auxiliares. Ao contrário da CL, entradas redundantes eventualmente podem ser selecionadas, desde que contribuam minimamente para a modelagem do sinal de saída.

O maior índice de redução obtido pela CL também é explicado pela sua capacidade de capturar somente relacionamentos de natureza linear. Para todos os casos, foram selecionadas somente entradas contínuas relacionadas a valores atrasados da própria série de carga. Na esmagadora maioria dos casos, estas entradas corresponderam às sazonalidades verificadas nas séries, ou seja, $L(k-24)$ e $L(k-168)$, para os casos

1 e 3, e $L(k-7)$, para o caso 2, além de $L(k-1)$. As variáveis exógenas utilizadas, contendo informações sobre temperatura e preço da energia, foram descartadas em virtude da conhecida influência não-linear destes sinais na dinâmica de curto prazo da série de carga. Conforme mostra a Tabela 6.1, o negligenciamento destas informações provocou severas reduções no desempenho de previsão de todos os modelos que utilizaram esta técnica de seleção de entradas, mostrando a importância destas variáveis na modelagem.

A discussão anterior sobre redundância, também conhecida como multicolinearidade, e relação linear entre entrada e saída fornece embasamento para o número reduzido de variáveis selecionadas pelo teste dos multiplicadores de *Lagrange* aplicado aos modelos lineares (LINEAR).

Outra questão que deve ser destacada diz respeito à forma na qual as entradas são selecionadas pelos métodos automáticos propostos. Lembrando das variáveis de prova auxiliares, são descartadas do modelo final somente as variáveis que contribuem menos para o cálculo da saída do que a variável sabidamente descorrelacionada com aquela. Esta contribuição é mensurada através de indicadores otimizados que controlam a magnitude da ligação de cada entrada ao modelo, ou seja, o hiperparâmetro α_i para o BMLP, e os ponderadores σ_i para SVM e η_k para o as RVMs. Em outras palavras, apesar de retirar poucas entradas, estes métodos possuem mecanismos para ponderar cada entrada de acordo com o seu grau de importância para o cálculo da saída. Assim, se uma variável apresentar relevância superior do que o sinal de prova, porém ainda for irrelevante para modelagem da saída, o indicador associado a esta entrada apresentará pequena magnitude, reduzindo a sua contribuição no cálculo da saída apesar de ainda fazer parte do modelo final.

Na Tabela 6.5 são apresentadas as estruturas utilizadas por cada método, em termos do número de neurônios na camada oculta dos MLPs, de vetores suporte para SVMs e vetores relevantes para RVMs. Visto que para o modelo linear utilizado a estrutura é expressa pelo número de entradas mais um parâmetro de intercepto, esta informação será omitida da Tabela 6.5, já que a primeira pode ser obtida diretamente da Tabela 6.3. De maneira análoga à Tabela 6.3, são mostrados valores médios, visto que para cada caso são desenvolvidos diversos modelos, sem contar o caso 1 onde o horizonte de previsão exige a estimação da curva de carga para várias semanas. Da mesma forma, a Tabela 6.6 apresenta o desvio padrão do numero médio de neurônios e vetores suporte ou relevantes obtidos.

Tabela 6.5 – Número médio de neurônios, vetores suporte e vetores relevantes utilizados por cada método

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
RP	10	10	10	10	10	10	10	10
EGFA	10	10	10	10	10	10	10	10
ES-EGFA	8	1	7	9	8	9	9	9
CL-ES-EGFA	6	2	4	6	6	5	6	6
PAT	10	10	10	10	10	10	10	10
ES-PAT	8	8	8	8	8	8	8	8
CL-ES-PAT	8	6	7	5	7	8	7	6
BMLP	8	7	7	8	7	5	5	3
L2-SVM	642	669	523	519	515	514	509	509
CVL2-SVM	428	464	344	338	328	330	333	347
CL-CVL2-SVM	425	464	347	348	346	344	342	341
AL2-SVM	642	707	518	515	509	513	510	505
A-RVM	112	34	89	75	78	56	61	53

Esta Tabela mostra a estrutura do MLP treinado através do algoritmo de retropropagação do erro (RP) original. A escolha de uma estrutura excessivamente parametrizada (861 parâmetros livres para o caso 1, com cerca de 650 padrões disponíveis para treinamento) teve por objetivo ilustrar a necessidade do controle de complexidade dos modelos. Esta mesma estrutura foi estimada com escalonamento do ganho da função de ativação e através de parada antecipada do treinamento, visando

verificar a capacidade de regularização destas heurísticas. Conforme esperado, para a maioria dos casos, o melhor resultado obtido por estas técnicas utilizou também estabilização de estrutura, evidenciando a importância da escolha do modelo em conjunto com a regularização da estrutura. Em relação aos modelos baseados em *kernel* (SVM e RVM), as máquinas de vetores relevantes apresentaram estruturas mais esparsas do que as SVMs. Em termos percentuais, a relação entre o número de vetores relevantes e vetores suporte variou de cerca de 5 % (caso 2) a 17 % (caso 1). Exemplificando, o número médio de vetores relevantes para o caso 2 representou cerca de 5 % do número médio de vetores suporte obtidos para este caso. Este resultado corrobora a característica das RVMs de promover representações mais esparsas quando comparadas com as SVMs, sem comprometer a capacidade de modelagem em termos de precisão das previsões.

Tabela 6.6 – Desvio padrão do número de neurônios, vetores suporte e vetores relevantes utilizados por cada método

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
RP	0	0	0	0	0	0	0	0
EGFA	0	0	0	0	0	0	0	0
ES-EGFA	4	1	4	1	3	1	1	1
CL-ES-EGFA	3	1	2	2	3	2	3	2
PAT	0	0	0	0	0	0	0	0
ES-PAT	4	2	1	1	1	3	2	2
CL-ES-PAT	3	3	2	2	3	2	2	2
BMLP	2	3	3	3	2	4	2	1
L2-SVM	9	11	2	2	3	6	7	6
CVL2-SVM	20	15	8	13	12	16	18	15
CL-CVL2-SVM	9	19	6	5	7	8	10	10
AL2-SVM	10	9	2	2	3	6	7	8
A-RVM	10	10	8	32	6	24	6	8

A Tabela 6.7 mostra o esforço computacional em minutos (min) necessário para estimação de cada uma das estruturas utilizadas, considerando um Processador Intel® Core™ 2 Duo 2,66 GHz, 3323 MB de Memória RAM com sistema operacional Windows Vista 32 Bits. Vale destacar que os valores apresentados nesta Tabela não

dizem respeito ao desenvolvimento de todo o modelo de previsão para o respectivo caso, mas sim ao tempo necessário para estimação de um único modelo. Para o caso 3, por exemplo, o BMLP necessitou de cerca de 10 minutos para especificação e estimação de cada um dos sete modelos desenvolvidos para cada passo à frente.

Tabela 6.7 – Esforço computacional de cada método (min)

	Caso 1	Caso 2	Caso 3
RP	0,83	0,83	0,88
EGFA	0,68	0,69	0,34
ES-EGFA	4,46	4,86	3,64
CL-ES-EGFA	3,99	4,82	3,73
PAT	1,23	1,33	1,13
ES-PAT	8,01	8,72	7,53
CL-ES-PAT	7,31	8,21	6,73
BMLP	7,15	2,97	9,87
L2-SVM	0,02	0,01	0,02
CVL2-SVM	11,45	15,54	7,43
CL-CVL2-SVM	44,14	27,40	50,45
AL2-SVM	7,28	0,70	3,36
A-RVM	27,81	3,72	8,26
LINEAR	0,01	0,01	0,00

Os resultados apresentados na Tabela 6.7 mostram que as técnicas automáticas propostas requerem maior esforço computacional em relação aos métodos comumente encontrados na literatura, com exceção do CL-CVL2-SVM. Visto que para SVMs a complexidade do modelo independente da dimensionalidade do espaço de entrada, o reduzido número de entradas contínuas selecionado pela análise de correlação linear (CL), comprometendo a capacidade de mapeamento do modelo, pode explicar o elevado tempo de processamento médio verificado para esta metodologia. Retornando para as técnicas propostas nesta tese, além de promovorem melhorias em termos de desempenho de previsão, o tempo dispendido por estes métodos não constitui um empecilho de ordem prática. Em que pese a configuração robusta do micro utilizado nas simulações, tomando o caso 1 como exemplo, onde o BMLP levou cerca de 7 minutos para obtenção de uma única estrutura, vale lembrar que este modelo individual está apto

para realização de previsões para todo o dia. Considerando computadores com configuração mais simples em que o tempo de processamento seja inferior na ordem de 10 vezes em relação ao utilizado no trabalho, o BMLP levaria cerca de 70 minutos para previsão da curva de carga diária em base horária, espaço de tempo que pode ser considerado razoável para aplicações práticas. Além disso, os algoritmos utilizados foram implementados em MATLAB, uma linguagem interpretada, sendo esperadas reduções no esforço computacional na medida em que estes métodos forem migrados para linguagens mais rápidas, como C++ por exemplo.

Este conjunto inicial de testes, considerando conjuntos de entrada e partições definidas pelo usuário, evidenciou o desempenho satisfatório obtido pela inferência *bayesiana* aplicada ao desenvolvimento de MLPs. Por outro lado as SVMs, apesar de atrativas sob o ponto de vista teórico visto que visam na sua formulação original à minimização do risco estrutural tendo por objetivo o desenvolvimento de modelos com considerável capacidade de generalização, não apresentaram resultados convincentes. As dificuldades encontradas no ajuste do algoritmo de minimização do limite superior do erro de generalização estimado por validação cruzada única, incluindo multimodalidade e sensibilidade a parâmetros como passo de atualização e η (constante definida pelo usuário e responsável pela diferenciabilidade de $\widetilde{\Psi}_i^2$), contribuem para justificar o desempenho inferior obtido por estes modelos. O desempenho de previsão inferior obtido pelas SVMs com hiperparâmetros estimados por validação cruzada evidencia a necessidade de ajuste adequado destas constantes, visto que este tipo de modelo apresentou o melhor resultado encontrado na literatura para o caso 2.

Ao utilizar inferência *bayesiana* na definição do modelo, as máquinas de vetores relevantes dão origem a modelos baseados em *kernel* com representação esparsa similar à obtida pelas SVMs. Além de possuirem metodologias automáticas para estimação dos

diversos hiperparâmetros envolvidos, as RVMs produzem estruturas ainda mais esparsas do que as SVMs sem deteriorar o desempenho de previsão, como mostrado na Tabela 6.1 e na Tabela 6.5. Desta forma, à luz dos resultados preliminares e tendo em mente o desenvolvimento de modelos neurais autônomos, as técnicas baseadas na teoria do caos para seleção do conjunto inicial foram aplicadas ao BMLP e à metodologia automática para especificação de RVMs.

Para utilização adequada da técnica de seleção do conjunto inicial resumida na seção 3.1.5, é necessária a identificação de traços caóticos nas séries envolvidas. Conforme apresentado no capítulo 3, o estudo do espectro de expoentes de *Lyapunov* de um dado sistema permite identificar a existência de características caóticas. Especificamente, sistemas caóticos apresentam expoente máximo positivo. Contudo, para estimação do espectro de *Lyapunov* a partir de um histórico de medições de um dado sistema, é necessária a reconstrução do espaço de fase, através do teorema de *Takens* por exemplo. Utilizando os métodos discutidos na seção 3.1, a Tabela 6.8 apresenta algumas características caóticas identificadas para cada uma das séries utilizadas no trabalho. Visto que para o caso 1 as previsões são realizadas de forma iterativa ao longo do período de 1º de novembro de 1990 a 31 de março de 1991, os valores apresentados nesta Tabela foram estimados considerando os históricos horários referentes ao período de 1º de janeiro de 1989 a 31 de outubro de 1990. Vale ressaltar que a cada sessão de treinamento, o conjunto inicial de entradas é obtido novamente por meio da estimação dos parâmetros de imersão e da avaliação da existência de sincronismo entre as séries.

Na Tabela 6.8 são apresentados os parâmetros da imersão obtidos para cada histórico de carga $L(k)$, temperatura $T(k)$ e temperatura ao quadrado $T^2(k)$, ou seja, dimensão d e atraso τ, juntamente com o expoente de *Lyapunov* máximo λ_1. Visto

que não são desenvolvidos modelos de previsão de preço, esta série não será utilizada neste conjunto de testes (vide equação (3.35)). Além destes parâmetros, são apresentadas informações relacionadas à avaliação do sincronismo entre as diversas séries consideradas em cada caso, como expoente de *Lyapunov* condicional máximo λ_1^R, o valor médio m_{medio} da estatística $m\left[\underline{x}(t), \underline{y}(t)\right]$ conhecida como falsos vizinhos mais próximos mútuos, a dimensão ν_Y do atrator no espaço de estados completo e a dimensão ν_X do atrator no espaço do sistema guia. Visto que cada caso apresenta o seu conjunto específico de séries exógenas, as colunas da Tabela 6.8 são preenchidas somente com as informações referentes às respectivas séries consideradas no caso em questão.

Tabela 6.8 – Características caóticas das séries analisadas nos três casos

		Caso 1			Caso 2			Caso 3		
		d	τ	λ_1	d	τ	λ_1	d	τ	λ_1
L(k)		10	6	0,0303	12	4	0,0146	12	13	0,0110
T(k)		18	13	0,0069	14	15	0,0048	19	13	0,0100
T^2(k)		17	13	0,0082	-	-	-	-	-	-
$\lambda^R{}_1$	T(k)	0,0159			-0,0002			0,0062		
	T^2(k)	0,0135			-			-		
$m_{médio}$	T(k)	2,9599			1,7232			1,9526		
	T^2(k)	2,9720			-			-		
ν_Y	T(k)	18,1330			9,5872			22,5674		
	T^2(k)	17,0550			-			-		
ν_X	T(k)	12,6600			8,2221			15,4746		
	T^2(k)	12,3830			-			-		

Os resultados apresentados na Tabela 6.8 sinalizam a existência de traços caóticos nas diversas séries envolvidas, visto que todas apresentam expoente máximo λ_1 positivo. Conforme esperado, os expoentes máximos obtidos a partir das séries $T(k)$ e $T^2(k)$ são próximos, visto que o espectro de expoentes de *Lyapunov* de um dado sistema dinâmico é uma das diversas medidas invariantes a transformações que

caracterizam os atratores de sistemas caóticos. Neste ponto vale destacar que a conclusão acerca das características caóticas está diretamente atrelada aos empecilhos verificados na estimação empírica do espectro de expoentes de *Lyapunov*, tais como sensibilidade à presença de ruído e existência de expoentes espúrios em função da sobreestimação da dimensão de imersão d. Estas restrições comprometem o uso da análise do expoente de *Lyapunov* condicional máximo λ_1^R como ferramenta exclusiva para identificação de sincronismo entre sistemas caóticos. Segundo este critério, à luz dos resultados apresentados na Tabela 6.8, nos casos 1 e 3 o sistema reconstruído a partir da série de carga não apresenta sincronismo com as dinâmicas oriundas das séries de temperatura, visto que para estes casos λ_1^R é positivo. Além disso, o valor negativo de pequena magnitude verificado para o caso 2 não confirma a existência de relacionamento entre carga e temperatura para esta base de dados.

A ausência de confirmação da existência de sincronismo a partir do estudo do expoente de *Lyapunov* condicional máximo evidencia a necessidade do uso de mais um critério para corroborar as conclusões, motivando o cálculo do valor médio m_{medio} da estatística $m\left[\underline{x}(t), \underline{y}(t)\right]$ conhecida como falsos vizinhos mais próximos mútuos. Conforme destacado na seção 3.1.3, sistemas em sincronismo generalizado apresentam m_{medio} próximo da unidade; por outro lado, sistemas caóticos independentes possuem m_{medio} tendendo ao infinito. A partir deste critério, os resultados da Tabela 6.8 sinalizam a existência de sincronismo generalizando entre as dinâmicas reconstruídas das séries de carga e temperatura para os três casos estudados.

A aparente divergência entre os métodos de identificação de sincronismo não constitui um empecilho para os modelos de previsão desenvolvidos neste trabalho. O uso posterior de metodologias encapsuladas para seleção de entradas permite identificar,

ao longo do desenvolvimento do modelo, variáveis irrelevantes eventualmente selecionadas na fase de definição do conjunto inicial. Desta forma, além da representação do espaço reconstruído a partir da série de carga, o conjunto inicial utilizará entradas relacionadas às variáveis exógenas cujo sincronismo com a dinâmica oriunda do histórico de carga for detectado ao menos por um dos métodos. Além disso, visto que a reconstrução por meio do teorema de *Takens* permite recuperar a dinâmica determinística do sistema caótico em estudo, o primeiro conjunto de testes utilizará somente variáveis contínuas. Assim, para o caso 1, o par entrada saída $\left(\underline{x}_k, d_k\right)$ será dado por:

$$\begin{aligned} \underline{x}_{k+1} &= \left[\underline{L}(k)^t \quad \underline{T}(k+1)^t \quad \underline{T}_2(k+1)^t\right]^t \\ d_{k+1} &= L(k+1) \end{aligned} \tag{6.23}$$

onde

$$\begin{aligned} \underline{L}(k) &= \left[L(k) \quad L(k-6) \quad \ldots \quad L(k-54)\right]^t \\ \underline{T}(k) &= \left[T(k) \quad T(k-13) \quad \ldots \quad T(k-221)\right]^t \\ \underline{T}_2(k) &= \left[T^2(k) \quad T^2(k-13) \quad \ldots \quad T(k-208)\right]^t \end{aligned} \tag{6.24}$$

Para o caso 2,

$$\begin{aligned} \underline{x}_{k+1} &= \left[\underline{L}(k)^t \quad \underline{T}(k+1)^t\right]^t \\ d_{k+1} &= L(k+1) \end{aligned} \tag{6.25}$$

onde

$$\begin{aligned} \underline{L}(k) &= \left[L(k) \quad L(k-4) \quad \ldots \quad L(k-44)\right]^t \\ \underline{T}(k) &= \left[T(k) \quad T(k-15) \quad \ldots \quad T(k-195)\right]^t \end{aligned} \tag{6.26}$$

Lembrando que para este caso, as informações de carga representam pico de carga diário, enquanto que as de temperatura representam temperatura média diária. Por fim, para o caso 3,

$$\underline{x}_{k+1} = \left[\underline{L}(k)^t \quad \underline{T}(k+1)^t \right]^t \tag{6.27}$$
$$d_{k+1} = L(k+1)$$

onde

$$\underline{L}(k) = \left[L(k) \quad L(k-13) \quad \ldots \quad L(k-143) \right]^t \tag{6.28}$$
$$\underline{T}(k) = \left[T(k) \quad T(k-13) \quad \ldots \quad T(k-234) \right]^t$$
$$\underline{P}(k) = \left[P(k) \quad P(k-8) \quad \ldots \quad T(k-104) \right]^t$$

O conjunto inicial de entradas descrito anteriormente apresenta algumas diferenças em relação à bateria de testes realizada anteriormente. Para o caso 1, a série de temperatura máxima diária não pode ser utilizada em virtude da diferença entre os períodos de amostragem desta série e do histórico horário. Por outro lado, a ausência de modelos de previsão de preço impossibilita o uso desta informação como entrada dos modelos desenvolvidos para o caso 3. Para os demais casos, o uso do teorema de *Takens* e do conceito de sincronismo generalizado entre sistemas (vide equações (3.1), (3.3) e (3.35)) inviabiliza o uso de um modelo para cada passo à frente conforme utilizado anteriormente. Desta forma, as previsões para os s passos à frente em cada caso são realizadas por meio de recursão. Por fim, são desenvolvidos modelos locais a partir das partições utilizadas no primeiro conjunto de simulações e descritas na seção 6.1.

A Tabela 6.9 e a Tabela 6.10 apresentam os resultados obtidos, em termos do erro absoluto percentual médio (MAPE) e do erro absoluto percentual (MAE e MAE%) respectivamente, considerando os conjuntos iniciais de entradas especificados nas equações (6.23) a (6.28). Visando avaliar o desempenho da técnica de seleção do conjunto inicial de entradas à luz dos resultados obtidos para a primeira bateria de testes, inicialmente esta metodologia foi aplicada somente ao BMLP.

Os resultados apresentados nestas Tabelas são desanimadores. Apesar da elevação do nível de automatização do processo, a deterioração do desempenho de

previsão foi flagrante. Na melhor situação (caso 3, terceiro passo à frente), o MAPE verificado foi cerca de 88 % superior ao melhor resultado encontrado na literatura. Em termos de erro máximo, a menor redução de desempenho foi da ordem de 43 % (caso 2, primeiro passo à frente). Tendo em mente os resultados animadores verificados inicialmente, a representação inadequada das sazonalidades envolvidas pode justificar o desempenho pífio evidenciado na Tabela 6.9 e na Tabela 6.10. Em outras palavras, a reconstrução do espaço de estados por meio do teorema de *Takens* não foi capaz de modelar de forma adequada as sazonalidades.

Tabela 6.9 – Desempenho dos modelos utilizando teoria do caos para seleção do conjunto inicial de entradas (MAPE)

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
BMLP	11,62	4,37	1,20	1,70	1,88	2,41	2,28	2,66
Benchmark	4,73	1,98	0,56	0,83	1,00	1,15	1,20	1,30
Ganho (%)	-145,62	-120,35	-114,80	-104,83	-87,67	-109,80	-90,02	-104,92

Tabela 6.10 – Desempenho dos modelos utilizando teoria do caos para seleção do conjunto inicial de entradas (MAE e MAE%)

	Caso 1 (%)	Caso 2 (MW)	Caso 3 (%)					
			1 passos	2 passos	3 passos	4 passos	5 passos	6 passos
BMLP	107,56	85,54	4,64	5,07	5,79	10,14	8,49	12,56
Benchmark	-	51,42	3,24	3,43	4,11	3,87	5,57	5,20
Ganho (%)	-	-66,37	-43,17	-47,71	-40,89	-161,90	-52,36	-141,57

Visando verificar a veracidade das conclusões acima em busca da melhoria do desempenho de previsão, a representação binária das sazonalidades utilizada no primeiro conjunto de simulações foi incorporada ao conjunto inicial de entradas. Considerando históricos de carga em base horária e diária, esta especificação não chega a comprometer o nível de automatização do processo, visto que os períodos das sazonalidades existentes nestas séries são de amplo conhecimento. Desta forma, além do conjunto de entradas contínuas listado nas equações (6.23) a (6.28), são acrescentadas mais 24, 19 e 24 entradas binárias para os casos 1, 2 e 3, respectivamente,

sendo obtidos os resultados apresentados na Tabela 6.11 e na Tabela 6.12. Este conjunto inicial de entradas ampliado foi aplicado ao BMLP (CHAOS-BMLP) e à RVM (CHAOS-RVM). Em relação aos resultados apresentados na Tabela 6.9e na Tabela 6.10, o desempenho do BMLP apresentou melhora considerável, evidenciando a incapacidade da representação em espaço de estados via teorema de *Takens* em capturar de maneira satisfatória as sazonalidades existentes nas séries de carga consideradas neste trabalho. De outra forma, visto que a reconstrução da dinâmica a partir de dados históricos visa identificar relacionamentos determinísticos, a melhoria de desempenho em função da representação explícita de componentes sazonais pode sinalizar a inexistência de relacionamento determinístico entre a carga e suas parcelas sazonais sabidamente existentes. Para o caso 1 em específico, os resultados obtidos tanto para o BMLP quanto para a RVM foram similares aos verificados nos testes considerando o conjunto inicial definido pelo usuário, apresentados na Tabela 6.1 e na Tabela 6.2. Além de um maior nível de automatização, estes resultados prescindiram da informação sobre a temperatura máxima diária utilizada nos testes preliminares. Para os casos 2 e 3, o uso de recursão deteriorou o desempenho dos modelos, promovendo reduções que variaram de 14,77 a 56,08% em relação aos *benchmarks* encontrados na literatura. Vale destacar novamente o grau de automatização considerado nas metodologias propostas no trabalho, contrastando com o elevado nível de conhecimento de especialistas dispendido no ajuste dos modelos referenciados na literatura.

Tabela 6.11 – Desempenho dos modelos utilizando teoria do caos para seleção do conjunto inicial de entradas em conjunto com variáveis binárias (MAPE)

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
CHAOS-BMLP	4,83	3,25	0,64	1,02	1,55	1,69	1,87	1,88
CHAOS-RVM	8,64	3,00	1,09	1,80	2,10	2,29	2,72	2,94
Benchmark	4,73	1,98	0,56	0,83	1,00	1,15	1,20	1,30
Ganho (%)	-2,11	-51,27	-14,77	-23,01	-54,55	-46,54	-56,08	-44,75

Tabela 6.12 – Desempenho dos modelos utilizando teoria do caos para seleção do conjunto inicial de entradas em conjunto com variáveis binárias (MAE e MAE%)

	Caso 1 (%)	Caso 2 (MW)	Caso 3 (%)					
			1 passos	2 passos	3 passos	4 passos	5 passos	6 passos
CHAOS-BMLP	41,23	81,45	3,59	6,34	10,36	10,45	7,63	8,34
CHAOS-RVM	64,65	55,85	4,32	5,64	6,65	7,01	9,85	13,38
Benchmark	-	51,42	3,24	3,43	4,11	3,87	5,57	5,20
Ganho (%)	-	-8,63	-10,90	-64,48	-61,83	-81,16	-36,96	-60,45

Para ilustrar o desempenho dos modelos utilizando o conjunto inicial de entradas definido via teoria do caos em conjunto com representação binária (1 de n) das sazonalidades, a Tabela 6.13 apresenta algumas estatísticas sobre o erro absoluto percentual verificado para cada hora ao longo do horizonte de previsão para o caso 1. Para cada hora, é apresentado o valor médio, o desvio padrão e o valor máximo verificado para o erro absoluto percentual. A Figura 6.15 ilustra a diferença entre o erro absoluto percentual médio (MAPE) verificado para cada hora para os dois métodos desenvolvidos neste trabalho. Enquanto que para o BMLP o comportamento do MAPE é regular em torno de 5 %, para a RVM esta estatística varia ao longo do dia entre 7,5 e 10 %.

Finalizando a apresentação dos resultados sobre este conjunto de simulações, a Tabela 6.14 apresenta o número de entradas selecionadas pelas duas metodologias encapsuladas propostas no trabalho. Acompanhando os resultados apresentados na Tabela 6.3, novamente as técnicas promoveram pequenas reduções de dimensionalidade, com as RVMs novamente não detectando entradas irrelevantes. Vale destacar a otimização dos hiperparâmetros diretamente relacionados com as entradas permite ponderar com maior intensidade os sinais mais relevantes para o cálculo da saída apesar da não retirada explícita de variáveis eventualmente irrelevantes. Por fim, a Tabela 6.15 destaca as estruturas obtidas para os três casos estudados, em termos de número de neurônios e vetores relevantes.

Tabela 6.13 – Desempenho dos modelos para cada hora do dia para o caso 1

Hora	CHAOS-BMLP			CHAOS-RVM		
	Média	Desvio Padrão	Máximo	Média	Desvio Padrão	Máximo
1	5.22	4.06	19.11	7.81	6.76	32.37
2	5.22	4.14	21.31	8.25	7.23	31.26
3	5.14	4.31	22.97	8.79	8.11	47.10
4	5.21	4.59	25.85	9.73	9.13	62.95
5	5.44	4.76	25.88	10.14	9.69	64.65
6	5.49	5.11	26.14	9.83	8.72	44.87
7	5.50	5.82	41.23	9.05	8.18	38.06
8	4.90	5.33	34.02	8.13	8.55	42.40
9	4.03	4.12	24.48	7.45	9.13	46.60
10	3.72	3.96	22.59	7.18	9.71	48.68
11	3.95	3.97	20.36	7.46	9.71	47.25
12	4.23	3.89	18.92	7.66	9.33	45.62
13	4.57	3.86	19.03	7.95	9.06	45.57
14	4.79	3.84	17.68	8.11	8.87	44.76
15	5.04	3.92	16.53	8.45	8.78	44.16
16	5.20	3.98	18.66	9.00	8.81	42.77
17	5.65	4.30	24.31	10.00	9.52	44.04
18	5.35	4.54	29.92	10.21	10.15	46.97
19	4.31	4.28	32.56	9.56	10.20	47.13
20	4.15	3.97	31.37	9.21	9.90	46.10
21	4.29	3.85	25.78	8.89	9.48	45.21
22	4.40	3.81	19.39	8.32	9.04	43.61
23	4.75	4.00	20.30	7.96	8.35	40.46
24	5.38	4.32	20.23	8.33	8.05	35.95

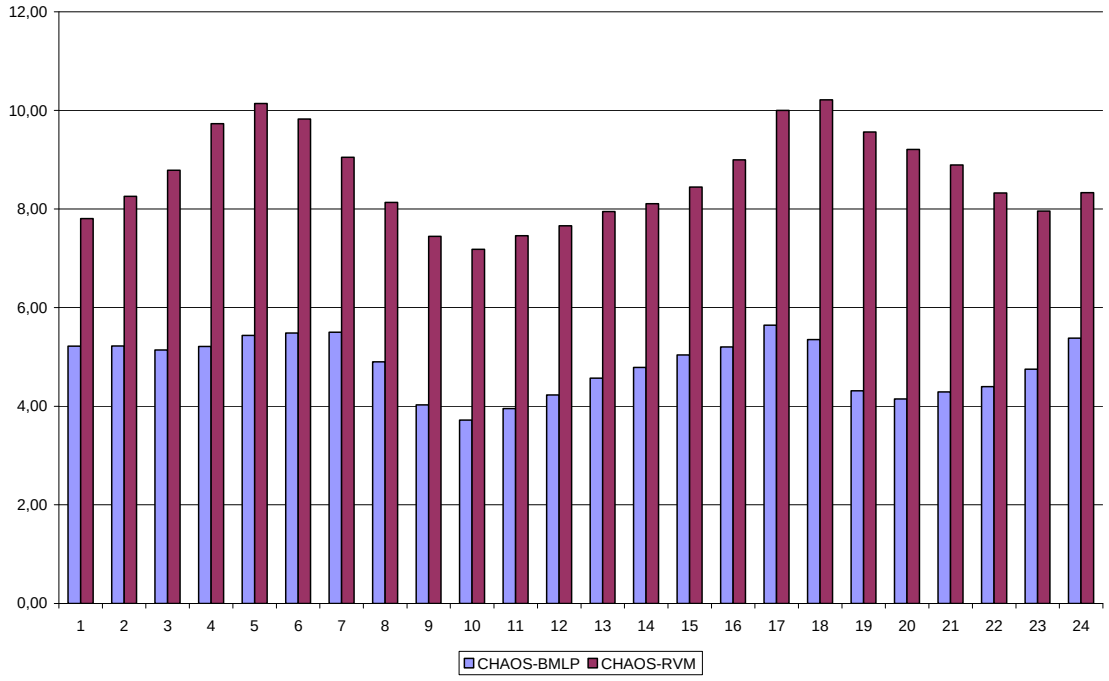

Figura 6.15 – Comparativo entre o erro absoluto percentual médio verificado para cada hora ao longo do horizonte de previsão do caso 1

Tabela 6.14 – Número de entradas selecionadas pelos diferentes métodos considerando teoria do caos para seleção do conjunto inicial de entradas acrescido de representação binária das sazonalidades

	Caso 1	Caso 2	Caso 3
CHAOS-BMLP	62	34	47
CHAOS-RVM	69	45	55
Número inicial	69	45	55
Redução	10,5	24,4	14,3

Tabela 6.15 – Estruturas (número de neurônios e de vetores relevantes) selecionadas pelos diferentes métodos considerando teoria do caos para seleção do conjunto inicial de entradas acrescido de representação binária das sazonalidades

	Caso 1	Caso 2	Caso 3
CHAOS-BMLP	5	1	7
CHAOS-RVM	112	28	73

O uso do conceito de sincronismo generalizado para definição do conjunto inicial de entradas requer o conhecimento do comportamento futuro da série guia. Neste trabalho, as séries de temperatura horária e diária foram utilizadas neste sentido, sendo inicialmente suposto como previsão o valor efetivamente verificado. Tal suposição não corresponde à realidade, visto que até os mais precisos sistemas de previsão meteorológica trazem consigo erros de previsão. Desta forma, visando obter resultados mais fidedignos com a eventual utilização prática das metodologias propostas, a Tabela 6.16 e a Tabela 6.17 apresentam os resultados obtidos para o CHAOS-BMLP e para CHAOS-RVM considerando previsões de temperatura, geradas de formas distintas para cada caso. Para o caso 1 a última curva de temperatura horária disponível na base de dados para treinamento foi considerada como previsão. Para o caso 2, a temperatura média diária para janeiro de 1999 foi obtida a partir da média verificada em 1997 e 1998 para o mesmo mês. Finalmente para o caso 3, a última temperatura verificada foi utilizada como previsão para os seis passos à frente analisados. A Por fim, visando

identificar a importância da seleção adequada tanto do conjunto de entradas quanto da estrutura, as Tabela 6.19 a Tabela 6.22 apresentam resultados obtidos a partir da seleção única do conjunto de entradas ou da estrutura. Em outras palavras, a Tabela 6.19 e a

Comparando com a Tabela 6.11 e com a Tabela 6.12, os resultados apresentados nas Tabela 6.19 a Tabela 6.22 mostram a importância da seleção conjunta tanto do espaço de entrada quanto da estrutura, visto que na maioria dos casos o desempenho foi reduzido em virtude da fixação do conjunto de entradas ou da estrutura. Por outro lado, a queda de desempenho não foi tão acentuada, evidenciando tanto a capacidade de ponderação das entradas menos relevantes quando fixado o conjunto de entradas quanto o controle de complexidade da estrutura fixada e eventualmente sobredimensionada. Estas duas questões contribuem sobremaneira para o desempenho superior obtido pelo CHAOS-BMLP, que apesar de não obter resultados superiores aos melhores encontrados na literatura, traz consigo um elevado grau de automatização do processo de modelagem em conjunto com resultados satisfatórios.

Tabela 6.20 ilustram o desempenho do CHAOS-BMLP considerando uma estrutura fixa com 10 neurônios na camada oculta, sendo selecionado via maximização da evidência o conjunto de entradas. De outra forma, a Tabela 6.21 e a Tabela 6.22 apresentam os resultados obtidos a partir da fixação do conjunto inicial de entradas definido via teoria do caos e representação binária das sazonalidades, sendo selecionado por meio da maximização da evidência o número de neurônios do modelo. Diante da seleção automática de estrutura intrínseca às RVMs em conjunto com a incapacidade de redução da dimensionalidade do espaço de entrada verificada pelo CHAOS-RVM, este conjunto de testes foi aplicado somente ao CHAOS-BMLP.

Tabela 6.18 apresenta estatísticas do erro absoluto, em [°C], apurado para os diferentes esquemas de previsão. A ocorrência de temperaturas nulas inviabiliza o cálculo do erro percentual.

Os resultados da Tabela 6.16 e da Tabela 6.17 mostram a robustez das metodologias desenvolvidas em relação aos erros na previsão de temperatura. Comparando com a Tabela 6.11 e com a Tabela 6.12, a queda de desempenho foi pequena, com a CHAOS-RVM apresentando até mesmo alguns resultados superiores, como para o caso 2 e alguns passos à frente do caso 3. Este resultado satisfatório em termos de manutenção de desempenho pode ser explicado pelo reduzido erro médio verificado pelos esquemas de previsão de temperatura utilizados, destacados na Por fim, visando identificar a importância da seleção adequada tanto do conjunto de entradas quanto da estrutura, as Tabela 6.19 a Tabela 6.22 apresentam resultados obtidos a partir da seleção única do conjunto de entradas ou da estrutura. Em outras palavras, a Tabela 6.19 e a Comparando com a Tabela 6.11 e com a Tabela 6.12, os resultados apresentados nas Tabela 6.19 a Tabela 6.22 mostram a importância da seleção conjunta tanto do espaço de entrada quanto da estrutura, visto que na maioria dos casos o desempenho foi reduzido em virtude da fixação do conjunto de entradas ou da estrutura. Por outro lado, a queda de desempenho não foi tão acentuada, evidenciando tanto a capacidade de ponderação das entradas menos relevantes quando fixado o conjunto de entradas quanto o controle de complexidade da estrutura fixada e eventualmente sobredimensionada. Estas duas questões contribuem sobremaneira para o desempenho superior obtido pelo CHAOS-BMLP, que apesar de não obter resultados superiores aos melhores encontrados na literatura, traz consigo um elevado grau de automatização do processo de modelagem em conjunto com resultados satisfatórios.

Tabela 6.20 ilustram o desempenho do CHAOS-BMLP considerando uma estrutura fixa com 10 neurônios na camada oculta, sendo selecionado via maximização da evidência o conjunto de entradas. De outra forma, a Tabela 6.21 e a Tabela 6.22 apresentam os resultados obtidos a partir da fixação do conjunto inicial de entradas definido via teoria do caos e representação binária das sazonalidades, sendo selecionado por meio da maximização da evidência o número de neurônios do modelo. Diante da seleção automática de estrutura intrínseca às RVMs em conjunto com a incapacidade de redução da dimensionalidade do espaço de entrada verificada pelo CHAOS-RVM, este conjunto de testes foi aplicado somente ao CHAOS-BMLP.

Tabela 6.18.

Tabela 6.16 – Desempenho dos modelos utilizando teoria do caos para seleção do conjunto inicial de entradas em conjunto com variáveis binárias (MAPE), considerando previsões de temperatura

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
CHAOS-BMLP	5.50	3.47	0.64	1.08	1.45	1.81	2.29	2.30
CHAOS-RVM	8.99	2.33	0.92	1.49	1.79	2.26	2.57	3.03
Benchmark	4.73	1.98	0.56	0.83	1.00	1.15	1.20	1.30
Ganho (%)	-16.36	-17.44	-15.14	-30.66	-45.44	-57.21	-91.10	-76.58

Tabela 6.17 – Desempenho dos modelos utilizando teoria do caos para seleção do conjunto inicial de entradas em conjunto com variáveis binárias (MAE e MAE%), considerando previsões de temperatura

	Caso 1 (%)	Caso 2 (MW)	Caso 3 (%)					
			1 passos	2 passos	3 passos	4 passos	5 passos	6 passos
CHAOS-BMLP	51.08	85.06	3.30	6.72	8.04	8.85	9.46	8.99
CHAOS-RVM	49.35	59.10	4.42	5.27	7.13	7.59	8.35	10.35
Benchmark	-	51.42	3.24	3.43	4.11	3.87	5.57	5.20
Ganho (%)	-	-14.94	-1.98	-53.55	-73.56	-96.19	-49.91	-72.82

Por fim, visando identificar a importância da seleção adequada tanto do conjunto de entradas quanto da estrutura, as Tabela 6.19 a Tabela 6.22 apresentam resultados obtidos a partir da seleção única do conjunto de entradas ou da estrutura. Em outras

palavras, a Tabela 6.19 e a Comparando com a Tabela 6.11 e com a Tabela 6.12, os resultados apresentados nas Tabela 6.19 a Tabela 6.22 mostram a importância da seleção conjunta tanto do espaço de entrada quanto da estrutura, visto que na maioria dos casos o desempenho foi reduzido em virtude da fixação do conjunto de entradas ou da estrutura. Por outro lado, a queda de desempenho não foi tão acentuada, evidenciando tanto a capacidade de ponderação das entradas menos relevantes quando fixado o conjunto de entradas quanto o controle de complexidade da estrutura fixada e eventualmente sobredimensionada. Estas duas questões contribuem sobremaneira para o desempenho superior obtido pelo CHAOS-BMLP, que apesar de não obter resultados superiores aos melhores encontrados na literatura, traz consigo um elevado grau de automatização do processo de modelagem em conjunto com resultados satisfatórios.

Tabela 6.20 ilustram o desempenho do CHAOS-BMLP considerando uma estrutura fixa com 10 neurônios na camada oculta, sendo selecionado via maximização da evidência o conjunto de entradas. De outra forma, a Tabela 6.21 e a Tabela 6.22 apresentam os resultados obtidos a partir da fixação do conjunto inicial de entradas definido via teoria do caos e representação binária das sazonalidades, sendo selecionado por meio da maximização da evidência o número de neurônios do modelo. Diante da seleção automática de estrutura intrínseca às RVMs em conjunto com a incapacidade de redução da dimensionalidade do espaço de entrada verificada pelo CHAOS-RVM, este conjunto de testes foi aplicado somente ao CHAOS-BMLP.

Tabela 6.18 – Erro absoluto (°C) das previsões de temperatura utilizadas

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
Média	3.12	4.78	0.63	0.88	1.12	1.36	1.58	1.78
Desvio	2.10	4.05	0.61	0.92	1.19	1.42	1.61	1.77
Máximo	8.28	28.00	4.38	7.35	9.46	10.93	11.94	12.66
Mínimo	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabela 6.19 – Desempenho do CHAOS-BMLP (MAPE) considerando fixa a estrutura e selecionando as variáveis de entrada

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
CHAOS-BMLP	4.54	3.84	0.72	1.18	1.45	1.61	1.80	2.02
Benchmark	4.73	1.98	0.56	0.83	1.00	1.15	1.20	1.30
Ganho (%)	4.08	-93.81	-28.36	-42.75	-44.65	-40.36	-50.36	-55.29

Comparando com a Tabela 6.11 e com a Tabela 6.12, os resultados apresentados nas Tabela 6.19 a Tabela 6.22 mostram a importância da seleção conjunta tanto do espaço de entrada quanto da estrutura, visto que na maioria dos casos o desempenho foi reduzido em virtude da fixação do conjunto de entradas ou da estrutura. Por outro lado, a queda de desempenho não foi tão acentuada, evidenciando tanto a capacidade de ponderação das entradas menos relevantes quando fixado o conjunto de entradas quanto o controle de complexidade da estrutura fixada e eventualmente sobredimensionada. Estas duas questões contribuem sobremaneira para o desempenho superior obtido pelo CHAOS-BMLP, que apesar de não obter resultados superiores aos melhores encontrados na literatura, traz consigo um elevado grau de automatização do processo de modelagem em conjunto com resultados satisfatórios.

Tabela 6.20 – Desempenho do CHAOS-BMLP (MAE e MAE%) considerando fixa a estrutura e selecionando as variáveis de entrada

	Caso 1	Caso 2	Caso 3 (%)					
	(%)	(MW)	1 passos	2 passos	3 passos	4 passos	5 passos	6 passos
CHAOS-BMLP	40.023	83.72	3.77	5.07	5.70	6.53	8.20	8.27
Benchmark	-	51.42	3.24	3.43	4.11	3.87	5.57	5.20
Ganho (%)	-	-62.83	-16.37	-47.69	-38.68	-68.74	-47.25	-59.09

Tabela 6.21 – Desempenho do CHAOS-BMLP (MAPE) fixando as entradas e selecionando a melhor estrutura

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
CHAOS-BMLP	5,09	3,86	0,68	1,16	1,49	1,69	1,88	2,02
Benchmark	4,73	1,98	0,56	0,83	1,00	1,15	1,20	1,30
Ganho (%)	-7,57	-94,74	-21,98	-40,13	-48,83	-46,60	-57,07	-55,29

Tabela 6.22 – Desempenho do CHAOS-BMLP (MAE e MAE%) fixando as entradas e selecionando a melhor estrutura

	Caso 1	Caso 2	Caso 3 (%)					
	(%)	(MW)	1 passos	2 passos	3 passos	4 passos	5 passos	6 passos
CHAOS-BMLP	41,29	83,11	3,44	4,42	5,56	6,92	8,25	8,27
Benchmark	-	51,42	3,24	3,43	4,11	3,87	5,57	5,20
Ganho (%)	-	-61,64	-6,32	-28,97	-35,25	-78,81	-48,04	-59,09

7 Conclusão e Trabalhos Futuros

Esta tese teve por objetivo o desenvolvimento de modelos neurais autônomos para previsão de carga. Autonomia aqui deve ser entendida como procedimentos automáticos para escolha do espaço de entrada e controle de complexidade da estrutura, incluindo seleção do modelo. Tomando por base dois modelos *feedforward* comumente utilizados em previsão de carga, para cada um foi desenvolvido um método para automatização do processo de especificação e treinamento, a saber: inferência *bayesiana* aplicada ao desenvolvimento de MLPs e minimização de limites superiores do erro de generalização para seleção de SVMs. Além destes modelos popularmente conhecidos, as máquinas de vetores relevantes (RVMs) foram aplicadas de forma pioneira ao problema de previsão de carga, sendo também desenvolvida uma metodologia automática para especificação dos hiperparâmetros do *kernel* e seleção de entradas.

Com foco na busca por metodologias automáticas para previsão de carga por meio de modelos neurais, os estudos preliminares indicaram a necessidade de sistemáticas para definição do conjunto inicial de entradas. Assim, tendo por base a teoria do caos e o conceito de sincronismo generalizado entre sistemas caóticos, foi desenvolvido um método para definição automática deste conjunto inicial. Além disso, diante da característica comparativa dos métodos encapsulados desenvolvidos para seleção de entradas, uma metodologia simples baseada na inserção de variáveis de prova foi desenvolvida e aplicada, estimando assim os referenciais de irrelevância requeridos pelos métodos.

Os resultados apresentados no capítulo 6 confirmam a necessidade de técnicas analíticas adequadas para seleção de entradas e controle de complexidade de modelos neurais de previsão de carga. Os métodos comumente encontrados na literatura, como

seleção linear de entradas e parada antecipada do treinamento, mostraram resultados decepcionantes, comprovando a inadequabilidade da primeira para modelos não-lineares e o caráter heurístico da segunda. Por outro lado, as técnicas *bayesianas* mostraram a sua aplicabilidade e eficiência em termos de capacidade de generalização no desenvolvimento de MLPs para previsão de carga. Fazendo uso de todos os níveis hierárquicos de inferência, desde a estimação dos parâmetros do modelo até a escolha da estrutura mais provável à luz dos dados, passando pela estimativa dos hiperparâmetros, este método permite selecionar, de forma automática e acoplada, tanto as entradas mais relevantes para o cálculo da saída, quanto o próprio modelo.

A técnica automática de especificação de SVMs, baseada na minimização de limites superiores do erro de generalização, mostrou resultados inferiores aos obtidos pela inferência *bayesiana*. Ao contrário da última, desenvolvida em 1992 por *David J. Mackay* [54], a primeira é relativamente nova, sendo proposta originalmente em 2005 [183] para escolha dos parâmetros C e ε, considerando fixo o *kernel* e os parâmetros que o definem. Em 2007, esta mesma técnica foi proposta por [184] para otimização dos parâmetros do *kernel*, considerando C e ε constantes. Desta forma, a busca pelas constantes que definem a SVM ainda é uma área de pesquisa em aberto, com a otimização conjunta destes parâmetros para problemas de regressão constituindo um problema ainda não abordado. O desacoplamento entre os processos, isolando a busca pelos parâmetros do *kernel* da otimização de C e ε, apesar de promissor, ainda necessita de alguns ajustes, visando acelerar a taxa de convergência e incrementar a qualidade da busca, ou seja, dar origem a modelos com melhor capacidade de generalização. Vale destacar que este método permite a utilização de *kernels* com número arbitrário de parâmetros, os quais podem ser otimizados através de um procedimento direcionado. Validação cruzada para busca em espaços de

dimensionalidade considerável é proibitiva, visto que a explosão combinatorial de possibilidades inviabiliza o processo em tempo prático.

Popular para problemas de classificação, em problemas de regressão este paradigma que funde treinamento e especificação da estrutura ainda vem sendo desenvolvido, tendo vencido a competição promovida pelo EUNITE em 2001 [29]. Entretanto, a busca por valores ótimos para os parâmetros que definem a SVM ainda constitui o principal empecilho na aplicação destes modelos. Estas restrições motivaram a inclusão das RVMs no trabalho. Em relação às SVMs, estes modelos necessitam somente da especificação dos hiperparâmetros do *kernel*, com os demais hiperparâmetros sendo estimados por meio da maximização da evidência de forma similar à aplicada por *David J. Mackay* [54] em MLPs. Além disso, os *kernels* utilizados não necessitam atender às condições de *Mercer*, ampliando o leque de funções que podem ser aplicadas. Por fim, as RVMs produzem representações mais esparsas do que as SVMs, fato confirmado pelos resultados apresentados no capítulo 6.

Além do pioneirismo no uso de RVMs em previsão de carga, foi desenvolvido um método encapsulado para seleção de entradas de RVMs inspirado em [63]. Conforme relatado nesta referência, a forma na qual a atualização dos múlitplos hiperparâmetros do *kernel* é combinada com a estimação dos demais parâmetros e hiperparâmetros do modelo ainda é um problema em aberto. A utilização de um ciclo único para atualização de todos os parâmetros e hiperparâmetros da forma proposta nesta tese não permitiu a estimação efetiva dos ponderadores das entradas (hiperparâmetros do *kernel*), limitando assim a capacidade de redução da dimensionalidade do espaço de entrada da técnica proposta. Apesar desta restrição, as RVMs apresentaram resultados superiores em relação às SVMs. Em conjunto com as vantagens teóricas acerca da maior esparsidade na modelagem e no menor número de

hipeparâmetros a serem especificados *a priori* pelo usuário, os resultados obtidos pelas RVMs indicam um novo caminho a seguir no uso de modelos baseados em *kernel* para previsão de carga.

Apesar dos resultados ainda inferiores verificados de uma forma geral para os modelos baseados em *kernel*, a inferência *bayesiana* aplicada ao desenvolvimento de MLPs (BMLP) apresentou desempenho de previsão satisfatório. De posse de conjuntos iniciais de entrada definidos por especialistas, esta metodologia superou os resultados encontrados na literatura para os três casos estudados, a menos para o caso 1 onde a técnica proposta apresentou resultado residualmente inferior. Este desempenho destacado, em conjunto com os resultados desabonadores obtidos por metodologias comumente encontradas na literatura, evidencia a necessidade de seleção adequada do espaço de entrada e controle de complexidade da estrutura estimada visando obter modelos com considerável capacidade de generalização.

Na busca por metodologias efetivamente autônomas, era necessário incorporar às técnicas de modelagem desenvolvidas sistemáticas para definição automática do conjunto inicial de entradas. Além de inadequadas a modelos não-lineares, técnicas para identificação linear de sistemas com base nas funções de autocorrelação e autocorrelação parcial dependem de heurísticas para detecção dos respectivos decaimentos das funções. Da mesma forma, não existem na literatura metodologias analíticas e automáticas para identificação de relacionamentos lineares entre séries temporais com base em funções de correlação cruzada. Assim, com base no conceito de sincronismo generalizado entre sistemas caóticos foi desenvolvido um método automático para definição do conjunto inicial de entradas.

Antes da avaliação do sincronismo entre sistemas é necessária a reconstrução dos respectivos espaços de estados, realizada utilizando o teorema de Takens em

conjunto com metodologias para especificação dos parâmetros de imersão, ou seja, atraso τ e dimensão d. Para automatização do processo de estimação destes parâmetros de imersão, o uso de algumas heurísticas foi necessário, passando pela escolha do método utilizado para cálculo da informação mútua $I_X(k)$ até o nível de significância α do teste de hipótese utilizado para definição do ponto de saturação da estatística $\delta(d)$ para obtenção da dimensão d. O uso de suposições simplificadoras como a hipótese de distribuição *gaussiana* para obtenção da expressão para o intervalo de discretização h_{hist} dos histogramas utilizados no cálculo de $I_X(k)$ (vide APÊNDICE A) mostra que o desenvolvimento de modelos totalmente autônomos e não-paramétricos ainda é um ideal distante.

Apesar das heurísticas utilizadas, em termos de aplicações práticas o nível de parametrização das metodologias propostas pode ser considerado satisfatório. Os poucos parâmetros definidos heuristicamente estão relacionados com funções secundárias dentro dos métodos como um todo, não apresentando relação direta com o desempenho dos modelos. Em outras palavras, o uso de um nível de significância α maior que o utilizado nesta tese pode resultar em um aumento da dimensão de imersão d estimada. Visto que as metodologias desenvolvidas incluem procedimentos encapsulados de seleção de entradas, eventuais sobreajustes deste parâmetro prejudiciais ao desempenho do modelo podem ser detectados *a posteriori*. Da mesma forma, o uso de um valor elevado para a dimensão máxima $d_{\max}$ implica somente em aumento do esforço computacional necessário para estimação de d. Esta mesma consideração acerca do tempo de processamento cabe para a definição dos limites de variação $[N_{\min}, N_{\max}]$ para o número de neurônios pesquisados no caso do BMLP.

Apesar de interessante sob o ponto de vista teórico visando à obtenção de um procedimento automático para identificação não-linear de sistemas, a metodologia para definição do conjunto inicial baseada exclusivamente na teoria do caos não apresentou resultados satisfatórios em virtude da modelagem inadequada das componentes sazonais presentes nas séries de carga. Conforme mencionado anteriormente, este fato pode indicar a inexistência de influência sazonal determinística sobre a dinâmica da carga, visto que esta não foi capturada na reconstrução do espaço de estados via teorema de *Takens*. A melhoria nos resultados promovida pela representação direta das sazonalidades por meio de codificação binária confirmou esta questão. Vale ressaltar que especificamente para os casos 1 e 3, a segmentação das bases de dados através do desenvolvimento de um modelo para cada dia da semana treinado com dados referentes a períodos específicos do ano contribuiu para modelagem das componentes sazonais semanal e anual.

Estas questões, referentes à identificação automática das sazonalidades presentes em conjunto com a sua correta modelagem, constituem duas vertentes de pesquisa importantes a serem seguidas. Análise espectral é uma ferramenta importante para a tarefa de identificação dos períodos de sazonalidade existentes, sendo necessários métodos para automatização do processo. De outra forma, ao dispor de métodos para identificação das funções de base a serem inseridas no modelo a cada passo do algoritmo iterativo, o estudo aprofundado das RVMs pode contribuir para definição de uma metodologia automática para identificação de períodos sazonais. Através da definição de diferentes *kernels* periódicos com hiperparâmetros ω_i relacionados com o inverso dos respectivos períodos, a estimação de ω_i por meio dos métodos apresentados na seção 5.2 pode contribuir para a modelagem automática das sazonalidades. Neste sentido, é necessário aprofundar o entendimento da iteração entre a estimação dos

hiperparâmetros do *kernel* e os demais parâmetros e hiperparâmetros do modelo, sendo também importante o estudo funcional visando o desenvolvimento de um *kernel* periódico com as características listadas acima.

Além da identificação e modelagem das sazonalidades, a segmentação da base de dados apresenta importância vital no desenvolvimento de modelos de previsão. Metodologias para partição automática do conjunto de treinamento visando obter modelos locais de previsão são necessárias no desenvolvimento de modelos neurais autônomos. Além disso, conforme mencionado anteriormente a segmentação da base de dados é uma forma alternativa para abordagem de comportamentos sazonais. Na seção 3.1.4 foram mencionadas algumas técnicas existentes para este fim, as quais necessitam da especificação de parâmetros como número de agrupamentos ou número de vizinhos a serem considerados. Desta forma, o desenvolvimento de métodos analíticos para especificação destes parâmetros é uma importante linha de pesquisa na busca por modelos neurais autônomos.

Especificamente sobre as metodologias para especificação automática e treinamento de redes neurais, a inferência *bayesiana* aplicada ao desenvolvimento de MLPs (BMLP) apresentou destaque em termos de desempenho de previsão dentre as metodologias testadas. Em conjunto com o método de definição do conjunto inicial baseado na teoria do caos e na representação binária das sazonalidades, este modelo surge como opção mais promissora no desenvolvimento de modelos autônomos de previsão de carga. Apesar das premissas simplificadoras consideradas na abordagem baseada na maximização da evidência, as estimativas analíticas para os parâmetros e hiperparâmetros do modelo apresentaram robustez no que tange ao desempenho do modelo. O uso de distribuições não-informativas, requerendo o uso de métodos de integração numérica baseados em simulações de Monte Carlo e modelos híbridos de

Markov [173], [174], ao abdicar de hipóteses restritivas sobre as distribuições envolvidas pode contibuir para a melhoria do desempenho de previsão, merecendo atenção em trabalhos futuros. Todavia, conforme destacado na seção 4.1, ao não fornecer estimativas pontuais para os hiperparâmetros esta abordagem inviabiliza o método de seleção de entradas proposto, devendo ser desenvolvido uma metodologia encapsulada específica para esta abordagem.

Ainda sobre o BMLP, a baixa redução na dimensionalidade do espaço de entrada para o modelo final pode ser explicada pelo tipo de distribuição utilizada para obtenção dos sinais de prova. Gerados a partir de distribuições uniformes, estes sinais produzem hiperparâmetros de referência extremamente conservadores. Com inspiração no nível de significância de testes de hipótese, o uso de distribuições mais informativas, porém com a geração de sinais ainda descorrelacionados com a saída, pode contribuir para o aumento da eficiência do método em termos de redução da dimensionalidade do espaço de entrada.

Sobre as máquinas baseadas em *kernel*, o limite superior do erro de generalização estimado por meio de validação única mostrou ser uma medida de difícil minimização, em virtude principalmente da sua característica multimodal. Este fato foi destacado pela sensibilidade do método ao passo de atualização da descida em gradiente, produzindo estimativas insatisfatórias para os hiperparâmetros em termos de desempenho de previsão. Além das dificuldades no processo de otimização, os resultados apresentados no capítulo 6 indicam que a avaliação dos modelos tomando por base os respectivos limites superiores do erro de generalização nem sempre conduzirá à escolha daquele com melhor capacidade de generalização. De maneira informal, é esperado que modelos mais complexos apresentem limite superior maior do que o estimado para modelos mais simples, em virtude do maior grau de flexibilidade

disponibilizado. Todavia, se o histórico em análise apresentar elevada variabilidade, o erro de generalização efetivamente verificado para o modelo mais complexo pode ser eventualmente menor do que o obtido para o modelo mais simples. Esta questão motiva a pesquisa de índices adequados para estimação da capacidade de generalização de SVMs, os quais possam ser usados em algoritmos para otimização dos diversos hiperparâmetros que definem estes modelos.

O reduzido número de hiperparâmetros a serem especificados pelo usuário constitui uma das principais vantagens das máquinas de vetores relevantes (RVMs) em relação às SVMs. Em conjunto com o método encapsulado de seleção de entradas desenvolvido a partir da otimização dos hiperparâmetros do *kernel*, a metodologia autônoma desenvolvida para RVMs apresentou resultados superiores em relação às SVMs. O método construtivo apresentado na seção 5.2 é especialmente interessante no sentido de permitir a utilização de diversas funções de base, as quais são selecionadas em virtude do impacto da sua inserção no modelo. Além da abordagem automática de sazonalidades, este critério analítico para escolha de funções de base pode ser também aplicado para utilização de *kernels* lineares, incluindo na modelagem eventuais relacionamentos lineares entre o espaço de entrada e a saída.

As questões levantadas neste capítulo indicam a existência de diversas linhas de pesquisa a serem seguidas na busca por modelos neurais autônomos para previsão de carga. Conforme mencionado anteriormente, modelagem totalmente autônoma e não-paramétrica é ainda um objetivo distante. Todavia, a partir de pressupostos e parametrizações restritos a níveis secundários do processo de modelagem, esta tese apresentou o desenvolvimento de três metodologias para especificação e treinamento automático de modelos neurais. Dentre os métodos propostos, a inferência bayesiana aplicada ao desenvolvimento de MLPs (BMLP) mostrou os resultados mais

promissores. Em conjunto com a técnica para definição do conjunto inicial de entradas baseada na teoria do caos e na representação binária das sazonalidades envolvidas, o BMLP surge como alternativa promissora na busca por métodos automáticos de identificação não-linear de sistemas. Tais métodos são importantes tendo em mente aplicações práticas como previsão de carga por barramento em sistemas de grande porte, possuindo número de barras da ordem de dezenas inviabilizando o estudo individualizado de cada histórico. Métodos automáticos como os desenvolvidos neste trabalho podem ser aplicados em setores diversos, como otimização de portfólios de ações, por exemplo. Neste problema, devem ser modelados diversos históricos de ações, tendo por objetivo a determinação da quantidade ótima de cada ação a ser adquirida em um dado período. Analogamente ao problema de previsão por barramento, o número elevado de séries inviabiliza o estudo individualizado de cada uma por parte de especialistas, requisitando o uso de métodos automáticos de identificação de sistemas.

Além das aplicações práticas mencionadas acima, as competições promovidas recentemente por diversas entidades ao redor do mundo objetivando o estudo e a avaliação de modelos automáticos de previsão evidencia a relevância do assunto abordado nesta tese. A distância existente entre as metodologias propostas e o ideal de modelagem puramente autônomo será diminuída com o advento de métodos para identificação e modelagem de sazonalidades, além de técnicas de segmentação automática da base de dados visando à obtenção de modelos locais de previsão. Estas duas questões constituem as principais linhas de pesquisa a serem seguidas na busca por modelos autônomos para previsão de carga.

8 Referências Bibliográficas

[1] HOBBS, B.F.; JITPRAPAIKULSARN, S.; MARATUKULAM, D.J.; KONDA, S.; CHANKONG, V.; LOPARO, K.A.; "Analysis of the Value for Unit Commitment of Improved Load Forecasts", *IEEE Transactions on Power Systems*, v.14, n.4, pp. 1342-1348, Nov. 1999.

[2] RANAWEERA, D.K.; KARADY, G.G.; FARMER, R.G.; "Economic Impact Analysis of Load Forecasting", *IEEE Transactions on Power Systems*, v.13, n.3, pp. 1388-1392, Aug. 1997.

[3] DOUGLAS, A.P.; BREIPHOL, A.M.; LEE, F.N.; ADAPA, R.; "Risk Due to Load Forecast Uncertainty in Short Term Power Systems Planning", *IEEE Transactions on Power Systems*, v.13, n.4, pp. 1493-1499, Nov. 1998.

[4] VALENZUELA, J.; MAZUMDAR, M.; KAPOOR, A.; "Influence of Temperature and Load Forecast Uncertainty on Estimates of Power Generation Costs", *IEEE Transactions on Power Systems*, v.15, n.2, pp. 668-674, May 2000.

[5] MOGHRAM, I.; RAHMAN, S.; "Analysis and Evaluation of Five Short-term Load Forecasting Techniques", *IEEE Transactions on Power Systems*, v.4, n.4, pp. 1484-1491, Oct. 1989.

[6] PAPALEXOPOULOS, A.D.; HESTERBERG, T.C.; "A Regression-based Approach to Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.5, n.4, pp. 1535-1550, Nov. 1990.

[7] HYDE, O.; HODNETT, P.F.; "An Adaptable Automated Procedure for Short-Term Electricity Load Forecasting", *IEEE Transactions on Power Systems*, v.12, n.1, pp. 84-94, Feb. 1997.

[8] RAMANATHAN, R.; ENGLE, R.; GRANGER, C.W.J.; ARAGHI, F.V., BRACE, C.; "Short-Run Forecasts of Electricity Loads and Peaks", *International Journal of Forecasting*, v.13, n.2, pp. 161-174, June 1997.

[9] MBAMALU, G.A.N.; EL-HAWARY, M.E.; "Load Forecasting via Suboptimal Seasonal Autoregressive Models and Iteratively Reweighted Least Squares Estimation", *IEEE Transactions on Power Systems*, v.8, n.1, pp. 343-348, Feb. 1993.

[10] FAN, J.Y.; MCDONALD, J.D.; "A Real Time Implementation of Short-term Load Forecasting for Distribution Power Systems", *IEEE Transactions on Power Systems*, v.9, n.2, pp. 988-994, May 1994.

[11] DARBELLAY, G.A.; SLAMA, M.; "Forecasting the Short-Term Demand for Electricity: Do Neural Networks Stand a Better Chance?", *International Journal of Forecasting*, v.16, n.1, pp. 71-83, Jan. 2000.

[12] ZAGRAJEK, J.N.; WERON, R.; "Modeling Electricity Loads in California: ARMA Models with Hyperbolic Noise", *Signal Processing*, v.82, n.12, pp. 1903-1915, Dec. 2002.

[13] HUANG, S.J.; SHIH, K.R.; "Short-Term Load Forecasting Via ARMA Model Identification Including Non-Gaussian Process Considerations", *IEEE Transactions on Power Systems*, v.18, n.2, pp. 673-679, May 2003.

[14] PARK, D.C; EL-SHARKAWI, M.A.; MARKS II, R.J.; "An Adaptively Trained Neural Network", *IEEE Transactions on Neural Networks*, v.2, n.3, pp. 334-345, May 1991.

[15] PAPALEXOPOULOS, A.D.; HAO, S.; PENG, T.M.; "An Implementation of a Neural Network Based Load Forecasting Model for the EMS", *IEEE Transactions on Power Systems*, v.9, n.4, pp. 1956-1962, Nov. 1994.

[16] RANAWEERA, D.K.; HUBELE, N.F.; PAPALEXOPOULOS, A.D.; "Application of Radial Basis Function Neural Network Model for Short-term Load Forecasting", *IEE Proceedings on Generation, Transmission and Distribution*, v.142, n.1, pp. 45-50, Jan. 1995.

[17] MOHAMMED, O. PARK, D.; MERCHANT, R.; DINH, T.; TONG, C.; AZEEM, A.; FARAH, J.; DRAKE, C.; "Practical Experiences with an Adaptive Neural Network Short-term Load Forecasting System", *IEEE Transactions on Power Systems*, v.10, n.1, pp. 254-265, Feb. 1995.

[18] KHOTANZAD, A.; AFKHAMI-ROHANI, R.; LU, T.L.; ABAYE, A.; DAVIS, M.; MARATUKULAM, D.J.; "ANNSTLF – A Neural-Network-Based Electric Load Forecasting System", *IEEE Transactions on Neural Networks,* v.8, n.4, pp. 835-846, July 1997.

[19] DREZGA, I.; RAHMAN, S.; "Input Variable Selection for Ann-Based Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.13, n.4, pp. 1238-1244, Nov. 1998.

[20] KHOTANZAD, A.; AFKHAMI-ROHANI, R.; MARATUKULAM, D.; "ANNSTLF – Artificial Neural Network Short-Term Load Forecaster – Generation Three", *IEEE Transactions on Power Systems*, v.13, n.4, pp. 1413-1422, Nov. 1998.

[21] DREZGA, I.; RAHMAN, S.; "Short-term Load Forecasting with Local ANN Predictors", *IEEE Transactions on Power Systems*, v.14, n.3, pp. 844-850, Aug. 1999.

[22] ALVES DA SILVA, A.P; MOULIN, L.; "Confidence Intervals for Neural Network Based Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.15, n.4, pp. 1191-1196, Nov. 2000.

[23] ALVES DA SILVA, A.P.; RODRIGUES, U.P.; REIS, A.J.R.; MOULIN, L.S.; "Oráculo – Uma ferramenta para Previsão de Carga", *XVI SNPTEE – Seminário Nacional de Produção e Transmissão de Energia Elétrica*, GOP/012, Campinas, São Paulo, Brasil, 21-26 de Outubro de 2001.

[24] OSOWSKI, S.; SIWEK, K.; "Regularization of Neural Networks for Improved Load Forecasting in the Power System", *IEE Proceedings on Generation, Transmission and Distribution*, v.149, n.3, pp. 340-344, May 2002.

[25] TAYLOR, J.W.; BUIZZA, R.; "Neural Network Load Forecasting with Weather Ensemble Predictions", *IEEE Transactions on Power Systems*, v.17, n.3, pp. 626-632, Aug. 2002.

[26] SAINI, L.M.; SONI, M.K.; "Artificial Neural Network-Based Peak Load Forecasting Using Conjugate Gradient Methods", *IEEE Transactions on Power Systems*, v.17, n.3, pp. 907-912, Aug. 2002.

[27] CARPINTEIRO, O.A.S.; REIS, A.J.R.; ALVES DA SILVA, A.P.; "A Hierarchical Neural Model in Short-Term Load Forecasting", *Applied Soft Computing*, v.4, n.4, pp. 405-412, Sept. 2004.

[28] SATISH, B.; SWARUP, K.S.; SRINIVAS, S.; RAO, A.H.; "Effect of Temperature on Short Term Load Forecasting Using a Integrated ANN", *Electric Power Systems Research*, v.72, n.1, pp. 95-101, Nov. 2004.

[29] CHEN, B.-J.; CHANG, M.-W.; LIN, C.-J; "Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001", *IEEE Transactions on Power Systems*, v.19, n.4, pp. 1821-1830, Nov. 2004.

[30] REIS, A.J.R.; ALVES DA SILVA, A.P.. "Feature Extraction Via Multi-Resolution Analysis for Short-Term Load Forecasting", *IEEE Transactions on Power Systems,* v.20, n.1, pp. 189-198, Feb. 2005.

[31] MANDAL, P.; SENJYU, T.; UEZATO, K.; FUNABASHI, T.; "Several-Hours-Ahead Electricity Price and Load Forecasting Using Neural Networks", In: *IEEE PES General Meeting*, San Francisco, USA, June 2005.

[32] FERREIRA, V.H.; *Técnicas de Regularização de Modelos Neurais Aplicadas à Previsão de Carga a Curto Prazo*, Tese de Ms.C. COPPE/UFRJ, Rio de Janeiro, Rj, Brasil, 2005.

[33] FERREIRA, V.H.; ALVES DA SILVA, A.P., "Complexity Control of Neural Models for Load Forecasting", In: *Proc. International Conference on Intelligent System Application to Power Systems*, pp. 100-104, Washington D.C., USA, Nov. 2005.

[34] FERREIRA, V.H.; ALVES DA SILVA, A.P., "Toward Estimating Autonomous Neural Network-Based Electric Load Forecasters", *IEEE Transactions on Power Systems,* v.22, n.4, pp. 1554-1562, Nov. 2007.

[35] MORI, H.; KOBAYASHI, H.; "Optimal Fuzzy Inference for Short-term Load Forecasting", *IEEE Transactions on Power Systems,* v.11, n.1, pp. 390-396, Feb. 1996.

[36] SENJYU, T.; HIGA, S.; UEZATO, K.; "Future Load Curve Shaping Based on Similarity Using Fuzzy Logic Approach", *IEE Proceedings on Generation, Transmission and Distribution*, v.145, n.4, pp. 375-380, July 1998.

[37] MASTOROCOSTAS, P.A.; THEOCHARIS, J.B.; BAKIRTZIS, A.G.; "Fuzzy Modeling for Short Term Load Forecasting Using the Orthogonal Least Squares Method", *IEEE Transactions on Power Systems*, v.14, n.1, pp. 29-36, Feb. 1999.

[38] BAKIRTZIS, A.G.; THEOCHARIS, J.B.; KIARTZIS, S.J.; SATSIOS, K.J.; "Short-term Load Forecasting Using Fuzzy Neural Networks", *IEEE Transactions on Power Systems*, v.10, n.3, pp. 1518-1524, Aug. 1995.

[39] KIM, K.H.; PARK, J.K.; HWANG, K.J.; KIM, S.H.; "Implementation of Hybrid Short-term Load Forecasting System Using Artificial Neural Networks and Fuzzy Expert Systems", *IEEE Transactions on Power Systems,* v.10, n.3, pp. 1534-1539, Aug. 1995.

[40] YANG, H.T.; HUANG, C.M.; HUANG, C.L.; "Identification of ARMAX Model for Short Term Load Forecasting: An Evolutionary Programming Approach", *IEEE Transactions on Power Systems*, v.11, n.1, pp. 403-408, Feb. 1996.

[41] YANG, H.T.; HUANG, C.M.; "A New Short-term Load Forecasting Approach Using Self-Organizing Fuzzy ARMAX Models", *IEEE Transactions on Power Systems*, v.13, n.1, pp. 217-225, Feb. 1998.

[42] KASSAEI, H.R.; KEYHANI, A.; WOUNG, T.; RAHMAN, M.; "A Hybrid Fuzzy, Neural Network Bus Load Modeling and Predication", *IEEE Transactions on Power Systems*, v.14, n.2, pp. 718-724, May 1999.

[43] SRINIVASAN, D.; TAN, S.S.; CHANG, C.S.; CHAN, E.K.; "Parallel Neural Network-Fuzzy Expert System Strategy for Short-Term Load Forecasting: System Implementation and Performance Evaluation", *IEEE Transactions on Power Systems*, v.14, n.3, pp. 1100-1106, Aug. 1999.

[44] TAMIMI, M.; EGBERT, R.; "Short Term Electric Load Forecasting via Fuzzy Neural Collaboration", *Electric Power Systems Research*, v.56, n.3, pp. 243-248, Dec. 2000.

[45] KHOTANZAD, A.; ZHOU, E.; ELRAGAL, H.; "A Neuro-Fuzzy Approach to Short-Term Load Forecasting in a Price-Sensitive Environment", *IEEE Transactions on Power Systems*, v. 17, n.4, pp. 1273-1282, Nov. 2002.

[46] VILLALBA, S.A.; BEL, C.A.; "Hybrid Demand Model for Load Estimation and Short Term Load Forecasting in Distribution Electric Systems", *IEEE Transactions on Power Delivery*, v.15, n.2, pp. 764-769, Apr. 2000.

[47] FIDALGO, J.N.; PEÇAS LOPES, J.A.; "Load Forecasting Performance Enhancement When Facing Anomalous Events", *IEEE Transactions On Power Systems*, v.20, n.1, pp. 408-415, Feb. 2005.

[48] AMJADY, N., "Short-Term Bus Load Forecasting of Power Systems by a New Hybrid Method", *IEEE Transactions on Power Systems*, v.22, n.1, pp. 333-341, Feb. 2007.

[49] ONS, OPERADOR NACIONAL DO SISTEMA ELÉTRICO, *Procedimentos de Rede - Módulo 5: Consolidação da Previsão de Carga*, 2002.

[50] BISHOP, C.M.; *Neural Networks for Pattern Recognition*, Oxford, New York, Oxford University Press, 1995.

[51] ALVES DA SILVA, A.P.; QUINTANA, V.H.; PANG, G.K.H. Pang; "Neural Networks for Topology Determination of Power Systems", In: *Proc. First International Forum on Applications of Neural Networks to Power Systems*, pp. 297-301, Seattle, USA, Jul. 1991.

[52] MACKAY, D.J.C.; "Bayesian Non-linear Modelling for the Prediction Competition", *American Society of Heating, Refrigeration and Air-Conditioning Engineers Symposium*, Denver, 1993. Disponível em <www.inference.phy.cam.ac.uk/mackay/pred.pdf>. Acesso em 07/06/2008, 11:41:30.

[53] TITO, E.; ZAVERUCHA, G.; VELLASCO, M.; PACHECO, M.A.; "Applying Bayesian Neural Networks to Electric Load Forecasting", In: *Proceedings of Sixth IEEE International Conference on Neural Information Processing*, v.1, pp. 407-411, Perth, Australia, November 1999.

[54] MACKAY, D.J.C.; *Bayesian Methods for Adaptive Models*, Ph.D. dissertation, California Institute of Technology, Pasadena, California, USA, 1992.

[55] MATSUI, T.; IIZAKA, T.; FUKUYAMA, Y; "A Novel Daily Peak Load Forecasting Method Using Analyzable Structured Neural Network", In: *IEEE PES Winter Meeting*, pp. 405-410, Columbus, USA, Jan. 2001.

[56] ALVES DA SILVA, A.P.; "Overcoming Limitations of NNs for On-Line DSA", In: *IEEE PES General Meeting*, San Francisco, USA, June 2005.

[57] AMARI, S.; MURATA, N.; MÜLLER, K.R.; FINKE, M.; YANG, H.; "Statistical Theory of Overtraining – Is Cross-Validation Asymptotically Effective?", *Advances in Neural Information Processing Systems*, v.8, MIT Press, pp. 176-182, 1996.

[58] CATATELPE, Z.; ABU-MOSTAFA, Y.S.; MAGDON-ISMAIL, M.; "No Free Lunch for Early Stopping", *Neural Computation*, v.11, n.4, pp. 995-1009, May 1999.

[59] KOHAVI, R.; JOHN, G.; "Wrappers for Feature Selection", *Artificial Intelligence*, v.97, n.1-2, pp.273-324, Dec. 1997.

[60] GUYON, I.; ELISSEEFF, A.; "An Introduction to Variable and Feature Selection", *Journal of Machine Learning Research*, n.3, pp. 1157-1182, Mar. 2003.

[61] VAPNIK, V.N.; *Statistical Learning Theory*, New York, John Wiley & Sons, 1998.

[62] SCHÖLKOPF, B.; SMOLA, A.J.; *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, Cambridge, Massachusetts, 2002.

[63] TIPPING, M.E.; "Sparse Bayesian Learning and the Relevance Vector Machine", *Journal of Machine Learning Research*, v.1, pp. 211-244, Sep. 2001.

[64] TIPPING, M.E.; FAUL, A.C.; "Fast Marginal Likelihood Maximisation for Sparse Bayesian Models", In: BISHOP, C.M.; FREY, B.J. (eds.); *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, FL, USA, Jan. 2003.

[65] TAKENS, F., "Detecting Strange Attractors in Turbulence", |In.: D.A. Rand, L.-S. Young (eds.), *Dynamical Systems and Turbulence, Lecture Notes in Mathematics*, v.898, pp. 366-381, Springer-Verlag, 1981.

[66] RULKOV, N.F.; SUSHCHIK, M.M.; TSIMRING, L.S.; ABARBANEL, H.D.I.; "Generalized Synchronization of Chaos in Directionally Coupled Chaotic Systems", *Physical Review E*, v.51, n.2, pp. 980-994, Feb. 1995.

[67] PYRAGAS, K.; "Weak and Strong Synchronization of Chaos", *Physical Review E*, v.54, n.5, pp. R4508-R4511, Nov. 1996.

[68] STOPPIGLIA, H., DREYFUS, G., DUBOIS, R., OUSSAR, Y.; "Ranking a Random Feature for Variable and Feature Selection", *Journal of Machine Learning Research*, n.3, pp. 1399-1414, Mar. 2003.

[69] MANDAL, P.; SENJYU, T.; URASAKI, N.; FUNABASHI, T.; "A neural network based several-hour-ahead electric load forecasting using similar days approach", *International Journal of Electrical Power & Energy Systems*, v.28, n.6, pp. 367-373, Jul. 2006.

[70] MANDAL, P.; SENJYU, T.; FUNABASHI, T.; "Neural Networks Approach to Forecast Several Hour Ahead Electricity Prices and Loads in Deregulated Market", *Energy Conversion and Management*, v.47, n.15-16, pp. 2128-2142, Sept. 2006.

[71] KODOGIANNIS, V.S.; ANAGNOSTAKIS, E.M.; "A Study of Advanced Learning Algorithms for Short-term Load Forecasting", *Engineering Applications of Artificial Intelligence*, v.12, n.2, pp. 159-173, April 1999.

[72] PADMAKUMARI K.; MOHANDAS K.P.; THIRUVENGADAM S.; "Long Term Distribution Demand Forecasting Using Neuro Fuzzy Computations", *International Journal of Electrical Power and Energy Systems*, v.21, n.5, pp. 315-322, Jun. 1999.

[73] YAO, S.J.; SONG, Y.H.; ZHANG, L.Z.; CHENG, X.Y.; "Wavelet Transform and Neural Networks for Short-Term Electrical Load Forecasting", *Energy Conversion and Management*, v.41, n.18, pp. 1975-1988, Dec. 2000.

[74] DJUKANOVIC, M.; BABIC, B.; SOBAJIC, D.J.; PAO, Y,H.; "Unsupervised/Supervised Learning Concept for 24-hour Load Forecasting", *IEE Proceedings C*, v.140, n.4, July 1993.

[75] DASH, P.K.; SATPATHY, H.P.; LIEW, A.C.; RAHMAN, S.; "A Real-Time Short-term Load Forecasting System Using Functional Link Network", *IEEE Transactions on Power Systems*, v.12, n.2, pp. 675-680, May 1997.

[76] YANG, J.; STENZEL, J., "Short-term Load Forecasting with Increment Regression Tree", *Electric Power Systems Research*, v.76, n.9-10, pp. 880–888, Jun. 2006.

[77] CHEN, B.J.; CHANG, M.W.; LIN, C.J.; "Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001", *IEEE Transactions on Power Systems*, v.19, n.4, pp. 1821-1830, Nov. 2004.

[78] DONG, B.; CAO, C.; LEE, S.E.; "Applying Support Vector Machines to Predict Building Energy Consumption in Tropical Region", *Energy and Buildings*, v.37, n.5, pp. 545–553, May 2005.

[79] PAI, P.-F., HONG, W.-C.; "Forecasting Regional Electricity Load Based on Recurrent Support Vector Machines with Genetic Algorithms", *Electric Power Systems Research*, v.74, n.3, pp. 417–425, Jun. 2005.

[80] PAI, P.-F., HONG, W.-C.; "Support Vector Machines with Simulated Annealing Algorithms in Electricity Load Forecasting", *Energy Conversion and Management*, v.46, n.17, pp. 2669–2688, Oct. 2005.

[81] FAN, S.; CHEN, L.; "Short-Term Load Forecasting Based on an Adaptive Hybrid Method" *IEEE Transactions on Power Systems*, v.21, n.1, pp. 392-401, Feb. 2006.

[82] VERMAAK, J.; BOTHA, E.C.; "Recurrent Neural Networks for Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.13, n.1, pp. 126-132, Feb. 1998.

[83] AL-SABA, T.; EL-AMIN, I.; "Artificial Neural Networks as Applied to Long-term Demand Forecasting", *Artificial Intelligence in Engineering*, v.13, n.2, pp. 189-197, April 1999.

[84] DING, A.A.; "Neural-Network Prediction with Noisy Predictors", *IEEE Transactions on Neural Networks*, v.10, n.5, pp. 1196-1203, Sept. 1999.

[85] KIM, K.H.; YOUN, H.S.; KANG, Y.C.; "Short-term Load Forecasting for Special Days in Anomalous Load Conditions Using Neural Networks and

Fuzzy Inference Method", *IEEE Transactions on Power Systems*, v.15, n.2, pp. 559-565, May 2000.

[86] MORI, H.; YUIHARA, A.; "Deterministic Annealing Clustering for ANN-Based Short-Term Load Forecasting", *IEEE Transactions on Power Systems*, v.16, n.3, pp. 545-551, Aug. 2001.

[87] SENJYU, T.; TAKARA, H.; UEZATO, K.; FUNABASHI, T.; "One-Hour-Ahead Load Forecasting Using Neural Network", *IEEE Transactions on Power Systems*, v.17, n.1, pp. 113-118, Feb. 2002.

[88] MARIN, F.J.; GARCIA-LAGOS, F.; JOYA, G.; SANDOVAL, F.; "Global Model for Short-Term Load Forecasting Using Artificial Neural Networks", *IEE Procedings on Generation, Transmission and Distribution*, v.149, n.2, pp. 121-125, Mar. 2002.

[89] KERMANSHAHI, B.; IWAMIYA, H.; "Up to Year 2020 Load Forecasting Using Neural Nets", *International Journal of Electrical Power & Energy Systems*, v.24, n.9, pp. 789-797, Nov. 2002.

[90] HSU, C.C.; CHEN, C.Y.; "Regional Load Forecasting in Taiwan – Applications of Artificial Neural Networks", *Energy Conversion and Management*, v.44, n.12, pp. 1941-1949, July 2003.

[91] BECCALI, M.; CELLURA, M., LO BRANO, V.; MARVUGLIA, A.; "Forecasting Daily Urban Electric Load Profiles Using Artificial Neural Networks", *Energy Conversion and Management*, v.45, n.18, pp. 2879-2900, Nov. 2004.

[92] YALCINOZ, T.; EMINOGLU, U., "Short Term and Medium Term Power Distribution Load Forecasting by Neural Networks", *Energy Conversion and Management*, v.46, n.9-10, pp. 1393-1405, Jun. 2005.

[93] CHATFIELD, C.; *The Analysis of Time Series: An Introduction*, 6th. edition, Chapman and Hall/CRC, 2004.

[94] PIRAS, A.; BUCHENEL, B.; JACCARD, Y.; GERMOND, A.; IMHOF, K.; "Heterogeneous Artificial Neural Network for Short-term Electrical Load Forecasting", *IEEE Transactions on Power Systems*, v.11, n.1, pp. 397-402, Feb. 1996.

[95] CHARYTONIUK, W.; CHEN, M.-S.; "Very Short-term Load Forecasting Using Artificial Neural Networks", *IEEE Transactions on Power Systems*, v.15, n.1, pp. 263-268, Feb. 2000.

[96] EL DESOUKY, A.A.; ELKATEB, M.M.; "Hybrid Adaptive Techniques for Electric-Load Forecast Using ANN and ARIMA", *IEE Proceedings on Generation, Transmission and Distribution*, v.147, n.4, pp. 213-217, July 2000.

[97] LIANG, R.H.; CHENG, C.C.; "Short-Term Load Forecasting by a Neuro-Fuzzy Based Approach", *International Journal of Electrical Power & Energy Systems*, v.24, n.2, pp. 103-111, Feb. 2002.

[98] FAY, D.; RINGWOOD, J.V.; CONDON, M.; KELLY, M.; "24-h Electrical Load Data – A Sequential or Partitioned Time Series?", *Neurocomputing*, v.55, pp. 469-498, Mar. 2003.

[99] TSEKOURAS, G.J.; HATZIARGYRIOU, N.D.; DIALYNAS, E.N.; "An optimized adaptive neural network for annual midterm energy forecasting" *IEEE Transactions on Power Systems*, v.21, n.1, pp. 385-391, Feb. 2006.

[100] KANDIL, N.; WAMKEUE, R.; SAAD, M.; GEORGES, S.; "An Efficient Approach for Short Term Load Forecasting Using Artificial Neural Networks", *International Journal of Electrical Power & Energy Systems*, v.28, n.8, pp. 525-530, Oct. 2006.

[101] SHANMUGAN, K.S.; BREIPOHL, A.M.; *Random Signals: Detection, Estimation and Data Analysis*, John Wiley & Sons, 1988.

[102] SAINI, L.M.; SONI, M.K.; "Artificial Neural Network Based Peak Load Forecasting Using Levenberg-Marquardt and Quasi-Newton Methods", *IEE Proceedings on Generation, Transmission and Distribution*, v.149, n.5, pp. 578-584, Sep. 2002.

[103] DOVEH, E.; FEIGIN, P.; GREIG, D.; HYAMS, L.; "Experience with FNN Models for Medium Term Power Demand Predictions", *IEEE Transactions on Power Systems,* v.14, n.2, pp. 538-546, May 1999.

[104] COVER, T.M.; THOMAS, J.A.; *Elements of Information Theory,* John Wiley & Sons, 1991.

[105] WESTON, J., MUKHERJEE, S., CHAPELLE, O., PONTIL, M., POGGIO, T., VAPNIK, V.; "Feature Selection for SVMs", *Advances in Neural Information Processing Systems*, v.13, 2000.

[106] ZHANG, B.L.; DONG, Z.Y.; "An Adaptive Neural-Wavelet Model for Short Term Load Forecasting", Electric Power Systems Research, v.59, n.2, pp. 121-129, Sept. 2001.

[107] PARK, D.C.; EL-SHARKAWI, M.A.; MARKS II, R.J.; ATLAS, L.E.; DAMBORG, M.J.; "Electric Load Forecasting Using An Artificial Neural Network", *IEEE Transactions on Power Systems*, v.6, n.2, pp. 442-449, May 1991.

[108] HSU, Y.Y.; YANG, C.C.; "Design of Artificial Neural Networks for Short-Term Load Forecasting. Part II: Multilayer Feedforward Networks for Peak Load and Valley Load Forecasting", *IEE Proceedings C*, v.138, n.5, pp. 414-418, Sept. 1991.

[109] LEE, K.Y.; CHA, Y.T.; PARK, J.H.; "Short-term Load Forecasting Using an Artificial Neural Network", *IEEE Transactions on Power Systems*, v.7, n.1, pp. 124-132, Feb. 1992.

[110] HO, K.L.; HSU, Y.Y.; YANG, C.C.; "Short-term Load Forecasting Using a Multilayer Neural Network with an Adaptive Learning Algorithm", *IEEE Transactions on Power Systems*, v.7, n.1, pp. 141-149, Feb. 1992.

[111] SRINIVASAN, D.; LIEW, A.C.; CHANG, C.S.; "Forecasting Daily Load Curves Using a Hybrid Fuzzy-Neural Approach", *IEE Proceedings on Generation, Transmission and Distribution*, v.141, n.6, pp. 561-567, Nov. 1994.

[112] GIRGIS, A.A.; VARADAN, S.; "Unit Commitment Using Load Forecasting Based on Artificial Neural Networks", *Electric Power Systems Research*, v.32, n.3, pp. 213-217, Mar. 1995.

[113] SRINIVASAN, D.; CHANG, C.S.; LIEW, A.C.; "Demand Forecasting Using Fuzzy Neural Computation, with Special Emphasis on Weekend And Public Holiday Forecasting", *IEEE Transactions on Power Systems*, v.10, n.4, pp. 1897-1903, Nov. 1995.

[114] LIU, K.; SUBBARAYAN, S.; SHOULTS, R.R.; MANRY, M.T.; KWAN, C.; LEWIS, F.I.; NACCARINO, J.; "Comparison of very short-term load forecasting techniques", *IEEE Transactions on Power Systems*, v.11, n.2, pp. 877-882, May 1996.

[115] CHOW, T.W.S.; LEUNG, C.T.; "Nonlinear Autoregressive Integrated Neural Network Model for Short-Term Load Forecasting", *IEE Proceedings on Generation, Transmission and Distribution*, v.143, n.5, pp. 500-506, Sept. 1996.

[116] CHOW, T.W.S.; LEUNG, C.T.; "Neural Network Based Short-Term Load Forecasting Using Weather Compensation", *IEEE Transactions on Power Systems*, v.11, n.4, pp. 1736-1742, Nov. 1996.

[117] LAMEDICA, R.; PRUDENZI, A.; SFORNA, M.; CACIOTTA, M; CENCELLI, V.O.; "A Neural Network Based Technique For Short-Term Load Forecasting of Anomalous Load Periods", *IEEE Transactions on Power Systems*, v.11, n.4, pp. 1749-1756, Nov. 1996.

[118] ALFUHAID, A.S.; EL-SAYED, M.A.; MAHMOUD, M.S.; "Cascaded Artificial Neural Networks for Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.12, n.4, pp. 1524-1529, Nov. 1997.

[119] KIARTZIS, S.J.; ZOUMAS, C.E.; THEOCHARIS, J.B.; BAKIRTZIS, A.G.; PETRIDIS, V.; "Short-term Load Forecasting in an Autonomous Power System Using Artificial Neural Networks", *IEEE Transactions on Power Systems*, v.12, n.4, pp. 1591-1596, Nov. 1997.

[120] SRINIVASAN, D.; TAN, S.S.; CHANG, C.S.; CHAN, E.K.; "Practical Implementation of a Hybrid Fuzzy Neural Network for One-day Ahead Load Forecasting", *IEE Proceedings on Generation, Transmission and Distribution*, v.145, n.6, pp. 687-692, Nov. 1998.

[121] DANESHDOOST, M.; LOTFALIAN, M.; BUMROONGGIT, G.; NGOY, J.P.; "Neural Network with Fuzzy Set-Based Classification for Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.13, n.4, pp. 1386-1391, Nov. 1998.

[122] ELKATEB, M.M.; SOLAIMAN, K.; AL-TURKI, Y.; "A Comparative Study of Medium-weather-dependent Load Forecasting Using Enhanced Artificial/Fuzzy

Neural Network and Statistical Techniques"; *Neurocomputing*, v.23, n.1, pp. 3-13, Dec.1998.

[123] KERMANSHAHI, B.; "Recurrent Neural Network for Forecasting Next 10 Years Loads of Nine Japanese Utilities" *Neurocomputing*, v.23, n.1-3, pp. 125-133, Dec. 1998.

[124] SRINIVASAN, D.; "Evolving Artificial Neural Networks for Short Term Load Forecasting", *Neurocomputing*, v.23, n.1, pp. 265-276, Dec. 1998.

[125] MURATA, N.; YOSHIZAWA, S.; AMARI, S.I.; "Network Information Criterion – Determining the Number of Hidden Units for an Artificial Neural Network", *IEEE Transactions on Neural Networks,* v.5, n.6, pp. 865-872, Nov. 1994.

[126] SWANSON, N.R.; WHITE, H.; "A Model-Selection Approach to Assessing the Information in the Term Structure Using Linear Models and Artificial Neural Networks", *Journal of Businees and Economic Statistics*, v.13, n.3, pp. 265-275, Jul. 1995.

[127] SWANSON, N.R.; WHITE, H.; "Forecasting Economic Time Series Using Flexible versus Fixed Specification and Linear versus Nonlinear Econometric Models", *International Journal of Forecasting*, v.13, n.4, pp. 439-461, Dec. 1997.

[128] SWANSON, N.R.; WHITE, H.; "A Model Selection Approach to Real-time Macroeconomic Forecasting Using Linear Models and Artificial Neural Networks", *Review of Economic and Statistics*, v.79, pp. 540-550, 1997.

[129] MEDEIROS, M.C.; VEIGA, A.; "A Flexible Coefficient Smooth Transition Time Series Model", *IEEE Transactions on Neural Networks,* v.16, n.1, pp. 97-113, Jan. 2005.

[130] ANDERS, U.; KORN, O.; "Model Selection on Neural Networks", *Neural Networks*, v.12, n.2, pp. 309-323, Mar. 1999.

[131] ATLAS, L.; COLE, R.; MUTHUSAMY, Y.; LIPPMAN, A.; CONNOR, J.; PARK, D.; EL-SHARKAWI, M.; MARKS, R.J.; "A Performance Comparison of Trained Multilayer Perceptrons and Trained Classification Trees", *Proceedings of IEEE*, v.78, n.10, pp. 1614-1619, Oct. 1990.

[132] PENG, T.M.; HUBELE, N.F.; KARADY, G.G.; "Advancement in the Application of Neural Networks for Short-term Load Forecasting", *IEEE Transactions on Power Systems,* v.7, n.1, pp. 250-257, Feb. 1992.

[133] CHEN, S.T; YU, D.C.; MOGHADDAMJO, A.R.; "Weather Sensitive Short-term Load Forecasting Using Nonfully Connected Artificial Neural Network", *IEEE Transactions on Power Systems*, v.7, n.3, pp. 1098-1105, Aug. 1992.

[134] LU, C.N.; WU, H.T.; VEMURI, S.; "Neural Network Based Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.8, n.1, pp.336-342, Feb. 1993.

[135] LAMEDICA, R.; PRUDENZI, A.; SFORNA, M.; CACIOTTA, M; CENCELLI, V.O.; "A Neural Network Based Technique For Short-Term Load Forecasting of Anomalous Load Periods", *IEEE Transactions on Power Systems*, v.11, n.4, pp. 1749-1756, Nov. 1996.

[136] RANAWEERA, D.K.; KARADY, G.G.; FARMER, R.G.; "Effect of Probabilistic Inputs on Neural Network-Based Electric Load Forecasting", *IEEE Transactions on Neural Networks*, v.7, n.6, pp. 1528-1532, Nov. 1996.

[137] CHIU, C.C.; KAO, L.J.; COOK, D.F.; "Combining a Neural Network with a Rule-Based Expert System Approach for Short-term Power Load Forecasting in Taiwan", *Expert Systems with Applications*, v.13, n.4, pp. 299-305, Nov. 1997.

[138] HIPPERT, H.S.; BUNN, D.W.; SOUZA, R.C.; "Large Neural Networks for Electricity Load Forecasting: Are They Overfitted?", *International Journal of Forecasting*, v.21, n.3, pp 425– 434, Jul. 2005.

[139] GHIASSI, M.; ZIMBRA, D.K.; SAIDANE, H.; "Medium term system load forecasting with a dynamic artificial neural network model", *Electric Power Systems Research*, v.76, n.5, pp 302–316, Mar. 2006.

[140] LIAO, G.-C.; TSAO, T.-P., "Application of a Fuzzy Neural Network Combined With a Chaos Genetic Algorithm and Simulated Annealing to Short-Term Load Forecasting", *IEEE Transactions On Evolutionary Computation*, v.10, n.3, pp. 330-340, Jun. 2006.

[141] CHAN, Z.S.H.; NGAN, H.W.; RAD, A.B., DAVID, A.K.; KASABOV, N.; "Short-term ANN Load Forecasting from Limited Data Using Generalization Learning Strategies", *Neurocomputing*, v.70, n.1-3, pp. 409–419, Dec. 2006.

[142] KWOK, T.Y.; YENUG, D.Y.; "Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems", *IEEE Transactions on Neural Networks*, v.8, pp. 630-645, May 1997.

[143] TREADGOLD, N.K.; GEDEON, T.D.; "Exploring Constructive Cascade Networks", *IEEE Transactions on Neural Networks,* v.10, n.6, pp. 1335-1350, Nov. 1999.

[144] CHOUEIKI, M.H.; MOUNT-CAMPBELL, C.A.; AHALT, S.C.; "Building a "Quasi Optimal" Neural Network to Solve the Short-term Load Forecasting Problem", *IEEE Transactions on Power Systems,* v.12, n.4, pp. 1432-1439, Nov. 1997.

[145] CHOUEIKI, M.H.; MOUNT-CAMPBELL, C.A.; AHALT, S.C.; "Implementing a Weighted Least Squares Procedure in Training a Neural

Network to Solve the Short-term Load Forecasting Problem", *IEEE Transactions on Power Systems*, v.12, n.4, pp. 1689-1694, Nov. 1997.

[146] SHYH-JIER, H.; CHING-LIEN, H.; "Genetic-based Multilayered Perceptron for Taiwan Power System Short-term Load Forecasting", *Electric Power Systems Research*, v.38, n.1, pp. 69-74, Jul. 1996.

[147] ABDEL-AAL, R.E.; "Improving Electric Load Forecasts Using Network Committees", *Electric Power Systems Research*, v.74, n.1, pp. 83-94, Apr. 2005.

[148] CARPINTEIRO, O.A.S.; LEME, R.C.; ZAMBRONI DE SOUZA, A.C.; PINHEIRO, C.A.M.; MOREIRA, E.M.; "Long-term load forecasting via a hierarchical neural model with time integrators", *Electric Power Systems Research*, v.77, n.3-4, pp. 371-378, Mar. 2007.

[149] REED, R.; MARKS II, R.J.; OH, S.; "Similarities of Error Regularization, Sigmoid Gain Scaling, Target Smoothing, and Training with Jitter", *IEEE Transactions on Neural Networks*, v.6, n.3, pp. 529-538, May 1995.

[150] POGGIO, T.; GIROSI, F.; "Networks for Approximation and Learning", *Proceedings of the IEEE,* v.78, n.9, pp. 1481-1497, Sept. 1990.

[151] YUAN, J.L.; FINE, T.L.; "Neural-Network Design for Small Training Sets of High Dimension", *IEEE Transactions on Neural Networks*, v.9, n.2, pp. 266-280, Mar. 1998.

[152] LI, K.-C.; "Sliced Inverse Regression for Dimension Reduction", *Journal of American Statistical Association*, v.86, n.404, pp.316-327, 1991.

[153] KANTZ, H.; SCHREIBER, T.; *Nonlinear Time Series Analysis*, Cambridge Nonlinear Science Series, n.7, Cambridge University Press, 1997.

[154] PECORA, L.M.; CARROLL, T.L.; HEAGY, J.F.; "Statistics for Mathematical Properties of Maps between Time Series Embeddings", *Physical Review E*, v.52, n.4, pp. 3420-3441, Oct. 1995.

[155] FRASER, A.M.; SWINNEY, H.L.; "Independent Coordinates for Strange Attractors from Mutual Information", *Physical Review A*, v.33, n.2, pp. 1134-1140, Feb. 1986.

[156] ABARBANEL, H.D.I.; BROWN, R.; SIDOROWICH, J.J.; TSIMRING, L.S.; "The Analysis of Observed Chaotic Data in Physical Systems", *Reviews of Modern Physics*, v.65, n.4, pp. 1331-1392, Oct. 1993.

[157] BUZUG, T.; REIMERS, T.; PFISTER, G.; "Optimal Reconstruction of Strange Attractors from Purely Geometrical Arguments", *Europhysics Letters*, v.13, n.7, pp. 605-610, Dec. 1990.

[158] ROSENSTEIN, M.T.; COLLINS, J.J.; DE LUCA, C.J.; "Reconstruction Expansion as a Geometry-based Framework for Choosing Proper Delay Times", *Physica D*, v.73, n.1-2, pp. 82-98, May 1994.

[159] KENNEL, M.B.; BROWN, R.; ABARBANEL, H.D.I.; "Determining Embedding Dimension for Phase-space Reconstruction Using a Geometrical Construction", *Physical Review A*, v.45, n.6, pp. 3403-3411, Mar. 1992.

[160] CAO, L.; "Practical Method for Determining the Minimum Embedding Dimension of a Scalar Time Series", *Physica D*, v.110, n.1-2, pp. 43-50, Dec. 1997.

[161] GRIFFITHS, W.E.; HILL, R.C.; JUDGE, G.G.; *Learning and Practicing Econometrics*, John Wiley & Sons, 1993.

[162] GUJARATI, D.N.; *Econometria Básica*, Makron Books, 2000.

[163] SANO, M.; SAWADA, Y.; "Measurement of the Lyapunov Spectrum from a Chaotic Time Series", *Physical Review Letters*, v.55, n.10, pp. 1082-1084, Sep. 1985.

[164] ZENG. X.; EYKHOLT, R.; PIELKE, R.A.; "Estimating the Lyapunov-Exponent Spectrum from Short Time Series of Low Precision", *Physical Review Letters*, v.66, n.25, pp. 3229-3232, Jun. 1991.

[165] ECKMANN, J.-P.; KAMPHORST, S.O.; RUELLE, D.; CILIBERTO, S.; "Liapunov Exponents from Time Series", *Physical Review A*, v.34, n.6, pp. 4971-4979, Dec. 1986.

[166] DARBYSHIRE, A.G.; BROOMHEAD, D.S.; "Robust Estimation of Tangent Maps and Liapunov Spectra", *Physica D*, v.89, n.3-4, pp.287-305, Jan. 1996.

[167] BROWN, R.; BRYANT, P., ABARBANEL, H.D.I.; "Computing the Lyapunov Spectrum of a Dynamical System from an Observed Time Series", *Physical Review A*, v.43, n.6, pp. 2787-2806, Mar. 1991.

[168] GENÇAY, R.; DECHERT, W.D.; "An Algorithm for the N Lyapunov Exponents of an N-dimensional Unkown Dynamical System", *Physica D*, v.59, n.1-3, pp. 142-157, Oct. 1992.

[169] PYRAGAS, K.; "Conditional Lyapunov Exponents from Time Series", *Physical Review E*, v.56, n.5, pp. 5183-5188, Nov. 1997.

[170] GENÇAY, R.; "A Statistical Framework for Testing Chaotic Dynamics via Lyapunov Exponents", *Physica D*, v.89, n.3-4, pp. 261-266, Jan. 1996.

[171] PECORA, L.M.; CARROLL, T.L.; "Synchronization in Chaotic Systems", *Physical Review Letters*, v.64, n.8, pp. 821-825, Feb. 1990.

[172] GOUTTE, C.; *Statistical Learning and Regularization for Regression: Application to System Identification and Time Series Modelling*, Ph.D. dissertation, Université Paris 6, Paris, France, 1997.

[173] LEE, H.K.H., "A Noninformative Prior for Neural Networks", *Machine Learning*, v.50, n.1-2, pp. 197-212, Jan. 2003.

[174] NEAL, R.M., *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics, n.118, Springer-Verlag, New York, 1996.

[175] BISHOP, C.M.; "Exact Calculation of the Hessian Matrix for the Multi-layer Perceptron", *Neural Computation,* v.4, n.4, pp. 494-501, 1992.

[176] HAYKIN, S.; *Redes Neurais: Princípios e Prática,* 2ª. Edição, Porto Alegre, RS, Brasil, Editora Bookman, 2001.

[177] CHERKASSKY, V.; MULIER, F.; *Learning from Data: Concepts, Theory and Methods*, John Wiley & Sons, New York, USA, 1998.

[178] SMOLA, A.J., SCHÖLKOPF, B.; "A Tutorial on Support Vector Regression", *Statistics and Computing*, v.14, n.3, pp.199–222, Aug. 2004.

[179] CRISTIANINI, N.; CAMPBELL, C.; SHAWE-TAYLOR, J.; *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.

[180] VAPNIK, V.; CHAPELLE, O.; "Bounds on Error Expectation for Support Vector Machines", *Neural Computation*, v.12, n.9, pp. 2013-2036, Sep. 2000.

[181] CHAPELLE, O.; VAPNIK, V., BOUSQUET, O.; MUKHERJEE, S.; "Choosing Multiple Parameters for Support Vector Machines", *Machine Learning*, v.46, n.1-3, pp. 131-159, Jan. 2002.

[182] VAPNIK, V.; *The Nature of Statistical Learning Theory*, Springer, 1995.

[183] CHANG, M.-W.; LIN, C.-J.; "Leave-One-Out Bounds for Support Vector Regression Model Selection", *Neural Computation*, v.17, n.5, pp. 1188-1222, May 2005.

[184] RAKOTOMAMONJY, A.; "Analysis of SVM Regression Bounds for Variable Ranking", *Neurocomputing*, v.70, n.7-9, pp. 1489-1501, Mar. 2007.

[185] CHERKASSKY, V.; MA, Y.; "Practical Selection of SVM Parameters and Noise Estimation for SVM Regression", *Neural Networks*, v.17, n.1, pp. 113-126, Jan. 2004.

[186] BISHOP, C.M.; TIPPING, M.E.; "Variational Relevance Vector Machines", In: BOUTILIER, C.; GOLDSZMIDT, M. (eds.), *Proceedings of the 16th. Conference on Uncertainty in Artificial Intelligence*, pp. 46-53, Morgan Kaufmann, 2000.

[187] FAUL, A.C.; TIPPING, M.E.; "Analysis of Sparse Bayesian Learning", In: DIETTERICH, T.G.; BECKER, S.; GHAHRAMANI, Z. (eds.), *Advances in Neural Information Processing Systems*, n.14, pp. 383-389, MIT Press, 2002.

[188] NABNEY, I.T.; *NETLAB: Algorithms for Pattern Recognition*, Springer-Verlag, 2002.

[189] CHANG, C.-C.; LIN, C.-J.; *LIBSVM: A Library for Support Vector Machines*, 2001 (available at http://www.csie.ntu.edu.tw/~cjlin/libsvm).

[190] FAN, R.-E; CHEN, P.-H; LIN, C.-J.; "Working Set Selection Using Second Order Information for Training Support Vector Machines", *The Journal of Machine Learning Research*, v.6, pp. 1889-1918, Dec. 2005.

[191] SILVERMAN, B.W.; *Density Estimation for Statistics and Data Analysis*, Monographs on Statistics and Applied Probability, n.26, Chapman & Hall, 1994.

[192] ZHANG, X.; KING, M.L.; HYNDMAN, R.J.; "A Bayesian Approach to Bandwidth Selection for Multivariate Kernel Density Estimation", *Computational Statistics & Data Analysis*, v.50, n.11, pp. 3009-3031, Jul. 2006.

[193] SCOTT, D.W.; "On Optimal and Data-based Histograms", *Biometrika*, v.66, n.3, pp. 605-610, Dec. 1979.

[194] FUKUNAGA, K.; *Introduction to Statistical Pattern Recognition*, 2nd. Edition, Academic Press, 1990.

[195] SHWARTZ, S.; ZIBULEVSKY, M.; SCHECHNER, Y.Y.; "Fast Kernel Entropy Estimation and Optimization", *Signal Processing*, v.85, n.5, pp. 1045-1058, May 2005.

[196] RUMELHART, D.E.; HINTON, G.E.; WILLIAMS, R.J.; McCLELLAND, J.L.; "Learning Internal Representations by Error Propagation", In: RUMELHART, D.E.; McCLELLAND, J.L. (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of the Cognition,* v.1, chapter 8, Cambridge, Massachusetts, MIT Press, 1986.

[197] WERBOS, P.J.; *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph.D. Thesis, Harvard University, Cambridge, Massachusetts, USA, 1974.

[198] WIDROW, B.; HOFF, M.E.; "Adaptive Switching Circuits", In: *IRE WESCON Convention Record,* pp. 96-104, 1960.

[199] LUENBERGER, D.G.; *Introduction to Linear and Nonlinear Programming*, Addison-Wesley Publishing Company, 1973.

[200] GIL, P.E.; MURRAY, W.; WRIGHT, M.H.; *Practical Optimization*, New York, Academic Press, 1981.

APÊNDICE A – Histogramas e estimadores de *Parzen*

O cálculo da informação mútua $I(X,Y)$ utilizando as expressões (2.15), (2.16) ou (2.19) exige a estimação das distribuições de probabilidade marginais de X e Y, além da distribuição conjunta, a partir de um conjunto de dados D. Para variáveis discretas, estas funções podem ser estimadas diretamente por meio de histogramas. No caso de variáveis contínuas, a estimação das respectivas funções de densidade de probabilidade não é trivial.

A utilização de histogramas para variáveis contínuas requer a definição de intervalos de discretização das variáveis. O valor ótimo para este intervalo, em relação à integral do erro médio quadrático, está relacionado com a desconhecida densidade de probabilidade geradora dos dados. Supondo que esta distribuição seja *gaussiana*, o valor ótimo para o intervalo de discretização h_{hist} é dado por [193]:

$$h_{hist} = 3,49\hat{\sigma}N^{-\frac{1}{3}} \tag{A.1}$$

Em (A.1), $\hat{\sigma}$ representa o desvio padrão amostral,

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2}{N-1}}\,. \tag{A.2}$$

Considerando que a variável aleatória X pertença ao intervalo $[a,b] \in \mathbb{R}$, o qual é dividido em p subintervalos de comprimento h_{hist}, a probabilidade de X pertencer ao *i*-ésimo subintervalo ν_i é estimada por:

$$\hat{P}\left(X \in \nu_i\right) = \frac{1}{N}\sum_{j=1}^{N}\psi\left(x_j, \nu_i\right) \tag{A.3}$$

onde

$$\psi\left(x_j, \nu_i\right) = \begin{cases} 1, \text{ se } x_j \in \nu_i \\ 0, \text{ se } x_j \notin \nu_i \end{cases} \tag{A.4}$$

Em outras palavras, $\hat{P}\left(X \in \nu_i\right)$ é estimada através da contagem do número de realizações de X no conjunto D que pertencem ao intervalo ν_i. A extensão para probabilidade conjunta é direta, com a estimativa sendo dada por:

$$\hat{P}\left(X \in \nu_i, Y \in \upsilon_k\right) = \frac{1}{N}\sum_{j=1}^{N}\psi\left(x_j, \nu_i\right)\psi\left(y_j, \upsilon_k\right) \tag{A.5}$$

De posse das respectivas probabilidades, as equações (2.13), (2.14) e (2.16) podem ser utilizadas diretamente para o cálculo das respectivas entropias e da informação mútua $I\left(X,Y\right)$.

Métodos não-paramétricos como estimadores baseados em *kernel*, conhecidos como estimadores de *Parzen* [194], também podem ser utilizados. Neste contexto, a função de densidade de probabilidade de X, $f\left(x\right)$, é estimada pela expressão [191]:

$$\hat{f}\left(x\right) = \frac{1}{Nh}\sum_{i=1}^{N}K\left(\frac{x - x_i}{h}\right) \tag{A.6}$$

Em (A.6), $K\left(t\right): \mathbb{R} \to \mathbb{R}$ é a função conhecida como *kernel*, com *h* representando um parâmetro, denominado comprimento do *kernel*, relacionado com a suavidade da estimativa. De forma semelhante à escolha do intervalo de discretização h_{hist} em histogramas, a escolha de pequenos valores para *h* dá origem a estimativas ruidosas, que além da função geradora $f\left(x\right)$ modelam características específicas do conjunto de dados disponível. De outra forma, definições de valores elevados para estes parâmetros produzem estimativas demasiadamente suaves, mascarando eventuais traços multi-modais presentes nos dados.

Mantendo a analogia com os histogramas, a definição de valores ótimos para *h* em termos do integral do erro médio quadrático depende do conhecimento da função a ser estimada $f(x)$. Para o caso em que esta distribuição é *gaussiana*, o valor ótimo para o comprimento do *kernel gaussiano* utilizado em (A.6) é dado por [191]:

$$h = 1,06\hat{\sigma}N^{-\frac{1}{5}} \tag{A.7}$$

Em (A.7), $\hat{\sigma}$ representa o desvio padrão amostral, estimado pela equação (A.2). O *kernel gaussiano* é dado por:

$$K(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} \tag{A.8}$$

O valor para o comprimento do *kernel* em (A.7) é ótimo somente para dados gerados a partir de distribuições *gaussianas*. Para conjuntos de dados multi-modais, esta escolha pode produzir estimativas demasiadamente suaves, omitindo a ocorrência de múltiplos modos. Esta questão pode ser amenizada utilizando uma nova estimativa para para h dada por [191]:

$$h = 0,9AN^{-\frac{1}{5}} \tag{A.9}$$

onde

$$A = \min\left(\hat{\sigma}, \frac{R}{1.34}\right) \tag{A.10}$$

com R respondendo pela distância entre quartis para a variável aleatória X.

Os estimadores de *Parzen* também podem ser estendidos para estimação de densidades multivariadas. Neste caso, considerando a variável aleatória $\underline{X} \in \mathbb{R}^n$, a equação (A.6) passa a ser dada por:

$$\hat{f}(\underline{x}) = \frac{1}{Nh^n} \sum_{i=1}^{N} K\left[\frac{1}{h}(\underline{x} - \underline{x}_i)\right] \tag{A.11}$$

A estimativa $\hat{f}(\underline{x})$ em (A.11) pondera de maneira equânime todas as direções de $\underline{x}$. Esta forma de ponderação faria sentido se todas as dimensões de $\underline{x}$ apresentassem mesma dispersão, o que nem sempre é verificado na prática. A escolha de uma matriz de parâmetros, ao invés de um único parâmetro, sobrepõe esta questão. Porém, se a escolha de um único parâmetro não é trivial, a definição de múltiplos parâmetros pode ser proibitiva em espaços de elevada cardinalidade.

Para evitar a busca por um número excessivo de parâmetros, podem ser aplicadas transformações aos dados a fim de obter dispersão idêntica em todas as dimensões. Uma transformação em especial, conhecida como branqueamento, do inglês *whitening* [194], produz conjuntos de dados com matriz de covariância igual à identidade. Para tal, seja $\underline{\underline{\sum}} \in \mathbb{R}^n \times \mathbb{R}^n$ a matriz de covariância amostral de $\underline{X}$, dada por:

$$\underline{\underline{\sum}} = \frac{1}{N}\sum_{i=1}^{N}\left(\underline{x}_i - \underline{\bar{x}}\right)\left(\underline{x}_i - \underline{\bar{x}}\right)^t \tag{A.12}$$

$$\underline{\bar{x}} = \frac{1}{N}\sum_{i=1}^{N}\underline{x}_i$$

A variável aleatória $\underline{Y} \in \mathbb{R}^n$ com matriz de covariância igual à identidade é obtida a partir da seguinte transformação:

$$\underline{Y} = \left(\underline{\underline{\Phi}}\,\underline{\underline{\Lambda}}^{-\frac{1}{2}}\right)^t \underline{X} \tag{A.13}$$

Em (A.13), $\underline{\underline{\Lambda}} \in \mathbb{R}^n \times \mathbb{R}^n$ é uma matriz diagonal contendo os autovalores de $\underline{\underline{\sum}}$, com $\underline{\underline{\Phi}} \in \mathbb{R}^n \times \mathbb{R}^n$ representando a matriz com os respectivos autovetores.

No espaço $\underline{Y}$ onde a dispersão é a mesma em todas as dimensões, a expressão (A.11) pode ser utilizada. De posse da estimativa da densidade de probabilidade de $\underline{Y}$, a respectiva função para a variável aleatória $\underline{X}$ pode ser obtida pela relação:

$$f\left(\underline{x}\right)=\left|\det\left(\underline{\underline{\Phi\Lambda}}^{-\frac{1}{2}}\right)\right|g\left(\underline{y}\right) \tag{A.14}$$

$$\hat{f}\left(\underline{x}\right)=\frac{\left|\det\left(\underline{\underline{\Phi\Lambda}}^{-\frac{1}{2}}\right)\right|}{Nh^{n}}\sum_{i=1}^{N}K\left[\frac{1}{h}\left(\underline{y}-\underline{y}_{i}\right)\right]$$

A utilização de estimadores baseados no método de *Parzen* para o cálculo das distribuições de probabilidade envolvidas na estimação da informação mútua $I\left(X,Y\right)$ entre variáveis contínuas exige a utilização de métodos numéricos para o cálculo dos integrais envolvidos nas equações (2.17) e (2.18). Diante do elevado custo computacional de tais técnicas, especialmente para o caso bidimensional, visto que a entropia é definida como um valor esperado, as expressões (2.17) e (2.18) podem ser substituídas pelas suas estimativas amostrais, dadas por [195]:

$$\hat{h}\left(X\right)=-\sum_{i=1}^{N}\hat{f}\left(x_{i}\right)\log\hat{f}\left(x_{i}\right) \tag{A.15}$$

$$\hat{h}\left(X,Y\right)=-\sum_{i=1}^{N}\sum_{j=1}^{N}\hat{q}\left(x_{i},y_{j}\right)\log\hat{q}\left(x_{i},y_{j}\right) \tag{A.16}$$

De posse das entropias diferenciais individuais e da conjunta, a equação (2.16) pode ser utilizada para o cálculo de $I\left(X,Y\right)$.

A escolha adequada do intervalo de discretização h_{hist} ou do comprimento do *kernel* h constitui a principal tarefa na estimação de probabilidades ou densidades de probabilidade para o cálculo da informação mútua $I\left(X,Y\right)$. As expressões (A.1) e (A.9) apresentam valores obtidos em referência a distribuições *gaussianas*, não existindo garantia acerca da sua otimalidade para outras densidades. Entretanto, a simplicidade das expressões em conjunto com a obtenção de estimativas suaves para as respectivas probabilidades favorecem a sua utilização. Métodos computacionalmente intensivos, como validação cruzada [191] e simulações de Monte Carlo [192], também

podem ser utilizados. Porém, visto que a estimação de densidades de probabilidade constitui o início do processo de desenvolvimento de modelos neurais, o uso de tais técnicas pode comprometer a aplicabilidade do sistema como um todo em termos de esforço computacional requerido.

APÊNDICE B – Algoritmos de treinamento de MLPs

RUMELHART *et. al.* [196] desenvolveram o algoritmo de retropropagação do erro para treinamento de modelos neurais *feedforward* com múltiplas camadas, dando origem ao MLP. Na realidade, este algoritmo foi originalmente desenvolvido por WERBOS [197], em 1974, podendo também ser considerado como uma generalização do algoritmo do mínimo quadrado médio, *least mean square* (LMS), também conhecido como regra delta, desenvolvido por WIDROW e HOFF [198] para filtragem linear adaptativa de sinais. O algoritmo LMS é um caso particular do algoritmo de retropropagação do erro, para o caso em que a rede apresenta um único neurônio linear.

Após o surgimento deste algoritmo, vários algoritmos para treinamento de MLPs foram propostos, porém com uma abordagem diferente. Neste novo contexto, o treinamento de MLPs passou a ser visto como um problema de otimização, com algumas técnicas desta área do conhecimento sendo aplicadas à estimação de parâmetros de modelos neurais, dando origem aos chamados métodos de segunda ordem, como os métodos *quasi-newton* e os métodos de gradiente conjugado.

Os próximos itens apresentarão uma breve descrição das duas abordagens para treinamento de MLPs, começando pelo algoritmo de retropropagação do erro.

B.1. Algoritmo de retropropagação do erro

O algoritmo de retropropagação do erro é um algoritmo supervisionado, visto que necessita de um conjunto de saídas desejadas para estimação dos parâmetros do modelo através da correção do erro gerado para cada saída. Dado um conjunto D contendo N pares entrada-saída, $D = \{\underline{x}_k, \underline{d}_k\}$, $k = 1, 2, ..., N$, $\underline{x}_k = [x_{k1}, x_{k2}, ..., x_{kn}]^t$, $\underline{d}_k = [d_{k1}, d_{k2}, ..., d_{km}]^t$, o objetivo deste algoritmo reside na estimação do vetor de

parâmetros $\underline{w}$ que minimize o erro médio quadrático para este conjunto de dados, também conhecido como risco empírico, dado por:

$$E_s(\underline{w}) = \frac{1}{N}\sum_{i=1}^{N} E_i(\underline{w}) \tag{B.1}$$
$$E_i(\underline{w}) = \frac{1}{2}\left[d_i - f(\underline{x}_i, \underline{w})\right]^2$$

Visto que o erro só pode ser obtido diretamente para os neurônios da camada de saída, a idéia do algoritmo reside na propagação deste erro através da rede, fazendo com que o MLP, além de propagar os sinais de entrada "para frente", propague os sinais de erro em sentido contrário, objetivando a modificação dos pesos sinápticos e dos *bias* de forma a minimizar o funcional descrito na equação (B.1). Daí o nome de retropropagação do erro. A derivação deste algoritmo pode ser encontrada em [50], [176], [196], e [197].

Para os MLPs utilizados nesta tese, que apresentam uma única camada escondida e uma única saída linear, este algoritmo pode ser resumido como segue:

1. Faça $l = 0$.
2. Inicialize o vetor de parâmetros $\underline{w}(l)$.
3. Apresente o conjunto de treinamento $D = \{\underline{x}_k, d_k\}$ ao modelo.
4. Para cada par entrada-saída $\{\underline{x}_k, d_k\}$, efetue os passos 5 a 10.
5. Propague o vetor de entrada $\underline{x}_k$ ao longo da rede, utilizando a equação (2.2).
6. Calcule o erro obtido para este padrão, dado pela equação:

$$e(l) = d_k - f\left[\underline{x}_k, \underline{w}(l)\right] \tag{B.2}$$

7. Atualize os pesos sinápticos w_{21j} que ligam os neurônios da camada intermediária ao neurônio de saída, através da equação:

$$w_{21j}(l+1) = w_{21j}(l) - \eta \left. \frac{\partial E_k(\underline{w})}{\partial w_{21j}} \right|_{\underline{w}=\underline{w}(l)} \tag{B.3}$$

$$\left. \frac{\partial E_k(\underline{w})}{\partial w_{21j}} \right|_{\underline{w}=\underline{w}(l)} = -e(l) f\left[\underline{x}_k, \underline{w}(l)\right]$$

8. Atualize os pesos sinápticos w_{1ij} que ligam as entradas aos neurônios sigmoidais da camada intermediária, dada pela equação:

$$w_{1ij}(l+1) = w_{1ij}(l) - \eta \left. \frac{\partial E_k(\underline{w})}{\partial w_{1ij}} \right|_{\underline{w}=\underline{w}(l)} \tag{B.4}$$

$$\left. \frac{\partial E_k(\underline{w})}{\partial w_{1ij}} \right|_{\underline{w}=\underline{w}(l)} = -\left[\left. \frac{d\varphi(a)}{da} \right|_{a=a(l)} \right] w_{21j}(l) e(l) x_{kj}$$

$$a(l) = \sum_{j=0}^{n_0} w_{1ij}(l) x_{kj}$$

9. Faça $l = l + 1$.

10. Se todos os padrões $\{\underline{x}_k, d_k\}$ foram apresentados ao modelo, vá para o passo 11. Do contrário, escolha um novo padrão $\{\underline{x}_k, d_k\}$ e retorne ao passo 5.

11. Se o critério de parada for atendido, encerre o algoritmo. Do contrário, retorne ao passo 3.

No algoritmo resumido acima, η representa um parâmetro chamado de taxa de aprendizagem e $d\varphi(a)/da$ a derivada da função de ativação sigmoidal dos neurônios da camada oculta em relação ao somatório ponderado das suas entradas. Como critérios de parada, são utilizados: erro médio para todo o conjunto de treinamento, número máximo de apresentações (épocas) do conjunto de treinamento e erro para um conjunto independente de dados. O algoritmo apresentado acima é conhecido como algoritmo de retropropagação do erro seqüencial, visto que os pesos são atualizados após a apresentação de cada par entrada-saída $\{\underline{x}_i, d_i\}$. A atualização dos pesos pode também

ser feita após a apresentação de uma época inteira de treinamento, dando origem ao chamado treinamento por batelada, ou lote. O algoritmo deste modo de treinamento de MLPs pode ser resumido como segue:

1. Faça $l = 0$.
2. Inicialize o vetor de parâmetros $\underline{w}(l)$.
3. Apresente o conjunto de treinamento $D = \{\underline{x}_k, d_k\}$ ao modelo.
4. Para cada par entrada-saída $\{\underline{x}_k, d_k\}$, efetue os passos 5 a 9.
5. Propague o vetor de entrada $\underline{x}_k$ ao longo da rede, utilizando a equação (2.2).
6. Calcule o erro obtido para este padrão, dado pela equação:

$$e(l) = d_k - f\left[\underline{x}_k, \underline{w}(l)\right] \tag{B.5}$$

7. Calcule as derivadas parciais $\partial E_k(\underline{w})/\partial w_{21j}$, relacionadas com os pesos sinápticos w_{21j} que ligam os neurônios da camada intermediária ao neurônio de saída, através da equação:

$$\left.\frac{\partial E_k(\underline{w})}{\partial w_{21j}}\right|_{\underline{w}=\underline{w}(l)} = -e(l) f\left[\underline{x}_k, \underline{w}(l)\right] \tag{B.6}$$

8. Calcule as derivadas parciais $\partial E_k(\underline{w})/\partial w_{1ij}$, relacionadas com os pesos sinápticos w_{1ij} que ligam as entradas aos neurônios sigmoidais da camada intermediária, através da equação:

$$\left.\frac{\partial E_k(\underline{w})}{\partial w_{1ij}}\right|_{\underline{w}=\underline{w}(l)} = -\left[\left.\frac{d\varphi(a)}{da}\right|_{a=a(l)}\right] w_{21j}(l)\, e(l)\, x_{kj} \tag{B.7}$$

$$a(l) = \sum_{j=0}^{n_0} w_{1ij}(l)\, x_{kj}$$

9. Se todos os padrões $\{\underline{x}_k, d_k\}$ foram apresentados ao modelo, vá para o passo 10. Do contrário, escolha o próximo padrão $\{\underline{x}_k, d_k\}$ do conjunto de treinamento *D* e retorne ao passo 5.

10. Atualize os pesos sinápticos w_{21j} que ligam os neurônios da camada intermediária ao neurônio de saída, através da equação:

$$w_{21j}(l+1) = w_{21j}(l) - \eta \left.\frac{\partial E_s(\underline{w})}{\partial w_{21j}}\right|_{\underline{w}=\underline{w}(l)} \tag{B.8}$$

$$\left.\frac{\partial E_s(\underline{w})}{\partial w_{21j}}\right|_{\underline{w}=\underline{w}(l)} = -\frac{1}{N}\sum_{k=1}^{N} \left.\frac{\partial E_k(\underline{w})}{\partial w_{21j}}\right|_{\underline{w}=\underline{w}(l)}$$

11. Atualize os pesos sinápticos w_{1ij} que ligam as entradas aos neurônios sigmoidais da camada intermediária, dada pela equação:

$$w_{1ij}(l+1) = w_{1ij}(l) - \eta \left.\frac{\partial E_s(\underline{w})}{\partial w_{1ij}}\right|_{\underline{w}=\underline{w}(l)} \tag{B.9}$$

$$\left.\frac{\partial E_s(\underline{w})}{\partial w_{1ij}}\right|_{\underline{w}=\underline{w}(l)} = -\frac{1}{N}\sum_{k=1}^{N} \left.\frac{\partial E_k(\underline{w})}{\partial w_{1ij}}\right|_{\underline{w}=\underline{w}(l)}$$

12. Faça $l = l+1$.

13. Se o critério de parada for atendido, encerre o algoritmo. Do contrário, retorne ao passo 3.

Assim como toda técnica baseada em descida em gradiente, categoria na qual o algoritmo de retropropagação de erro está enquadrado, conforme evidenciado nas equações (B.3) e (B.4), este algoritmo apresenta uma série de desvantagens. A existência de múltiplos mínimos locais em virtude da característica multi-modal da função a ser minimizada, quadrática em relação às saídas e extremamente não-linear em relação ao vetor de parâmetros $\underline{w}$, compromete o desempenho do algoritmo. Para contornar esse problema, existe uma série de heurísticas propostas para modificação do

algoritmo resumido acima, como inserção de uma parcela de momento nas equações (B.3) e (B.4), para o treinamento seqüencial, e (B.8) e (B.9), para o treinamento por batelada, normalização do conjunto de entrada-saída no intervalo [-1;1] e estratégias de escolha do conjunto inicial de pesos $\underline{w}_0$. Com o intuito de sobrepujar as limitações do algoritmo de retropropagação de erro, foram desenvolvidos os chamados métodos de segunda ordem.

B.2. Métodos de segunda ordem

O treinamento supervisionado de MLP's pode ser visto também como um problema de otimização. Para tanto, seja a expansão, em séries de *Taylor*, do funcional descrito na equação (B.1), desprezando os termos de ordem de superior, em torno de um ponto específico $\underline{w}(l)$ no espaço de pesos, dada por:

$$E_s\left[\underline{w}(l)+\Delta\underline{w}(l)\right]=E_s\left[\underline{w}(l)\right]+\left[\left.\frac{\partial E_s(\underline{w})}{\partial\underline{w}}\right|_{\underline{w}=\underline{w}(l)}\right]^t\Delta\underline{w}(l)+\frac{1}{2}\Delta\underline{w}^t(l)\left[\left.\frac{\partial^2 E_s(\underline{w})}{\partial\underline{w}^2}\right|_{\underline{w}=\underline{w}(l)}\right]\Delta\underline{w}(l) \quad \text{(B.10)}$$

$$\nabla E_s(\underline{w})=\frac{\partial E_s(\underline{w})}{\partial\underline{w}}=\left[\frac{\partial E_s(\underline{w})}{\partial w_1},\frac{\partial E_s(\underline{w})}{\partial w_2},\cdots,\frac{\partial E_s(\underline{w})}{\partial w_M}\right]^t$$

$$\underline{\underline{H}}(\underline{w})=\frac{\partial^2 E_s(\underline{w})}{\partial\underline{w}^2}=\begin{bmatrix}\frac{\partial^2 E_s(\underline{w})}{\partial w_1^2} & \frac{\partial^2 E_s(\underline{w})}{\partial w_1\partial w_2} & \cdots & \frac{\partial^2 E_s(\underline{w})}{\partial w_1\partial w_M}\\ \frac{\partial^2 E_s(\underline{w})}{\partial w_2\partial w_1} & \frac{\partial^2 E_s(\underline{w})}{\partial w_2^2} & \cdots & \frac{\partial^2 E_s(\underline{w})}{\partial w_2\partial w_M}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2 E_s(\underline{w})}{\partial w_M\partial w_1} & \frac{\partial^2 E_s(\underline{w})}{\partial w_M\partial w_2} & \cdots & \frac{\partial^2 E_s(\underline{w})}{\partial w_M^2}\end{bmatrix}$$

Na equação (B.10), $\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}$ representa o vetor gradiente, calculado no ponto $\underline{w}(l)$ e $\underline{\underline{H}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}$ a matriz *hessiana* calculada no mesmo ponto. A expressão (B.10) realiza uma aproximação quadrática, em torno do ponto $\underline{w}(l)$, da superfície de erro $E_s(\underline{w})$ no espaço de pesos. Portanto, a modificação $\Delta\underline{w}(l)$ que deve ser aplicada aos pesos

sinápticos $\underline{w}(l)$ de forma a obter o ponto de mínimo dessa superfície quadrática aproximada é dada por:

$$\nabla E_s\left[\underline{w}(l)+\Delta\underline{w}(l)\right]=\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}+\left[\underline{\underline{H}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right]\Delta\underline{w}(l)=0 \tag{B.11}$$

$$\Delta\underline{w}(l)=-\left[\underline{\underline{H}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right]^{-1}\left[\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right]$$

Na equação (B.11), $\left[\underline{\underline{H}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right]^{-1}$ representa a inversa da matriz *hessiana*. A equação (B.11) é o princípio do método de *Newton*, apresentando as seguintes desvantagens quando aplicado diretamente ao treinamento de MLPs:

- Ausência de garantia da existência da inversa da matriz *hessiana* $\underline{\underline{H}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}$, devido à possibilidade de existência de colunas desta matriz linearmente dependentes.
- Se $\underline{\underline{H}}\left(\underline{w}\right)$ for inversível, o cálculo da sua inversa pode ser computacionalmente custoso para problemas de grande porte.
- A modificação $\Delta\underline{w}(l)$ dada pela equação (B.11) conduz à minimização do funcional $E_s\left(\underline{w}\right)$, ou seja, $E_s\left[\underline{w}(l)+\Delta\underline{w}(l)\right]<E_s\left[\underline{w}(l)\right]$, somente para os casos em que a matriz $\underline{\underline{H}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}$ é positiva definida, ou seja, com todos auto-valores maiores que zero, o que não é sempre válido para o MLP.
- A convergência do método de *Newton* é garantida apenas para casos em que o funcional $E_s\left(\underline{w}\right)$ é quadrático em relação aos parâmetros $\underline{w}$, convergindo em uma única iteração. Entretanto, essa condição não é satisfeita para o MLP.

Apesar das deficiências acima citadas da aplicação direta do método de *Newton* para treinamento de MLPs, algumas das suas características vantajosas podem ser obtidas através da aplicação dos chamados métodos *quasi-newton*, que não requerem o

cálculo direto de $\left[\underline{\underline{H}}\left(\underline{w}\right)\right]^{-1}$, e sim uma estimativa desta utilizando apenas informação do gradiente $\nabla E_s\left(\underline{w}\right)$. Uma outra forma de utilizar informação de segunda ordem para treinamento de MLPs reside na aplicação dos métodos baseados em gradiente conjugado, que buscam a combinação entre a descida em gradiente, base do algoritmo de retropropagação de erro apresentado na seção B.1 deste apêndice, e a informação de segunda ordem contida na matriz *hessiana* $\underline{\underline{H}}\left(\underline{w}\right)$, sem a necessidade de cálculo explícito da mesma.

Os métodos baseados em gradiente conjugado, também conhecidos como métodos de direção conjugada [199], podem ser considerados como métodos intermediários entre aqueles baseados em descida em gradiente e o método de *Newton*. Estes métodos foram desenvolvidos com o intuito de acelerar a convergência da descida em gradiente, sem o esforço computacional extensivo necessário para a aplicação do método de *Newton.* A derivação do método de otimização baseado em gradiente conjugado pode ser encontrada em [199] e [200], com a sua aplicação ao treinamento de MLPs podendo ser encontrada em [50] e [176].

De uma maneira geral, o algoritmo para treinamento de MLPs baseado em gradiente conjugado pode ser resumido como segue [50]:

1. Faça $l = 0$.
2. Escolha o vetor inicial de pesos $\underline{w}\left(l\right)$.
3. Utilizando o algoritmo de retropropagação do erro por batelada, calcule $\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}$.
4. Determine a direção inicial de busca através da equação:

$$\underline{d}\left(l\right) = -\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)} \tag{B.12}$$

5. Resolva o problema de otimização irrestrito dado por:

$$\min_{\alpha(l)} E_s\left[\underline{w}(l)+\alpha(l)\underline{d}(l)\right] \tag{B.13}$$

6. Obtida a solução $\alpha_{\min}(l)$ do problema de otimização descrito na equação (B.13), atualize o vetor de pesos $\underline{w}$ através da equação:

$$\underline{w}(l+1)=\underline{w}(l)+\alpha_{\min}(l)d(l) \tag{B.14}$$

7. Se o critério de parada for atendido para $\underline{w}=\underline{w}(l+1)$, encerre o algoritmo. Do contrário, vá para o passo 8.
8. Utilizando o algoritmo de retropropagação do erro por batelada, calcule $\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}$.
9. Calcule a nova direção de busca através da equação:

$$\underline{d}(l+1)=-\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}+\beta(l)\underline{d}(l) \tag{B.15}$$

10. Faça $l=l+1$ e retorne ao passo 5.

No algoritmo descrito acima, os parâmetros $\alpha_{\min}(l)$ e $\beta(l)$ são responsáveis pelo passo e pela direção da busca, respectivamente. Enquanto $\alpha_{\min}(l)$ é obtido através da solução do problema de minimização dado por (B.13), duas expressões podem ser utilizadas para obtenção do parâmetro $\beta(l)$. Uma delas, conhecida como fórmula de *Polak-Ribiere*, é dada por [50]:

$$\beta(l)=\frac{\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}\right]^t\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}-\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]}{\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]^t\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]} \tag{B.16}$$

A segunda, chamada de fórmula de *Fletcher-Reeves*, é dada por [50]:

$$\beta(l)=\frac{\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}\right]^t\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}\right]}{\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]^t\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]} \tag{B.17}$$

As expressões (B.16) e (B.17) são equivalentes para problemas de otimização quadrática, onde o treinamento de MLPs não está inserido. No contexto de MLPs, a fórmula de *Polak-Ribiere* apresenta melhores resultados, visto que à medida que são obtidos sucessivos vetores $\nabla E_s\left(\underline{w}\right)$ similares ao longo do algoritmo, $\beta(l)$ tende a zero, reiniciando a busca na forma de descida em gradiente [50]. Entretanto, a convergência do algoritmo baseado em gradiente conjugado utilizando a equação (B.16) é garantida apenas se esta equação sofrer a seguinte modificação:

$$\beta(l) = \max\left\{\frac{\left[\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l+1)}\right]^t\left[\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l+1)} - \nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right]}{\left[\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right]^t\left[\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right]}, 0\right\} \tag{B.18}$$

Pela expressão (B.18), o algoritmo baseado em gradiente conjugado deve ser reiniciado, começando a busca utilizando a direção da descida em gradiente, para $\beta(l) \leq 0$ [176].

Os chamados métodos *quasi-newton*, como o próprio nome já diz, utilizam a idéia básica do método de *Newton,* buscando superar as deficiências do mesmo quando aplicado ao treinamento de MLPs. Nestes métodos, é calculada uma estimativa da matriz *hessiana* $\underline{\underline{H}}\left(\underline{w}\right)$ utilizando apenas informações do gradiente $\nabla E_s\left(\underline{w}\right)$. Para tal, seja $E_s\left(\underline{w}\right): \mathbb{R}^M \rightarrow \mathbb{R}$, um funcional com derivadas de segunda ordem contínuas, dois pontos consecutivos, $\underline{w}(l+1)$ e $\underline{w}(l)$, e uma constante θ, $0 < \theta < 1$. Pelo teorema do valor médio, a seguinte expressão é obtida [199]:

$$\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l+1)} - \nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)} = \left\{\underline{\underline{H}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)+\theta\left[\underline{w}(l+1)-\underline{w}(l)\right]}\right\}\left[\underline{w}(l+1) - \underline{w}(l)\right] \tag{B.19}$$

Para o caso em que a matriz *hessiana* $\underline{\underline{H}}\left(\underline{w}\right)$ é constante, pressuposto inicial dos métodos *quasi-newton* [50], a equação (B.19) passa a ser dada por:

$$\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l+1)} - \nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)} = \underline{\underline{H}}\left(\underline{w}\right)\left[\underline{w}(l+1) - \underline{w}(l)\right] \tag{B.20}$$

A equação (B.20) mostra que o cálculo do gradiente $\nabla E_s\left(\underline{w}\right)$ em dois pontos consecutivos fornece informação sobre a matriz *hessiana* $\underline{\underline{H}}\left(\underline{w}\right)$. Sejam $\underline{\underline{P}}\left(\underline{w}\right)$ e $\underline{\underline{Q}}\left(\underline{w}\right)$ matrizes de dimensão $M \times M$, dadas por:

$$\underline{\underline{P}}\left(\underline{w}\right) = \begin{bmatrix} w_1(1) - w_1(0) & w_1(2) - w_1(1) & \cdots & w_1(M) - w_1(M-1) \\ w_2(1) - w_2(0) & w_1(2) - w_1(1) & \cdots & w_2(M) - w_2(M-1) \\ \vdots & \vdots & \ddots & \vdots \\ w_M(1) - w_M(0) & w_M(2) - w_M(1) & \cdots & w_M(M) - w_M(M-1) \end{bmatrix} \tag{B.21}$$

$$\underline{\underline{Q}}\left(\underline{w}\right) = \begin{bmatrix} \dfrac{\partial E_s\left(\underline{w}\right)}{\partial w_1}\Big|_{\underline{w}(1)} - \dfrac{\partial E_s\left(\underline{w}\right)}{\partial w_1}\Big|_{\underline{w}(0)} & \cdots & \dfrac{\partial E_s\left(\underline{w}\right)}{\partial w_1}\Big|_{\underline{w}(M)} - \dfrac{\partial E_s\left(\underline{w}\right)}{\partial w_1}\Big|_{\underline{w}(M-1)} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial E_s\left(\underline{w}\right)}{\partial w_M}\Big|_{\underline{w}(1)} - \dfrac{\partial E_s\left(\underline{w}\right)}{\partial w_M}\Big|_{\underline{w}(0)} & \cdots & \dfrac{\partial E_s\left(\underline{w}\right)}{\partial w_M}\Big|_{\underline{w}(M)} - \dfrac{\partial E_s\left(\underline{w}\right)}{\partial w_M}\Big|_{\underline{w}(M-1)} \end{bmatrix} \tag{B.22}$$

Se as *M* direções $\underline{w}(n+1) - \underline{w}(n)$ forem linearmente independentes, utilizando a expressão (B.20), a matriz *hessiana* $\underline{\underline{H}}\left(\underline{w}\right)$ pode ser estimada pela equação:

$$\underline{\underline{H}}\left(\underline{w}\right) = \underline{\underline{Q}}\left(\underline{w}\right)\left[\underline{\underline{P}}\left(\underline{w}\right)\right]^{-1} \tag{B.23}$$

Desta forma, a estimativa $\underline{\underline{S}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l+1)}$ da matriz *hessiana* $\underline{\underline{H}}\left(\underline{w}\right)$ para a $(n+1)$-ésima iteração é dada por:

$$\left[\underline{\underline{S}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(n+1)}\right]\left[\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l+1)} - \nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right] = \underline{w}(l+1) - \underline{w}(l),\ l = 0,...,M-1 \tag{B.24}$$

A cada iteração da equação (B.24), é obtido um sistema linear contendo *M* equações e M^2 incógnitas, referentes aos $M \times M$ elementos da matriz $\underline{\underline{S}}\left(\underline{w}\right)$. Portanto, após a aplicação de *M* direções $\underline{w}(l+1) - \underline{w}(l)$ linearmente independentes, é obtido um sistema linear contendo M^2 equações e M^2 incógnitas, cuja solução fornece a

estimativa final $\underline{\underline{S}}(\underline{w}) = \left[\underline{\underline{H}}(\underline{w})\right]^{-1}$. Porém, se o número de direções linearmente independentes for menor que *M*, o sistema linear obtido apresenta mais equações que incógnitas, resultando em infinitas soluções para a matriz $\underline{\underline{S}}(\underline{w})$ [199].

Para abordar o caso em que são possíveis infinitas soluções para a estimativa $\underline{\underline{S}}(\underline{w})$, foi proposto o método de *Davidon-Fletcher-Powell* (DFP), que, para treinamento de MLPs, pode ser resumido da forma que segue:

1. Faça $l = 0$.
2. Escolha uma matriz de dimensão $M \times M$ definida positiva como estimativa inicial da matriz $\underline{\underline{S}}(\underline{w})$.
3. Escolha o vetor inicial de parâmetros $\underline{w}(l)$.
4. Utilizando o algoritmo de retropropagação do erro por batelada, calcule $\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}$.
5. Faça $\underline{d}(l) = -\underline{\underline{S}}(\underline{w})\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]$.
6. Resolva o problema de otimização dado por:

$$\begin{aligned} &\min_{\alpha} E_s\left[\underline{w}(l) + \alpha(l)\underline{d}(l)\right] \\ &s.a \\ &\alpha \geq 0 \end{aligned} \tag{B.25}$$

7. Obtida a solução $\alpha_{\min}(l)$ do problema de otimização descrito na equação (B.25), atualize o vetor de pesos $\underline{w}$ através da equação:

$$\underline{w}(l+1) = \underline{w}(l) + \alpha_{\min}(l)\underline{d}(l) \tag{B.26}$$

8. Se o critério de parada for atendido para $\underline{w} = \underline{w}(l+1)$, encerre o algoritmo. Do contrário, vá para o passo 9.

9. Utilizando o algoritmo de retropropagação do erro por batelada, calcule $\nabla E_s\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l+1)}$.

10. Calcule a nova estimativa $\underline{\underline{S}}\left(\underline{w}\right)$ através da equação:

$$\underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l+1)} = \underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)} + \frac{\left[\alpha_{\min}\left(l\right)\underline{d}\left(l\right)\right]\left[\alpha_{\min}\left(l\right)\underline{d}\left(l\right)\right]^t}{\left[\alpha_{\min}\left(l\right)\underline{d}\left(l\right)\right]^t\left[\underline{q}\left(l\right)\right]} - \frac{\left[\underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}\left(l\right)\right]\left[\underline{q}\left(l\right)\right]^t\left[\underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)}\right]}{\left[\underline{q}\left(l\right)\right]^t\left[\underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}\left(l\right)\right]} \tag{B.27}$$

$$\underline{q}\left(l\right) = \nabla E_s\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l+1)} - \nabla E_s\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)}$$

11. Faça $l = l+1$ e retorne ao passo 4.

O algoritmo descrito acima foi o primeiro dos chamados métodos *quasi-newton* de otimização. Atualmente, o melhor método *quasi-newton* é o chamado método de *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) [176], cuja única modificação em relação ao DFP reside na estimativa recursiva da matriz $\underline{\underline{S}}\left(\underline{w}\right)$, originalmente dada pela equação (B.27), que passa a ser dada por:

$$\underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l+1)} = \underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)} + \frac{\left[\alpha_{\min}\left(l\right)\underline{d}\left(l\right)\right]\left[\alpha_{\min}\left(l\right)\underline{d}\left(l\right)\right]^t}{\left[\alpha_{\min}\left(l\right)\underline{d}\left(l\right)\right]^t\left[\underline{q}\left(l\right)\right]} - \frac{\left[\underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}\left(l\right)\right]\left[\underline{q}\left(l\right)\right]^t\left[\underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)}\right]}{\left[\underline{q}\left(l\right)\right]^t\left[\underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}\left(l\right)\right]} + \left[\underline{q}\left(l\right)\right]^t\left[\underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}\left(l\right)\right]\left[\underline{u}\left(l\right)\right]\left[\underline{u}\left(l\right)\right]^t \tag{B.28}$$

Na equação (B.28), $\underline{u}\left(l\right)$ é dado por:

$$\underline{u}\left(l\right) = \frac{\alpha_{\min}\left(l\right)\underline{d}\left(l\right)}{\left[\alpha_{\min}\left(l\right)\underline{d}\left(l\right)\right]^t\left[\underline{q}\left(l\right)\right]} - \frac{\left[\underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}\left(l\right)\right]}{\left[\underline{q}\left(l\right)\right]^t\left[\underline{\underline{S}}\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}\left(l\right)\right]} \tag{B.29}$$

Tanto o método baseado em gradiente conjugado quanto os métodos *quasi-newton* utilizam a cada iteração uma aproximação quadrática, em torno do ponto $\underline{w}(l)$, de um funcional arbitrário $E_s(\underline{w})$. Para o caso específico em que $E_s(\underline{w})$ é dado pela equação (B.1), ou seja, para problemas de minimização do erro médio quadrático, onde o treinamento de MLPs está inserido, existe o método de *Levenberg-Marquardt*, que, assim como os métodos *quasi-newton*, utiliza uma aproximação da matriz *hessiana* $\underline{\underline{H}}(\underline{w})$ tomando por base informações do gradiente e do erro para cada padrão. A matriz *jacobiana* $\underline{\underline{J}}(\underline{w})$ relacionada com o funcional $E_k(\underline{w})$ dado pela equação (B.1) é definida como segue:

$$\underline{\underline{J}}(\underline{w}) = \left[\frac{\partial}{\partial w_1} E_k(\underline{w}) \quad \cdots \quad \frac{\partial}{\partial w_M} E_k(\underline{w}) \right] \tag{B.30}$$

Desta forma, o gradiente $\nabla E_k(\underline{w})$ e a matriz *hessiana* $\underline{\underline{H}}(\underline{w})$ relacionada com o funcional $E_k(\underline{w})$ são dados pelas equações:

$$\nabla E_k(\underline{w}) = \left[\underline{\underline{J}}(\underline{w}) \right]^t E_k(\underline{w}) \tag{B.31}$$

$$\underline{\underline{H}}(\underline{w}) = \left[\underline{\underline{J}}(\underline{w}) \right]^t \underline{\underline{J}}(\underline{w}) + E_k(\underline{w}) \underline{\underline{H}}_k(\underline{w}) \tag{B.32}$$

$$\underline{\underline{H}}_k(\underline{w}) = \begin{bmatrix} \frac{\partial^2}{\partial {w_1}^2} E_k(\underline{w}) & \cdots & \frac{\partial^2}{\partial w_1 \partial w_M} E_k(\underline{w}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial w_M \partial w_1} E_k(\underline{w}) & \cdots & \frac{\partial^2}{\partial {w_M}^2} E_k(\underline{w}) \end{bmatrix}$$

Desprezando os termos de segunda ordem, ou seja, fazendo $\underline{\underline{H}}_k(\underline{w}) \approx \underline{\underline{0}}$, a regra de atualização dos pesos dada pela equação (B.11), princípio do método de *Newton*, passa a ser dada por:

$$\Delta(\underline{w}) = -\left\{ \left[\underline{\underline{J}}(\underline{w}) \right]^t \underline{\underline{J}}(\underline{w}) \right\}^{-1} \left[\underline{\underline{J}}(\underline{w}) \right]^t \underline{e}(\underline{w}) \tag{B.33}$$

A utilização direta da equação (B.33) pode resultar em passos de atualização de magnitude elevada, conduzindo a soluções onde aproximação $\underline{\underline{S}}(\underline{w}) \approx \underline{\underline{0}}$ não é válida, comprometendo a eficiência do algoritmo [50]. Para garantir que o algoritmo realize a busca apenas na região onde esta aproximação é válida, o algoritmo de *Levenberg-Marquardt* utiliza a seguinte modificação da equação (B.33):

$$\Delta(\underline{w}) = -\left\{\left[\underline{\underline{J}}(\underline{w})\right]^t \underline{\underline{J}}(\underline{w}) + \lambda \underline{\underline{I}}\right\}^{-1} \left[\underline{\underline{J}}(\underline{w})\right]^t \underline{e}(\underline{w}) \tag{B.34}$$

Na equação (B.34), $\underline{\underline{I}}$ é a matriz identidade de dimensão $M \times M$ e λ é uma constante, relacionada com o tamanho da região onde a aproximação $\underline{\underline{S}}(\underline{w}) \approx \underline{\underline{0}}$ é válida. Desta forma, o algoritmo de *Levenberg-Marquardt* pode ser considerado como um algoritmo de otimização em regiões viáveis, visto que limita a busca apenas em regiões no entorno do ponto de operação, onde as aproximações consideradas pelo método são válidas [50] e [200]. Na prática, o valor da constante λ deve ser modificado ao longo do processo de otimização. Uma forma de atualização bastante utilizada consiste em fazer $\lambda = 0.1$ no início do processo iterativo, e, se o erro diminuir para a iteração n, diminuir λ em uma ordem de grandeza, ou seja, $\lambda(n+1) = 0.1\lambda(n)$. Em caso contrário, aumentar em uma ordem de grandeza, ou seja, $\lambda(n+1) = 10\lambda(n)$.

Para os MLPs utilizados nesta tese, contendo uma única camada escondida e uma única saída linear, o algoritmo de *Levenberg-Marquardt* para treinamento de MLPs pode ser resumido como segue:

1. Faça $l = 0$.
2. Inicialize o vetor de parâmetros $\underline{w}(l)$.
3. Faça $\lambda(l) = 0.1$.

4. Utilizando o algoritmo de retropropagação do erro seqüencial, calcule o vetor gradiente $\nabla E_k\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}$.

5. Calcule a matriz *jacobiana* $\underline{\underline{J}}\left(\underline{w}\right)$ através da equação:

$$\underline{\underline{J}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)} = \frac{1}{e(l)}\left[\nabla E_k\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right]^t \tag{B.35}$$
$$e(l) = d_k - f\left[\underline{x}_k, \underline{w}(l)\right]$$

6. Atualize o vetor de parâmetros $\underline{w}$ através da equação:

$$\underline{w}(l+1) = \underline{w}(l) - \left\{\left[\underline{\underline{J}}\left(\underline{w}\right)\right]^t \underline{\underline{J}}\left(\underline{w}\right) + \lambda(l)\underline{\underline{I}}\right\}^{-1}\left[\underline{\underline{J}}\left(\underline{w}\right)\right]^t \underline{e}\left(\underline{w}\right) \tag{B.36}$$

7. Atualize a constante λ através da equação:

$$\lambda(l+1) = \begin{cases} 0.1\lambda(l), \text{ se } E_k\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l+1)} < E_k\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)} \\ 10\lambda(l), \text{ se } E_k\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l+1)} > E_k\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)} \end{cases} \tag{B.37}$$

8. Se o critério de parada for atendido, encerre o algoritmo. Do contrário, faça $l = l+1$ e retorne ao passo 4.

Toward Estimating Autonomous Neural Network Based Electric Load Forecasters

Vitor Hugo Ferreira and Alexandre P. Alves da Silva, *Senior Member IEEE*

***Abstract*— Anticipation of load's future behavior is very important for decision making in power system operation and planning. During the last 40 years, many different load models have been proposed for short-term forecasting. After 1991, the literature on this subject has been dominated by neural network based proposals. This is mainly due to the neural networks' capacity for capturing the nonlinear relationship between load and exogenous variables. However, one major risk in using neural models is the possibility of excessive training data approximation, i.e., overfitting, which usually increases the out-of-sample forecasting errors. The extent of nonlinearity provided by neural network based load forecasters, which depends on the input space representation, has been adjusted using heuristic procedures. Training early stopping based on cross-validation, network pruning methods, and architecture selection based on trial and error are popular. The empirical nature of these procedures makes their application cumbersome and time consuming. This paper develops two nonparametric procedures for solving, in a coupled way, the problems of neural network structure and input selection for short-term load forecasting.**

***Index Terms*—** Load forecasting, feedforward neural networks, input selection, model complexity, Bayes procedures, support vector machines.

I. INTRODUCTION

Operational decisions in power systems, such as unit commitment, economic dispatch, automatic generation control, security assessment, maintenance scheduling, and energy commercialization depend on the future behavior of loads. Therefore, several short-term load forecasting methods have been proposed during the last four decades. Such a long experience in dealing with the load forecasting problem has revealed some useful models such as the ones based on multilinear regression, Box-Jenkins method, Artificial Neural Networks (ANNs) [1], fuzzy systems, and hybrid models. However, autonomous load forecasters, i.e., automatic input selection and model complexity control, are still needed to avoid expert intervention and to extend the application to the bus load level [2].

The relationship between electric load and its exogenous factors is complex and nonlinear, making it quite difficult to be modeled through conventional techniques such as linear time series and regression analyses. Classical methods are bias-prone, i.e., they are based on theoretical guesses about the underlying laws governing the system under study. On the other hand, after some years of practical experience, it has been recognized that ANNs can provide superior forecasting performance when dealing with nonlinear and multivariate problems involving large data sets, such as short-term load prediction. ANNs have more flexible functional forms in which there are few *a priori* assumptions about the relationships between input and output variables.

Although usually more robust than traditional load forecasting models, ANNs have overcome several problems in order to become commercially successful [3]. Since the first proposals of ANN based load forecasters [4], five major drawbacks have been tackled: heavy training burden, lack of prediction interval estimation, inference opacity, input space representation, and model complexity control.

Fast training algorithms have been developed since the early nineties [5], which have allowed the tracking of load non-stationarities. On the other hand, sometime has passed until the recognition of the practical importance of prediction interval estimation [6]. Qualitative interpretations of the ANN's forecasts have been proposed in references [7] and [8]. It seems that improvement on forecasting accuracy provided by ANNs cannot come without degrading model transparency. The ANN inference lack of interpretability can be mitigated using auxiliary tools such as the one described in [9]. However, it is hard to achieve a level of interpretability comparable to the one extractable from linear models.

The last two drawbacks are critical for short-term load forecasting, although they have not received much attention. The ANN input representation and complexity control should not be treated separately, as it is common practice in load forecasting. The extent of nonlinearity required from an ANN is strongly dependent on the selected input variables. One of the advantages of neural network models is the universal approximation capability, i.e., unlimited precision for continuous mapping. However, this theoretical advantage can backfire if data overfitting is not avoided [10]. The main objective of model complexity control is to match data regularity with model structure, maximizing the generalization capacity.

A popular procedure for ANN complexity control is based on cross-validation with training early stopping, i.e., the

This work was supported by the Brazilian Research Council (CNPq) and by the State of Rio de Janeiro Research Foundation (FAPERJ).

A.P. Alves da Silva is with COPPE/UFRJ, Electrical Engineering Graduate Program, Power Systems Laboratory, P.O. Box 68504, Rio de Janeiro, RJ, 21945-972, Brazil (e-mail: alex@coep.ufrj.br).

V.H. Ferreira is D.Sc. candidate at COPPE/UFRJ, Electrical Engineering Graduate Program, Power Systems Laboratory, P.O. Box 68504, Rio de Janeiro, RJ, 21945-972, Brazil (e-mail: vitor@vishnu.coep.ufrj.br).

iterative updating of the connection weights until the error for the validation subset stops decreasing. This procedure is very heuristic, because it is not easy to detect the right iteration for interrupting the training process. Besides, although cross-validation has been successfully applied to neural classifiers design, serial correlation information can be lost when it is used in time series forecasting. Shortcomings of cross-validation and early stopping are fully analyzed in [11], [12].

Input space representation is probably the most important subtask in load forecasting. It has been shown that input variable selection based on linear auto- and cross-correlation analyses is not appropriate for nonlinear models such as ANNs. Feature extraction via multiresolution analysis, based on wavelets, has been proposed to overcome this problem [13]. However, a more ANN oriented input selection scheme is still needed to capture the important information about the linear and nonlinear interdependencies in the associated multivariate data.

This paper develops two methods based on some of the most suitable techniques for controlling ANN complexity, with simultaneous selection of appropriate explanatory input variables for short-term load forecasting. In order to automatically minimize the out-of-sample prediction error, Bayesian training [14], [15] and Support Vector Machine (SVM) learning [16], [17] are investigated. These training methods include complexity control terms in their objective functions, which allow autonomous modeling and adaptation. An after-training complexity adjustment procedure, based on activation function gain scaling [18], is evaluated because of its simplicity.

Preliminary results on the applicability of Bayesian training and SVMs to short-term load forecasting have been reported in reference [19]. In the present paper, several open questions are answered. State-of-the-art nonparametric regression tools are extended in this work to fulfill the requirements of the problem of interest. In Bayesian training, assumptions of different priors for load and weather related input variables are considered. Specific learning parameters for each input are also employed in [20]. However, their estimation is performed by genetic algorithms with cross-validation based fitness function. Here, cross-validation is avoided in Bayesian and SVM training with the development of automatic analytical procedures for selecting among possible input variables and ANN structures.

The Bayesian approach has been fully exploited for the first time in load forecasting. A new procedure for determining useful inputs has been developed for avoiding the pre-determination of significance thresholds. For the first time, support vector regression learning parameters are estimated along with the kernel parameters without cross-validation, in contrast to recently proposed models [21]-[25].

Three databases have been used for testing. The first one corresponds to the load and temperature series, in hourly basis, from a North-American electric utility [4], [26], which has been used in load forecasting competitions. The second database is related to a daily peak load forecasting competition, with load and temperature data from the Eastern Slovakian Electricity Corporation [25]. The last dataset contains half-hourly loads, temperatures, and prices from the electricity market management company in Australia [27].

These internet based datasets have been employed to allow reproduction of the results presented in this paper. Considering the intended reader, the paper is written to make the theoretical parts (Sections II, III, and IV) as self contained as possible. Special emphasis is given to the important aspects in short-term load forecasting. Section V presents results and Section VI concludes with recommendations.

II. ANN Complexity Control

Neural network models commonly used in load forecasting have a feedforward structure with one hidden layer only (e.g., Multi-Layer Perceptrons (MLPs), Radial Basis Functions (RBFs)). In order to introduce the adopted nomenclature, this section describes the general structure of a feedforward ANN, with one hidden layer and one output neuron, under supervised learning.

Let $\underline{x} \in \Re^n$ be a vector representing input signals and $\underline{w} \in \Re^M$ the vector with the ANN connection weights, where $M = mn + 2m + 1$ and m is the number of neurons in the hidden layer. The biases of the hidden neurons sigmoidal activation functions are represented by b_k, $k = 1, 2, ..., m$, while b stands for the bias of the output neuron linear activation function. The final mapping is:

$$y = f(\underline{x}, \underline{w}) = \sum_{k=1}^{m} (w_k c_k) + b \tag{1}$$

where $c_k = \varphi\left(\sum_{i=1}^{n} (w_{ik} x_i) + b_k \right)$.

Given a dataset U with N input/output pairs, $U = \{X, D\}$, for $X = (\underline{x}_1, \underline{x}_2, ..., \underline{x}_N)$ and $D = (d_1, d_2, ..., d_N)$, where $d_j \in \Re$ represents the desired outputs, the ANN training objective usually is the estimation of the weight vector $\underline{w}$ such that the empirical risk (training error) is minimized, i.e.:

$$\min_{\underline{w}} \left\{ E_s(\underline{w}, U) = \frac{1}{2} \sum_{j=1}^{N} \left[d_j - f(\underline{x}_j, \underline{w}) \right]^2 \right\} \tag{2}$$

There are several algorithms for minimizing Eq. (2). Independently of using the classical error backpropagation, or second order methods, such as the Levenberg-Marquardt [13], or any other training method, the main drawback of this unconstrained training criterion is the absence of any concern regarding model complexity.

There are two basic approaches to control the ANN extent of nonlinearity. The first one is called structure stabilization, in which the objective is to determine the minimum number of neurons in the hidden layer. This approach can be implemented by comparing different structures using pruning or growing procedures [28], via cross-validation or analytical estimation of model complexity (e.g., VC - Vapnik Chervonenkis bounds [16] and NIC [29]).

Support vector machines (Section IV) belong to the structure stabilization approach. SVM learning is based on the minimization of the structural risk, i.e., the minimization of upper bounds on the generalization error (VC bounds), which hold with high confidence. Therefore, an SVM has its complexity implicitly controlled, with the model structure being a byproduct of training.

The second basic approach for controlling the neural network complexity is based on regularization theory, in which analytical methods adjust the ANN extent of nonlinearity without necessarily changing the model structure. Section III presents two methods related to this approach: gain scaling [18] and Bayesian training [14].

III. Regularization Techniques

Regularization theory shows how generalization behaves as the number of examples tends to infinity. A balance between training error and generalization capacity is obtained through the minimization of the total risk:

$$\min_{\underline{w}} \left\{ R\left(\underline{w}\right) = E_s\left(\underline{w},U\right) + \lambda E_c\left(\underline{w}\right) \right\} \tag{3}$$

In Eq. (3), $E_s\left(\underline{w},U\right)$ denotes the empirical risk, given by Eq. (2), while $E_c\left(\underline{w}\right)$ estimates the model complexity. The factor λ is known as the regularization parameter, which weights the bias-variance trade-off, i.e., training error versus model complexity. The setting of the regularization parameter λ is performed via resampling or by Bayesian estimation.

A. Gain Scaling Method

The activation function gain scaling method [18] is a post-training method equivalent to inserting noise in the training patterns (without doing that explicitly). The motivation for including corrupted versions of the original input patterns in the training set is to smooth the functional mapping, avoiding divergent outputs for similar inputs. Similar generalization capacity can be obtained with an ANN trained to minimize the empirical risk on the original dataset (i.e., without the corrupted patterns) if, after training, the gains (transition region slope) from the hidden neurons sigmoidal activation functions are multiplied by a_k, i.e.:

$$c_k = \varphi\left[a_k \left(\sum_{i=1}^{n} (w_{ik} x_i) + b_k \right) \right] \tag{4}$$

where

$$a_k = \frac{1}{\sqrt{\left\|\underline{w}_k\right\|^2 \sigma_{noise}^2 + 1}}$$

$$\underline{w}_k = \left[w_{1k}, w_{2k}, ..., w_{nk} \right]^t,\ k = 1, 2, ..., m.$$

The parameter $\sigma_{noise}^2 \in \Re^+$, associated with the variance of the noise "added" to the training input patterns, is estimated via cross-validation. For σ_{noise}^2 different from zero, the ANN model provides less nonlinearity because the slopes of the activation functions are decreased (increasing their linear segments). In this work, the ANNs trained by backpropagation have served for the gain scaling procedure as a starting point.

B. Bayesian ANN Training

One way to define the functional form of $\lambda E_c\left(\underline{w}\right)$, in Eq. (3), is through the application of Bayesian inference. Using Bayes' rule, the conditional probability density function (PDF) of $\underline{w}$, given a dataset U, $p\left(\underline{w}\middle| D, X\right)$, is estimated by:

$$p\left(\underline{w}\middle| D, X\right) = \frac{p\left(D\middle|\underline{w}, X\right) p\left(\underline{w}\middle| X\right)}{p\left(D\middle| X\right)} \tag{5}$$

Since X is conditioning all probabilities in Eq. (5), it will be omitted from this point on. Therefore, in Eq. (5), $p\left(D \mid \underline{w}\right)$ is the likelihood of D given $\underline{w}$, $p\left(\underline{w}\right)$ is $\underline{w}$'s *a priori* PDF, and $p\left(D\right) = \int p\left(D \mid \underline{w}\right) p\left(\underline{w}\right) d\underline{w}$ is enforcing $\int p\left(\underline{w} \mid D\right) d\underline{w} = 1$.

It is initially assumed that $\underline{w}$ presents a Gaussian distribution with zero mean and diagonal covariance matrix equal to $\alpha^{-1}\underline{\underline{I}}$, where $\underline{\underline{I}}$ is the $M \times M$ identity matrix, i.e.:

$$p\left(\underline{w}\right) = \frac{1}{Z_{\underline{w}}\left(\alpha\right)} e^{-\left(\frac{\alpha}{2}\left\|\underline{w}\right\|^2\right)},\ \text{where } Z_{\underline{w}}\left(\alpha\right) = \left(\frac{2\pi}{\alpha}\right)^{\frac{M}{2}} \tag{6}$$

The desired outputs can be represented by $d_j = f\left(\underline{x}_j, \underline{w}\right) + \zeta_j$, where ζ is Gaussian white noise with zero mean and variance equal to β^{-1}. The regularization factors α and β (learning parameters, also called hyperparameters), on the contrary of the other regularization techniques, are estimated along with the model parameters $\underline{w}$. Considering the previous hypotheses and assuming that the dataset patterns are independent, then:

$$p\left(D\middle|\underline{w}\right) = \frac{e^{\left\{-\frac{\beta}{2}\sum_{j=1}^{N}\left[d_j - f\left(\underline{x}_j, \underline{w}\right)\right]^2\right\}}}{Z_Y\left(\beta\right)},\ \text{where } Z_Y\left(\beta\right) = \left(\frac{2\pi}{\beta}\right)^{\frac{N}{2}} \tag{7}$$

Consequently, based on Eq. (5),

$$p\left(\underline{w}\middle| D\right) = \frac{e^{\left[-S\left(\underline{w}\right)\right]}}{\int e^{-S\left(\underline{w}\right)} d\underline{w}} \tag{8}$$

where

$$S\left(\underline{w}\right) = \frac{\beta}{2}\sum_{j=1}^{N}\left[d_j - f\left(\underline{x}_j, \underline{w}\right)\right]^2 + \frac{\alpha}{2}\sum_{l=1}^{M} w_l^2 \tag{9}$$

Therefore, the maximization of the *a posteriori* distribution of $\underline{w}$, $p\left(\underline{w} \mid D\right)$, is equivalent to the minimization of $S\left(\underline{w}\right)$ [14]. Dividing $S\left(\underline{w}\right)$ by β and making $\lambda = \alpha \div \beta$ in Eq. (3), the equivalence between $S\left(\underline{w}\right)$ and $R\left(\underline{w}\right)$ can be verified if:

$$E_c\left(\underline{w}\right) = \frac{1}{2}\left\|\underline{w}\right\|^2 \tag{10}$$

The regularization term in Eq. (10), known as *weight decay*, favors neural models with small magnitudes for the connection weights. Small values for the connection weights tend to propagate the input signals through the almost linear segment of the sigmoidal activation functions. Notice that the

requirement of prior information in Bayesian training is the primary instrument for controlling the ANN complexity.

One of the advantages of Bayesian training of an ANN is the embedded iterative mechanism for estimating λ, i.e., α and β, which avoids cross-validation. For multivariate problems such as load forecasting, the use of one single hyperparameter α for dealing with all connection weights is not recommended. Load and weather related input variables, such as temperature, require different priors. Even among the same type of variables, different levels of interdependency are involved (e.g., *P(k)* against *P(k+1)* and *P(K–23)* against *P(k+1)*, for an hourly basis load).

In this work, each group of connection weights directly related to an input variable receives a different α_i. The same idea is applied to the groups of weights associated with the biases (one α_i for the connections with the hidden neurons and another for the output neuron connection). One last α_i is associated with all connection weights between the hidden and output layers. Therefore, for *n* dimensional input vectors $\underline{x}$, the total number of α_i s is *n*+3.

1) Input Selection in Bayesian Training

For a given model structure, the magnitudes of the α_i s can be compared to determine the relevance of the corresponding input variables (taken from a pre-defined set). As $p(\underline{w}_i)$ is supposed to be normally distributed with zero mean and $\alpha_i^{-1}\underline{\underline{I}}$ covariance, then, the largest α_i s lead to the smallest $\underline{w}_i$ s. For estimating the *a posteriori* PDF of $\underline{w}$, Bayesian training combines the *a priori* PDF with the information provided by the training set (Eq. 5). If an α_i is large, the prior information about $\underline{w}_i$ is almost certain, and the effect of the training data on the estimation of $\underline{w}_i$ is negligible. Another way to see the influence of α_i on $\underline{w}_i$ is through Eq. (9).

The impact on the output caused by input variables with very small $\underline{w}_i$ s, i.e., very large α_i s, is not significant. However, a reference level for defining a very large α_i has to be established. For short-term load forecasting, two different references of irrelevance are needed: one reference for continuous variables, such as loads and temperatures, and another for dummy variables, such as hours of the day and days of the week. Uniformly distributed input variables can be employed to define the references of irrelevance [30]. For continuous input variables, an uniform random variable with lower and upper limits equal to $-\sqrt{3}$ and $\sqrt{3}$, respectively, is used as reference of irrelevance, since continuous variables have been standardized (zero mean and unit variance). For dummy variables, the reference is a binary random variable with uniform distribution. These two reference variables are added to the pre-defined set of inputs.

After training the model with the pre-defined set of input variables, continuous and dummy variables are separately ranked. For each rank, the variables with corresponding α_i s larger than α_{ref} (irrelevance level) are disregarded. After input selection, the ANN is retrained with the selected variables.

2) Structure Selection in Bayesian Training

Bayesian inference can also be employed to determine the best structure among a pre-defined set of possibilities, e.g., $\mathrm{H} = \{H_1, H_2, ..., H_K\}$, for which the corresponding inputs have been previously selected, i.e.,

$$P(H_h|D) = \frac{p(D|H_h)P(H_h)}{p(D)} \tag{11}$$

In Eq. (12), $p(H_h)$ represents the a priori probability of model H_h and $p(D|H_h)$ is given by:

$$p(D|H_h) = \iint p(D|\alpha,\beta,H_h)\,p(\alpha,\beta|H_h)\,d\alpha d\beta \tag{12}$$

Using Gaussian approximation around the estimated hyperparameters (from training), analytic integration of Eq. (13) is possible, leading to Eq. (13):

$$\ln p(D|H_h) = -S(\underline{w}) - \frac{1}{2}\ln\left|\nabla\nabla S(\underline{w})\right| + \frac{1}{2}\sum_{i=1}^{n+3} M_i\alpha_i + \frac{N}{2}\ln\beta + \ln(m!) + 2\ln m + \frac{1}{2}\sum_{i=1}^{n+3}\ln\left(\frac{2}{\gamma_i}\right) + \frac{1}{2}\ln\left(\frac{2}{N-\gamma}\right) \tag{13}$$

where *m* denotes the number of hidden neurons in the ANN model H_h. Since all models, *a priori*, are assumed equally probable, H_h is selected by maximizing $P(D|H_h)$, which is equivalent to maximizing $\ln p(D|H_h)$. Consequently, Eq. (13) can be used for ranking and selecting among MLPs with different numbers of neurons in the hidden layer.

3) Extended Bayesian Training Algorithm

The following steps describe the ANN structure and input selection via Bayesian inference.

Step 1. Set the minimum ($N_{\min}$) and maximum ($N_{\max}$) number of neurons in the hidden layer. In this work, $N_{\min} = 1$ and $N_{\max} = 10$.

Step 2. Make the number of neurons in the hidden layer $m = N_{\min}$.

Step 3. Add the reference of irrelevance variables to the user defined *n*-dimensional input vector. If dummy variables are used, the input set will contain $n = n+2$ input variables. Otherwise, i.e., if only continuous inputs are pre-selected, $n = n+1$.

Step 4. Set $l = 0$ and initialize $\underline{w}(l) = \left[\underline{w}_1(l), ..., \underline{w}_{n+3}(l)\right]^t$, $\underline{\alpha}(l) = \left[\alpha_1(l), ..., \alpha_{n+3}(l)\right]^t$, and $\beta(l)$.

Step 5. Minimize $S(\underline{w})$ on $\underline{w}(l)$ to obtain $\underline{w}(l+1)$.

Step 6. Calculate $\alpha_i(l+1)$, $\beta(l+1)$, and $\gamma_i(l+1)$ using the following equations:

$$\nabla\nabla S(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)} = \beta(l)\nabla\nabla E_s(\underline{w},U)\big|_{\underline{w}=\underline{w}(l+1)} + \alpha(l)\underline{\underline{I}}$$

$$\underline{\underline{B}}_i(l+1) = \left[\nabla\nabla S(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}\right]^{-1} \underline{\underline{I}}_i \tag{14}$$

$$\gamma_i(l+1) = M_i - trace\left\{\underline{\underline{B}}_i(l+1)\right\}$$

$$\alpha_i(l+1) = \frac{\gamma_i(l+1)}{\left\|\underline{w}_i(l+1)\right\|^2}$$

$$\beta(l+1) = \frac{N - \sum_{i=1}^{n+3}\gamma_i(l+1)}{\sum_{j=1}^{N}\left[d_j - f\left(\underline{x}_j, \underline{w}(l+1)\right)\right]^2}$$

Step 7. Make $l = l+1$ and return to Step 5 until convergence has been achieved. After convergence, go to the next step.

Step 8. Isolate in two lists the α_is associated with the continuous input variables and the α_js related to the dummy variables.

Step 9. For each list, select the inputs such that the corresponding $\alpha < \alpha_{ref}$, where α_{ref} stands for the hyperparameter associated with the added irrelevant input.

Step 10. Repeat Steps 4 to 7 using the inputs selected in Step 9, with n equal to the number of selected variables, to obtain the trained model H_m.

Step 11. Evaluate the log evidence of the hypothesis (ANN structure) H_m using Eq. (13).

Step 12. If $m = N_{\max}$, then go to Step 13. Else, $m = m+1$ and return to Step 3.

Step 13. Select the H_k with the largest log evidence.

In Eq. (14), $\underline{\underline{I}}_i$ is an $M \times M$ diagonal matrix with ones at the positions corresponding to the i^{th} group of weights and with zeros otherwise. M_i is the number of connection weights in each group. Details on how to calculate the Hessian $\nabla\nabla E_s(\underline{w}, U)$ can be found in [13].

IV. Support Vector Machines

In classification problems [31], maximum margin SVM classifiers are estimated to minimize the generalization error bounds. The training patterns that define the separation surface, based on which the maximum margin is obtained, are called support vectors. The other training patterns have no influence on the inference process.

In order to apply the same idea to regression problems, the concept of classification margin is adapted. A margin in regression means the amount by which the training and test accuracy can differ, i.e., different error functions are used for training and testing. During training, analogously to classification problems, an approximation error is not counted if it is inside a band of size ±ε (see Eq. 16). Any training point lying outside this band (support vectors) has its corresponding error taken into account.

As a linear machine on feature space, i.e., the space defined by a set of nonlinear basis functions $\underline{\phi}(\underline{x})$ that allows the model to produce nonlinear mappings on the original input space of $\underline{x}$, the SVM output is given by:

$$y = \sum_{j=0}^{m} W_j \phi_j(\underline{x}) = \underline{W}^t \underline{\phi}(\underline{x}) \tag{15}$$

where $\underline{\phi}(\underline{x}) = \left[1, \phi_1(\underline{x}), \ldots, \phi_m(\underline{x})\right]^t$ and $\underline{W} = \left[b, W_1, \ldots, W_m\right]^t$.

The following ε-insensitive cost function is adopted here:

$$L_\varepsilon(d, y) = \begin{cases} \left(|d-y| - \varepsilon\right)^2, \text{ for } |d-y| - \varepsilon \geq 0 \\ 0, \text{ otherwise} \end{cases} \tag{16}$$

SVMs that use Eq. (16) as the error function are called L2-SVMs [32], in contrast with previously proposed SVM load forecasters (L1-SVMs), which use an ε-insensitive linear loss function. L2-SVMs have been employed in this work because they lead to differentiable analytical bounds for the generalization error. Such bounds cannot be derived for L1-SVMs. Then, the SVM hyperparameters can be directly estimated through mathematical programming techniques, avoiding cross-validation.

In the following development, ε and c_0 are assumed to be known, i.e., defined by the user. This assumption will be removed later. The training objective of an SVM model is the following constrained minimization of the empirical risk:

$$\min_{\underline{W}} \left\{ E_s(\underline{W}, D) = \frac{1}{N}\sum_{i=1}^{N} L_\varepsilon(d_i, y_i) \right\} \tag{17}$$

subject to

$$\left\|\underline{W}\right\|^2 \leq c_0$$

where c_0 also affects the model complexity.

A. Support Vector Regression

The primal optimization problem formulated by Eq. (17) is transformed into its dual form, Eq. (18), to allow the incorporation of kernel functions, which avoid the requirement of knowing an appropriate $\underline{\phi}(\underline{x})$.

$$\max_{\underline{\alpha}, \underline{\alpha}'} \left\{ Q(\underline{\alpha}, \underline{\alpha}') = \sum_{i=1}^{N} d_i(\alpha_i - \alpha_i') - \varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i') \right.$$
$$\left. -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i')(\alpha_j - \alpha_j')\left[K(\underline{x}_i, \underline{x}_j) + \frac{\delta_{ij}}{C}\right] \right\} \tag{18}$$

subject to

$$\sum_{i=1}^{N}(\alpha_i - \alpha_i') = 0$$

$$\alpha_i \geq 0,\ \alpha_i' \geq 0,\ i = 1, 2, ..., N$$

In Eq. (18), $K(\underline{x}_i, \underline{x}_j) = \underline{\phi}^t(\underline{x}_i)\underline{\phi}(\underline{x}_j)$ is the inner product kernel defined according to Mercer's theorem [16], δ_{ij} is the Kronecker delta function, and C is the regularization hyperparameter . Then, the output of an SVM is given by:

$$y = f(\underline{x}, \underline{W}) = \sum_{i=1}^{N}(\alpha_i - \alpha_i')K(\underline{x}, \underline{x}_i) \tag{19}$$

As indicated in Eq. (19), the support vectors are the training patterns for which $\alpha_i \neq \alpha_i'$, i.e., the ones located outside the band defined by ε. In fact, an SVM model can be represented as a feedforward ANN model with hidden layer units activation functions defined by the kernel $K(\underline{x},\underline{x}_i)$. Notice that an SVM model, depending on the adopted kernel function, has the MLP and the RBF as special cases, when the kernels are specified as sigmoid and Gaussian functions, respectively. However, an important difference compared with traditional training algorithms for MLPs and RBFs is related to the convexity of the corresponding objective functions. While for error backpropagation and clustering algorithms local minima can be troublesome, in SVM training the solution is unique due to the corresponding quadratic optimization problem.

B. SVM Input Selection

Reference [33] develops the concept of *span of support vectors*, from which a differentiable upper bound on the generalization error for regression is derived in [32]:

$$T_{SB}\left[f(\underline{x},\underline{w})\right] = \sum_{i=1}^{p}(\alpha_i + \alpha_i')\tilde{S}_i^2 + N\varepsilon \tag{20}$$

where α_i, α_i' are the Lagrange multipliers associated with the support vector $\underline{x}_i$, p the number of support vectors, and

$$\tilde{S}_i^2 = \min_{\underline{\mu}} \left\| \underline{\tilde{\phi}}(\underline{x}_i) - \sum_{j=1, j\neq i}^{p} \mu_j \underline{\tilde{\phi}}(\underline{x}_j) \right\|^2 + \eta \sum_{j=1, j\neq i}^{p} \frac{\mu_j^2}{(\alpha_i + \alpha_i')} \tag{21}$$

subject to

$$\sum_{j=1, j\neq i}^{p} \mu_j = 1, \text{ for } \mu_j \in \Re$$

with η denoting a parameter responsible for promoting differentiability (η=0 turns the objective function in Eq. (21) to a non-differentiable one) and $\underline{\tilde{\phi}}(\underline{x}_j) = \left[\underline{\phi}(\underline{x}_j) \quad \underline{o}_j / \sqrt{C}\right]^t$ representing an extended feature space mapping, where $\underline{o}_j$ is an N dimensional vector with the j^{th} element equal to one and the other elements equal to zero.

The optimal solution for Eq. (21) is presented in [32], along with the partial derivatives of T_{SB} with respect to C, ε, and the kernel parameters. The minimization of Eq. (20), via gradient descent, is applied here to select inputs and the L2-SVM structure, which is determined not only by C, ε, and the kernel parameters, but also by the selected input variables. Therefore, extending the proposal in [32], the present work estimates the individual contributions of each input to the Gaussian kernel as a way to select input variables.

Input weights, σ_i s, for measuring the significance of each pre-selected input variable (i.e., input space is scaled by $\sigma_i x_i$) can be associated with the kernel parameters. This can be verified by writing the Gaussian kernels as follows:

$$K(\underline{x},\underline{y}) = e^{-\sum_{i=1}^{n}(\sigma_i x_i - \sigma_i y_i)^2} = e^{-\sum_{i=1}^{n}\sigma_i^2 (x_i - y_i)^2} \tag{22}$$

A small scaling factor means that the corresponding input is not relevant. Therefore, such an input is disregarded. Similarly to Bayesian training, a reference for a small σ_i is needed. Therefore, an analogous procedure is applied to rank the pre-defined inputs and disregard the less significant ones. Afterwards, the L2-SVM is retrained with the selected inputs. Notice that the standard SVM Gaussian kernel uses $\sigma_i = \sigma$ for all input variables.

Due to the nonconvex nature of T_{SB}, gradient descent depends on initialization, which is hard to set because the learning parameters optima values can be very different in magnitude. This is also troublesome for determining the gradients, because the sensitivity to parameters varying in small magnitude ranges is jeopardized. Logarithmic transformations can be used to overcome this problem. Regarding gradient descent initialization, reference [34] derives useful expressions (Eq. 23) for estimating C and ε, which are employed here to start the search. The σ_i values have been initialized at 0.1. The initial values for C and ε are:

$$C_{est} = \max\left(\left|\bar{d} + 3s_d\right|, \left|\bar{d} - 3s_d\right|\right) \tag{23}$$

$$\varepsilon_{est} = 3s\sqrt{\frac{\ln N}{N}}$$

where $s = \sqrt{\frac{1}{N-n}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2}$. In Eq. (23), $\bar{d}$ is the sample mean for the target values, s_d is the corresponding standard deviation, and s is the standard deviation of the regression model error. In this paper, s is estimated from the residues of an ARX (Auto Regressive Exogenous) linear model.

C. Automatic L2-SVM Learning

The proposed L2-SVM learning algorithm can be summarized as follows.

Step 1. Add the reference of irrelevance variables to the user defined set of inputs (as in the extended Bayesian training).

Step 2. Set $l = 0$ and initialize $C(l)$ and $\varepsilon(l)$, using Eq. (23). Initialize the scaling factors $\underline{\sigma}(l) = \left[\sigma_1(l),...,\sigma_n(l)\right]^t$. In this work, all scaling factors are initially equal to 0.1.

Step 3. Solve Eq. (18) to obtain $\underline{\alpha},\underline{\alpha}'$.

Step 4. Minimize $T_{SB}\left[f(\underline{x},\underline{w})\right]$ via gradient descent to get $C(l+1)$, $\varepsilon(l+1)$, and $\underline{\sigma}(l+1)$.

Step 5. Make $l = l+1$ and return to Step 3 until convergence has been achieved. After convergence, go to the next step.

Step 6. Isolate in two lists the σ_is associated with the continuous input variables and the σ_js related to the dummy variables.

Step 7. For each list, select the inputs such that the corresponding $\sigma > \sigma_{ref}$, where σ_{ref} denotes the hyperparameter associated with the added irrelevant input.

Step 8. Repeat Step 3 using the inputs selected in Step 7 and the previously optimized hyperparameters (Step 4) to obtain the final model.

V. Test Results

The three datasets are standardized. The first one, with hourly load and temperature values, available at *ee.washington.edu/class/555/el-sharkawi/index_files/Page3404.html*, contains data from January 1st, 1985 to March 31st, 1991. In this case, the task is to forecast the hourly load, from 16 up to 40 hours (steps) ahead for weekdays, and from 16 up to 80 hours ahead for weekends. The test (out-of-sample) period goes from November 1st, 1990 to March 31st, 1991. With training data from the month to be forecasted and from two months earlier, along with the data corresponding to the same "window" in the previous year, seven models are estimated, one for each day of the week. Around 650 patterns are used for each model.

As the initial set of inputs, the following variables are tested: 24 dummy variables codifying the hour of the day; lags *S(k–1)*, *S(k–2)*, ..., *S(k–6)*, *S(k–24)*, *S(k–25)*, ..., *S(k–29)*, *S(k–168)*, *S(k–169)*, ..., *S(k–173)* for load, temperature and temperature square series; the temperature forecast for hour *k* and its square value, i.e., $T(k)$ and $T^2(k)$, respectively; the daily maximum temperature forecast and its square value, $T_{max}(d)$ and $T_{max}^2(d)$; and the daily maximum temperature for the previous day and its square value, $T_{max}(d-1)$ and $T_{max}^2(d-1)$. Therefore, a total of 84 initial inputs (including dummies) have been presented to the models for selection. The output is the predicted hourly load *L(k)*. As weather services can provide quite precise forecasts for the horizons of interest, the true temperatures have been employed as "perfect" predictions. The forecasts up to 80 hours ahead are provided by recursion, i.e., load forecasts feed inputs. The number of pre-selected inputs has been deliberately made big. The idea is to verify the ability of the training algorithms in identifying the most significant variables. So far, the best results (benchmark) for this database are presented in [26].

For the second database, with daily peak load and temperature values from January 1st, 1997 to January 31st, 1999, available at *http://neuron.tuke.sk/competition*, the out-of-sample period for 31-step ahead predictions of daily peak load goes from January 1st, 1999 up to January 31st, 1999. To avoid recursion, 31 models are estimated, one for each step ahead, using all data until January 1st, 1999 (≈720 patterns per ANN). For the j^{th} model, the initial inputs are related to the seven most recent daily peak load values, plus j+7 lagged temperature variables, and 19 dummy variables, seven for the days of the week and twelve for the months. Therefore, a total of 33+j initial inputs (including dummies) have been presented to each model for selection. The lags for the load and temperature variables are *L(d–j)*, *L(d–j–1)*, ..., *L(d–(6+j))* and *T(d)*, *T(d–1)*, ..., *T(d–(6+j))*, respectively. The model output is the daily peak load *L(d)*. As before, the true temperatures for the forecasting horizon are used as "predictions". The benchmark results for this database are presented in [25].

For the last database, at *www.nemmco.com.au*, with half-hourly load, price, and temperature values from December 4th, 2001 to December 31st, 2003, the task is to forecast the hourly loads, from 1 up to 6 hours ahead for several weeks in 2003. The series are transformed to an hourly basis by averaging two half-hours. For any week to be forecasted, the corresponding training sets are built as for the first database (≈530 patterns per ANN). Six models are developed, one for each number of steps ahead, for each day of the week. The models for j steps ahead have the following pre-defined inputs: 19–j lagged load, price and temperature variables, plus j temperature forecasts (i.e., *T(k)*, *T(k–1)*, …, *T(k–j+1)*), and 24 dummy variables codifying the hour of the day, totalizing 81–2j inputs. The pre-selected lags are *S(k–j)*, *S(k–j–1)*, *..., S(k–6)*, *S(k–24)*, *S(k–25)*, ..., *S(k–29)*, *S(k–168)*, *S(k–169)*, ..., and *S(k–173)* for load, price and temperature. The output is the hourly load *L(k)*. The benchmark results for this database are presented in [27].

The previous datasets specifications have been applied to all training methods. Test results have been generated for the following training methods: conventional error back-propagation (minimization of the empirical error only); BackPropagation (BP) followed by gain scaling; BP with gain scaling and Structure Selection (SS) via Cross-Validation (CV); BP with early stopping for regularization of a invariable structure; BP with early stopping for regularization with SS; the extended Bayesian training; L2-SVM learning with parameters estimated via CV; and the proposed L2-SVM learning. Furthermore, Correlation based Input Selection (CIS) has been tested in combination with gain scaling, early stopping, and L2-SVM with CV. Correlation analysis is used not just for selecting significant linear dependencies between possible inputs and output. It is also employed to eliminate redundant input variables. All dummy variables are preserved when CIS is applied. Different from the proposed methods, CIS does not depend on the ANN model.

Table I presents the Mean Absolute Percentage Errors (MAPE) from the training methods. Its last line shows the performance improvements between the best models and the benchmarks. The extended Bayesian methodology produces superior results for all cases, except for case 1, in which it has been overcome by a small margin. Although exhibiting good results, L2-SVM with gradient descent has not been competitive for case 1. This method has defeated its counterpart based on CV for half of the test cases. However, execution time of L2-SVM with gradient descent is, in average, smaller than the one of L2-SVM with CV.

TABLE I
COMPARISON AMONG DIFFERENT MODELS (MAPE)

	Case 1	Case 2	Case 3					
			1 step ahead	2 step ahead	3 step ahead	4 step ahead	5 step ahead	6 step ahead
Backpropagation	10.43	5.05	0.97	1.33	1.49	1.57	1.80	1.72
Gain Scaling	14.18	4.87	1.53	1.60	1.86	1.97	2.09	2.44
Gain Scaling with SS	13.76	2.19	1.53	1.68	1.94	1.81	2.26	2.50
Gain Scaling with CIS and SS	17.80	2.77	2.41	3.58	3.51	3.03	3.24	3.38
Early Stopping	8.07	1.95	2.04	1.93	2.09	2.60	2.00	2.35
Early Stopping with SS	7.11	2.13	1.61	1.44	1.49	1.57	1.78	1.46
Early Stopping with CIS and SS	11.41	2.87	2.14	2.26	2.39	2.27	2.27	2.22
Extended Bayesian Training	4.89	1.75	0.49	0.72	0.82	0.94	0.99	1.07
L2-SVM with CV	4.88	3.52	0.81	0.93	1.06	1.15	1.20	1.36
L2-SVM with CIS and CV	10.54	2.87	1.57	2.15	2.15	2.15	2.24	2.24
L2-SVM Gradient Descent	8.72	2.07	0.88	0.84	1.01	1.20	1.56	1.20
Benchmark	4.73	1.98	0.56	0.83	1.00	1.15	1.20	1.30
Improvement (%)	-3.09	11.72	11.73	13.40	18.17	17.99	17.65	17.62

Table II indicates the computational burden on a 3 GHz/32 bits PC, using MATLAB interpreted code. The 32 minutes for the Extended Bayesian Training correspond to forecasting a

full day load curve in an hourly basis, which is compatible with practical requirements. The source codes associated with the proposed training algorithms have been based on [35] and [36]. In L2-SVM with CV, the number of hyperparameters has been decreased by making $\sigma_i = \sigma$, otherwise CV is not viable. Therefore, L2-SVM with CV has used the full set of pre-defined inputs and the input set determined by CIS (L2-SVM with CIS and CV). Notice that early stopping variations do not compete in accuracy with the extended Bayesian training, in which the datasets have been fully exploited. Furthermore, on the contrary of the benchmarks, the two leading proposals have their input spaces automatically selected. The activation function gain scaling procedure has not exhibited good results.

TABLE II
PROCESSING TIME (MINUTES)

	Case 1	Case 2	Case 3
Backpropagation	3.90	3.67	3.37
Gain Scaling	3.45	2.80	1.16
Gain Scaling with SS	23.99	20.94	16.29
Gain Scaling with CIS and SS	20.73	19.14	16.54
Early Stopping	0.03	0.02	0.01
Early Stopping with SS	24.04	9.65	0.04
Early Stopping with CIS and SS	7.00	4.54	0.03
Extended Bayesian Training	32.30	5.80	28.15
L2-SVM with CV	36.57	13.76	19.57
L2-SVM with CIS and CV	30.00	13.57	20.53
L2-SVM Gradient Descent	8.73	29.08	13.84

Table III presents the maximum absolute errors. Maximum error units have been chosen according to the benchmarks. Again, the extended Bayesian training exhibits the best overall performance. In Table IV, the average numbers of inputs selected by each model are presented. This table indicates the capacity of the leading methodologies to reduce the input dimensionality, improving the models' generalization ability. For example, in case 3, for 6-step ahead forecasts, *T(k–168), P(k–6), P(k–26), P(k–29), P(k–168), P(k–171), D(4), D(6), D(10), D(11),* and *D(16)* have been disregarded by the Bayesian method, where *P(·)* and *D(·)* stand for price and dummy variables, respectively. For the same case, correlation based input selection has saved *L(k–24)* and *L(k–168)*, only. Input variables related to temperature and price have been disregarded by CIS due to their strong nonlinear relationship with load. In Table V, the average numbers of neurons in the hidden layer of the MLPs and the average numbers of support vectors are presented.

TABLE III
COMPARISON AMONG DIFFERENT MODELS (MAXIMUM ERROR)

	Case 1 (%)	Case 2 (MW)	Case 3 (%)					
			1 step ahead	2 step ahead	3 step ahead	4 step ahead	5 step ahead	6 step ahead
Backpropagation	93.12	118.89	4.99	5.96	4.50	5.85	6.73	8.02
Gain Scaling	66.54	137.78	7.61	10.66	9.10	11.22	10.82	11.48
Gain Scaling with SS	87.50	55.95	6.89	9.16	14.87	7.46	11.88	10.21
Gain Scaling with CIS and SS	112.89	70.99	11.61	20.48	23.85	11.36	12.77	11.22
Early Stopping	43.98	40.28	7.84	10.38	7.91	15.13	7.02	12.95
Early Stopping with SS	46.07	50.90	5.97	6.79	5.56	6.18	7.16	5.73
Early Stopping with CIS and SS	54.03	71.26	7.32	9.43	8.66	8.34	8.52	8.72
Extended Bayesian Training	41.57	55.64	1.97	2.65	3.89	4.62	4.86	5.46
L2-SVM with CV	38.06	60.39	4.00	3.51	4.53	4.62	5.45	5.95
L2-SVM with CIS and CV	60.06	67.17	5.90	6.19	6.18	6.17	6.48	6.48
L2-SVM Gradient Descent	46.70	59.78	3.48	4.05	5.12	5.87	6.14	5.59
Benchmark	-	51.42	3.24	3.43	4.11	3.87	5.57	5.20
Improvement (%)	-	21.66	39.09	22.64	5.44	-19.26	12.69	-4.93

Figures 1 to 3 show some forecasts for the three databases. Fig. 1 presents one example for the first database (case 1). Fig. 2 shows predictions for the second database (case 2). Figure 3 presents forecasts for six steps ahead in case 3.

II. Conclusion

This paper has extended Bayesian and SVM learning techniques to propose autonomous neural network based short-term load forecasters. The proposed methodologies are fully

TABLE IV
AVERAGE NUMBERS OF INPUTS

	Case 1	Case 2	Case 3					
			1 step ahead	2 step ahead	3 step ahead	4 step ahead	5 step ahead	6 step ahead
Backpropagation	84	49	79	77	75	73	71	69
Gain Scaling	84	49	79	77	75	73	71	69
Gain Scaling with SS	84	49	79	77	75	73	71	69
Gain Scaling with CIS and SS	26	20	27	26	26	26	26	26
Early Stopping	84	49	79	77	75	73	71	69
Early Stopping with SS	84	49	79	77	75	73	71	69
Early Stopping with CIS and SS	26	20	27	26	26	26	26	26
Extended Bayesian Training	70	40	66	67	63	51	60	56
L2-SVM with CV	84	49	79	77	75	73	71	69
L2-SVM with CIS and CV	26	20	27	26	26	26	26	26
L2-SVM Gradient Descent	76	45	73	71	71	61	60	65
Reduction (%)	68.55	58.99	66.37	66.79	65.90	64.97	63.78	62.73

TABLE V
AVERAGE NUMBERS OF NEURONS AND SUPPORT VECTORS

	Case 1	Case 2	Case 3					
			1 step ahead	2 step ahead	3 step ahead	4 step ahead	5 step ahead	6 step ahead
Backpropagation	10	10	10	10	10	10	10	10
Gain Scaling	10	10	10	10	10	10	10	10
Gain Scaling with SS	8	1	7	9	8	9	9	9
Gain Scaling with CIS and SS	6	2	4	6	6	5	6	6
Early Stopping	10	10	10	10	10	10	10	10
Early Stopping with SS	8	8	8	8	8	8	8	8
Early Stopping with CIS and SS	8	6	7	5	7	8	7	6
Extended Bayesian Training	8	7	7	8	7	5	5	3
L2-SVM with CV	428	464	344	338	328	330	333	347
L2-SVM with CIS and CV	425	464	347	348	346	344	342	341
L2-SVM Gradient Descent	642	707	518	515	509	513	510	505

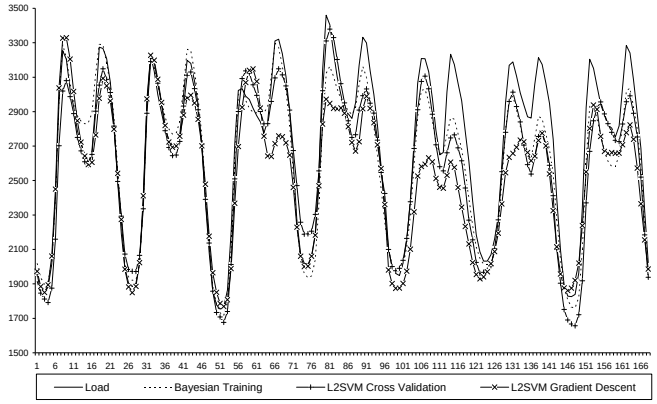

Fig. 1 Forecasts from 11/27/1990 to 12/03/1990, case 1.

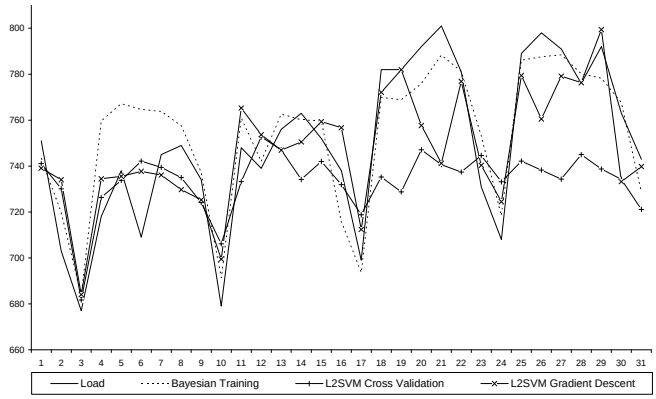

Fig. 2. Forecasts from 1/1/1999 to 1/31/1999, case 2.

data-driven, providing accurate forecasts with very little information from the user. Although requesting a heavy computational burden, they seem to be the answer for dealing with the large-scale bus load forecasting problem, in which the particular dynamics of each load series does not allow

manually tuned solutions.

Comparing the practical aspects of the hyperparameters estimation, without cross-validation, in Bayesian and L2-SVM training, the following facts have been observed. First of all, hyperparameter estimation cannot be performed in L2-SVM learning without an auxiliary procedure. Furthermore, it is easier to get good estimations for the Bayesian training hyperparameters, based on Gaussian priors, than to optimize the L2-SVM learning parameters using gradient descent. Gaussian priors seem to be robust for different load series. On the other hand, the gradient descent algorithm usually requires many iterations (with one L2-SVM training per iteration), and its convergence is strongly dependent on the stepsize control.

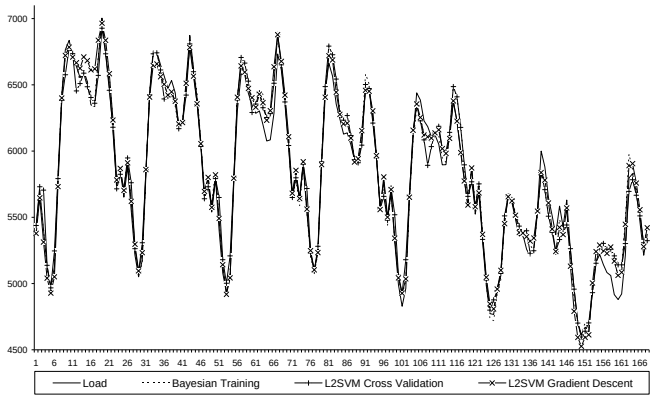

Fig. 3. Forecasts from 9/1/2003 to 9/07/2003, 6 step ahead, case 3.

Bayesian inference has been applied for clustering load dynamics to feed different SVM load forecasting models [24]. However, the application of Bayesian inference to the estimation of SVM learning parameters looks more promising. There is already some research effort on this idea [37], and it is worthwhile to pursue this direction for the next generation of short-term load forecasting tools.

III. REFERENCES

[1] H.S. Hippert, R.C. Souza, and C.E. Pedreira, "Neural Networks for Load Forecasting: A Review and Evaluation", *IEEE Trans. on Power Systems*, v.16, n.1, pp. 44-55, Feb. 2001.

[2] N. Amjady, "Short-Term Bus Load Forecasting of Power Systems by a New Hybrid Method", *IEEE Trans. on Power Systems*, v.22, n.1, pp. 333-341, Feb. 2007.

[3] A. Khotanzad, R. Afkhami-Rohani, and D. Maratukulam, "ANNSTLF – Artificial Neural Network Short-Term Load Forecaster – Generation Three", *IEEE Trans. on Power Systems*, v.13, n.4, pp. 1413-1422, Nov. 1998.

[4] D.C. Park, M.A. El-Sharkawi, and R.J. Marks II, "An Adaptively Trained Neural Network", *IEEE Trans. on Neural Networks*, v.2, n.3, pp. 334-345, May 1991.

[5] A.P. Alves da Silva, V.H. Quintana, and G.K.H. Pang, "Neural Networks for Topology Determination of Power Systems", Proc. First International Forum on Applications of Neural Networks to Power Systems, Seattle, USA, pp. 297-301, Jul. 1991.

[6] A.P. Alves da Silva and L.S. Moulin, "Confidence Intervals for Neural Network Based Short-Term Load Forecasting", *IEEE Trans. on Power Systems*, v.15, n.4, pp. 1191-1196, Nov. 2000.

[7] A.G. Bakirtzis, J.B. Theocharis, S.J. Kiartzis, and K.J. Satsios, "Short Term Load Forecasting Using Fuzzy Neural Networks", *IEEE Trans. on Power Systems,* v.10, n. 3, pp. 1518-1524, Aug. 1995.

[8] T. Matsui, T. Iizaka, and Y. Fukuyama, "A Novel Daily Peak Load Forecasting Method Using Analyzable Structured Neural Network", IEEE PES Winter Meeting, Columbus, USA, pp. 405-410, Jan. 2001.

[9] A.P. Alves da Silva, "Overcoming Limitations of NNs for On-Line DSA", IEEE PES General Meeting, San Francisco, USA, Jun. 2005.

[10] H.S. Hippert, D.W. Bunn, and R.C. Souza, "Large Neural Networks for Electricity Load Forecasting: Are They Overfitted?", *International Journal of Forecasting*, v.21, n.3, pp. 425-434, Jul. 2005.

[11] S. Amari, N. Murata, K.R. Müller, M. Finke, and H. Yang, "Statistical Theory of Overtraining – Is Cross-Validation Asymptotically Effective?", *Advances in Neural Information Processing Systems 8*, MIT Press, pp. 176-182, 1996.

[12] Z. Cataltepe, Y.S. Abu-Mostafa, and M. Magdon-Ismail, "No Free Lunch for Early Stopping", *Neural Computation*, v.11, n.4, pp. 995-1009, May 1999.

[13] A.J.R. Reis and A.P. Alves da Silva, "Feature Extraction Via Multi-Resolution Analysis for Short-Term Load Forecasting", *IEEE Trans. on Power Systems,* v.20, n.1, pp. 189-198, February 2005.

[14] D.J.C. Mackay, *Bayesian Methods for Adaptive Models*, Ph.D. Dissertation, California Institute of Technology, Pasadena, USA, 1992.

[15] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[16] V.N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

[17] B.Schölkopf and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, 2002.

[18] R. Reed., R.J. Marks II, and S. Oh, "Similarities of Error Regularization, Sigmoid Gain Scaling, Target Smoothing and Training with Jitter", *IEEE Trans. on Neural Networks*, v.6, n.3, pp. 529-538, May 1995.

[19] V.H. Ferreira and A.P. Alves da Silva, "Complexity Control of Neural Models for Load Forecasting", Proc. International Conference on Intelligent System Application to Power Systems", Washington D.C., USA, pp. 100-104, Nov. 2005.

[20] Z.S.H. Chan, H.W. Ngan, A.B. Rad, A.K. David, and N. Kasabov, "Short-Term ANN Load Forecasting from Limited Data Using Generalization Learning Strategies", *Neurocomputing*, v.70, n.1-3, pp. 409-419, Dec. 2006.

[21] P.F. Pai and W.C. Hong, "Forecasting Regional Electricity Load Based on Recurrent Support Vector Machines with Genetic Algorithms", *Electric Power Systems Research,* v.74, n.3, pp. 417-425, Jun. 2005.

[22] S. Fan and L.N. Chen, "Short-Term Load Forecasting Based on an Adaptive Hybrid Method", *IEEE Trans. on Power Systems*, v.21, n.1, pp. 392-401, Feb. 2006.

[23] J.F. Yang and J. Stenzel, "Short-Term Load Forecasting with Increment Regression Tree", *Electric Power Systems Research*, v.76, n.9-10, pp. 880-888, Jun. 2006.

[24] S. Fan, C.X. Mao, J.D. Zhang, and L.N. Chen, "Forecasting Electricity Demand by Hybrid Machine Learning Model", *Lecture Notes in Computer Science*, v.4233, pp. 952-963, Oct. 2006.

[25] B.-J. Chen, M.-W. Chang, and C.-J. Lin, "Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001", *IEEE Trans. on Power Systems*, v.19, n.4, pp. 1821-1830, Nov. 2004.

[26] R. Ramanathan, R. Engle, C.W.J. Granger, F. Vahid-Araghi, and C. Brace, "Short-Run Forecasts of Electricity Loads and Peaks", *International Journal of Forecasting*, v.13, n.2, pp. 161-174, Jun. 1997.

[27] P. Mandal, T. Senjyu, and T. Funabashi, "Neural Networks Approach to Forecast Several Hour Ahead Electricity Prices and Loads in Deregulated Market", *Energy Conversion and Management*, v.47, n.15-16, pp. 2128-2142, Sep. 2006.

[28] N.K. Treadgold and T.D. Gedeon, "Exploring Constructive Cascade Networks", *IEEE Trans. on Neural Networks*, v.10, n.6, pp. 1335-1350, Nov. 1999.

[29] N. Murata, S. Yoshizawa, and S.I. Amari, "Network Information Criterion – Determining the Number of Hidden Units for an Artificial Neural Network Model", *IEEE Trans. on Neural Networks*, v.5, n.6, pp. 865-872, Nov. 1994.

[30] H. Stoppiglia, G. Dreyfus, R. Dubois, and Y. Oussar, "Ranking a Random Feature for Variable and Feature Selection", *Journal of Machine Learning Research*, v.3, pp. 1399-1414, Mar. 2003.

[31] L.S. Moulin, A.P. Alves da Silva, M.A. El-Sharkawi, and R.J. Marks II, "Support Vector Machines for Transient Stability Analysis of Large-Scale Power Systems", *IEEE Trans. on Power Systems*, v.19, n.2, pp. 818-825, May 2004.

[32] M.-W. Chang and C.-J. Lin, "Leave-One-Out Bounds for Support Vector Regression Model Selection", *Neural Computation*, v.17, n.5, pp. 1188-1222, May 2005.

[33] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing Multiple Parameters for Support Vector Machines", *Machine Learning*, v.46, pp. 131-159, Jan. 2002.

[34] V. Cherkassky and Y. Ma, "Practical Selection of SVM Parameters and Noise Estimation for SVM Regression", *Neural Networks*, v.17, n.1, pp. 113-126, Jan. 2004.
[35] I.T. Nabney, *NETLAB: Algorithms for Pattern Recognition*, Springer-Verlag, 2002.
[36] C.-C. Chang and C.-J. Lin, *LIBSVM: A Library for Support Vector Machines*, 2001 (available at *http://www.csie.ntu.edu.tw/~cjlin/libsvm*)
[37] C. Wei, S.S. Keerthi, and J.O. Chong, "Bayesian Support Vector Regression Using a Unified Loss Function", *IEEE Trans. on Neural Networks*, v.15, n.1, pp. 29-44, Jan. 2004.

Vitor Hugo Ferreira received the B.Sc and M.Sc. degrees in Electrical Engineering from the Federal University of Itajubá, in 2002, and the Federal University of Rio de Janeiro, in 2005, respectively, both in Brazil. Currently, Mr. Ferreira is pursuing his Ph.D. degree at the Electrical Engineering Graduate Program, Federal University of Rio de Janeiro (COPPE/UFRJ). His research interests include time series forecasting and neural networks.

Alexandre P. Alves da Silva received the B.Sc, M.Sc. and Ph.D. degrees in Electrical Engineering from the Catholic University of Rio de Janeiro, in 1984 and 1987, and the University of Waterloo, Canada, in 1992, respectively. During 1999, he was a Visiting Professor in the Department of Electrical Engineering, University of Washington, USA. Currently, he is a Professor in Electrical Engineering at COPPE/UFRJ. He has authored and co-authored 200 papers on intelligent systems application to power systems.