UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Elétrica e de Computação

Comitê de Máquinas em Predição de Séries Temporais

Wilfredo Jaime Puma Villanueva

Orientador: Prof. Dr. Fernando José Von Zuben
Co-orientador: Prof. Dr. Clodoaldo Aparecido de Moraes Lima

Tese apresentada à Pós-graduação da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como requisito parcial à obtenção do Título de **Mestre em Engenharia Elétrica** na área de Engenharia de Computação.

Banca Examinadora:

Prof. Dr. Marinho Gomes de Andrade Filho – ICMC/USP
Prof. Dr. Basílio Ernesto de A. Milani – DT/FEEC/UNICAMP
Prof. Dr. Gilmar Barreto – DMCSI/FEEC/UNICAMP

Campinas, outubro de 2006

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Puma Villanueva, Wilfredo Jaime

P968c Comitê de máquinas em predição de séries temporais / Wilfredo Jaime Puma Villanueva . --Campinas, SP: [s.n.], 2006.

Orientadores: Fernando José Von Zuben, Clodoaldo Aparecido de Moraes Lima.

Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Redes neurais. 2. Análise de séries temporais - Processamento de dados. 3. Aprendizado de computador. I. Von Zuben, Fernando José. II. Lima, Clodoaldo Aparecido de Moraes. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Título em Inglês: Committee machines in time series prediction

Palavras-chave em Inglês: Neural networks (Computer science), Time-series analysis Data processing, Machine learning

Área de concentração: Engenharia de Computação

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Marinho Gomes de Andrade Filho, Basílio Ernesto de Almeida Milani, Gilmar Barreto

Data da defesa: 11/10/2006

Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA – TESE DE MESTRADO

Candidato: Wilfredo Jaime Puma Villanueva

Data da defesa: 11 de outubro de 2006

Título da Tese: "Comitê de Máquinas em Predição de Séries Temporais"

Prof. Dr. Fernando José Von Zuben (26395-8): _____

Prof. Dr. Marinho Gomes de Andrade Filho: _____

Prof. Dr. Basílio Ernesto de Almeida Milane : _____

Prof. Dr. Gilmar Barreto: _____

Secretária Soraia Cecília Montagner: _____

Coordenador PG Prof. Dr.Michel Daoud Yacoub : _____

Resumo

A capacidade de aproximação universal apresentada por redes neurais artificiais foi explorada nos últimos anos junto a problemas de classificação e regressão de dados, envolvendo técnicas de treinamento supervisionado. No entanto, as redes neurais resultantes podem produzir queda de desempenho frente a amostras de teste. Esta é a principal motivação para o emprego de comitês de máquinas, na forma de um *ensemble* ou uma mistura de especialistas. Um *ensemble* toma propostas de solução completas para um problema e se ocupa em selecionar e combinar essas propostas na obtenção de uma única resposta. Já numa mistura de especialistas, cada especialista é responsável por parte do problema e os especialistas, assim como o módulo que decide qual especialista irá atuar em cada caso, são sintetizados simultaneamente. A aplicação de comitês de máquinas em predição de séries temporais indica que esta estratégia pode conduzir a ganhos de desempenho, quando comparado ao uso de um único preditor e considerando vários casos de estudo. Ainda no contexto de predição, foram investigadas duas técnicas para seleção de variáveis, além de ser avaliado o desempenho de duas propostas de partição da série temporal em conjuntos de treinamento, validação e teste. Os resultados de teste de significância do ganho de desempenho permitem apontar uma técnica de seleção e uma proposta de partição como as mais indicadas.

Abstract

The universal approximation capability presented by artificial neural networks has been explored in recent years to solve classification and regression problems, using the supervised learning framework. However, the resulting neural networks may present degradation of performance when the test dataset is considered. This is the main motivation for the use of committee machines, in the form of an ensemble or a mixture of experts. An ensemble takes full-solution proposals and tries to select and combine them toward a single response. In the realm of a mixture of experts, each expert is devoted to a parcel of the original problem, and the experts, together with the module that allocates the individual role for each expert, are synthesized simultaneously. The application of committee machines to time series prediction indicates that these machine learning strategies can promote improvement in performance, when compared to the use of a single predictor and taking several case studies. Still in the context of prediction, two techniques for variable selection have been investigated, and two proposals for the partition of the time series in training, validation, and test datasets have been compared. The results in terms of test of significance of the gain in performance clearly indicate the superiority of one of the selection techniques and one of the partition proposals.

Dedicatória

A meus pais:
Brígida Justina e Demetrio Flavio
e irmãos:
Ronald Flavio e Luis Alfredo

Agradecimentos

A meus pais, por ser fruto do amor e do cuidado deles, por me brindarem as condições necessárias de crescer. A meus irmãos e familiares por me terem presente sempre nos seus pensamentos.

Ao meu orientador e amigo, o Prof. Fernando José Von Zuben, por ter confiado e apostado em mim e em meu trabalho, por ser o guia neste caminho percorrido e pelo exemplo de ser uma pessoa comprometida com o trabalho.

Ao meu co-orientador, o Prof. Clodoaldo A. de Moraes Lima, pelas discussões e ensinamentos ao longo de todo o desenvolvimento do trabalho.

Ao meu colega e amigo, Prof. Eurípedes P. dos Santos, principalmente, por ter cuidado das minhas leituras. Obrigado “Maestro”.

A todos os meus colegas do LBiC e do LCA, pela amizade, respeito e pelo apoio nos momentos difíceis.

À Jeanne Dobgenski, por ter me indicado o caminho de como chegar ao DCA-FEEC.

Aos professores da pós-graduação, pelo excelente trabalho que realizam.

Ao pessoal do setor administrativo, por estarem sempre dispostos a atender nossas solicitações da melhor forma.

À CAPES e ao CNPq, pelo apoio financeiro.

Em geral à FEEC-Unicamp, às pessoas que a compõem, desde a diretoria e coordenação até o pessoal de serviços, por cuidarem das melhores condições para que os alunos desenvolvam seus projetos de pesquisa.

Conteúdo

Resumo v

Abstract vii

Dedicatória ix

Agradecimentos xi

Conteúdo xiii

Lista de Figuras xix

Lista de Tabelas xxiii

Lista de Abreviações xxv

Lista de Símbolos xxvii

Capítulo 1 – Introdução

1.1 Escopo da dissertação 1

1.1.1 Séries temporais 1

1.1.2 Redes neurais artificiais 3

1.1.3 Comitê de máquinas 4

1.1.4 Seleção de variáveis 5

1.2 Motivação da proposta 5

1.3 Objetivos 7

1.4 Metodologia 8

1.5 Contribuições 8

1.6 Organização e conteúdo da dissertação 9

Capítulo 2 – Redes neurais artificiais em predição de séries temporais

2.1 Introdução às redes neurais artificiais 11

2.1.1 Taxonomia 12

2.1.1.1 Tipo de associação entre as informações de entrada e saída 13

2.1.1.2 Tipo de arquitetura 13

2.1.1.3 Tipo de mecanismo de aprendizagem 14

2.1.1.4 Tipo de procedimento de ajuste das conexões sinápticas 16

2.1.2 Motivações para o emprego de redes neurais artificiais 16

2.1.3 Principais aplicações 17

2.1.3.1 Problemas de reconhecimento de padrões 17

2.1.3.2 Problemas de regressão de dados 18

2.1.3.3 Problemas de otimização combinatória 19

2.1.3.4 Outras aplicações 19

2.1.4 Descrição do perceptron de múltiplas camadas – MLP 19

2.1.4.1 Descrição do algoritmo de treinamento 21

2.2 Introdução às séries temporais 24

2.2.1 Principais características das séries temporais 26

2.2.1.1 Tendência 26

2.2.1.2 Sazonalidade 27

2.2.1.3 Observações Aberrantes 28

2.2.1.4 Heterocedasticidade 29

2.2.1.5 Não-linearidade 29

2.2.2 Modelos básicos para predição de séries temporais 30

2.2.2.1 Modelo auto-regressivo – AR(p) 31

2.2.2.2 Modelo de médias móveis – MA(q) 31

2.2.2.3 Modelo ARMA(*p*,*q*) 32
2.2.2.4 Modelo ARIMA(*p*,*d*,*q*) 32
2.3 Predição de séries temporais via RNAs 33
2.3.1 Histórico do uso de RNAs em predição de séries temporais 34
2.3.2 Tratamento dos dados para o treinamento 38
2.3.2.1 Acondicionamento 38
2.3.2.2 Construção dos conjuntos de treinamento 38
2.3.3 Simulações e resultados 40
2.3.3.1 Configuração dos parâmetros 41
2.3.3.2 Resultados obtidos 42

Capítulo 3 – Comitê de máquinas

3.1 Introdução 45
3.1.1 Razões que levaram ao surgimento de comitê de máquinas 47
3.1.2 Estruturas: Estática e Dinâmica 48
3.2 *Ensemble* 49
3.2.1 Motivações para usar *ensemble* 51
3.2.2 Desempenho esperado para um *ensemble* 53
3.2.3 Um breve histórico dos *ensembles* 58
3.3 Etapas no projeto de um *ensemble* 59
3.3.1 Etapa de geração de componentes 59
3.3.1.1 Geração via *bagging* 61
3.3.1.2 Variações do *bagging* 62
3.3.1.3 Geração via *boosting* 63
3.3.1.4 Geração *Adaboost* 64
3.3.2 Etapa de combinação de componentes 64
3.3.3 Etapa de seleção de componentes 66

3.3.3.1 Método construtivo 67
3.3.3.2 Método de poda 67
3.4 Predição de séries temporais usando *ensemble* 68
3.5 Mistura de Especialistas 73
3.5.1 Propostas concorrentes à mistura de especialistas 75
3.5.2 Arquitetura de Mistura de Especialistas 76
3.5.3 Formas de aprendizado em mistura de especialistas 78
3.5.3.1 Treinamento acoplado: baseado no método do gradiente simples 80
3.5.3.2 Treinamento desacoplado via o método EM (Expectation-Maximization) 82
3.5.4 Predição de séries temporais usando mistura de especialistas heterogêneos 84
3.5.5 Outras propostas para ajuste de parâmetros em MEs 88
3.5.5.1 Computação evolutiva 88
3.5.5.2 Métodos Bayesianos 88

Capítulo 4 – Seleção de variáveis e predição de séries temporais

4.1. Introdução 89
4.1.1. Extração de características e seleção de variáveis 89
4.1.2. Extração e seleção em predição de séries temporais 93
4.2. Abordagens em seleção de variáveis 95
4.2.1. Filtro 95
4.2.2. Envoltório (*Wrapper*) 97
4.2.3. Embutido (*Embedding*) 98
4.2.4. Método híbrido 99

4.3. Seleção de variáveis em predição de séries temporais via o método Envoltório 99

4.3.1. Aspectos a considerar na abordagem envoltório 100

4.3.2. Seleção Progressiva (SP) 100

4.3.3. Seleção via Poda Baseada em Sensibilidade (PBS) 103

4.4. Simulações e resultados aplicando SP e PBS em predição de séries temporais 105

4.4.1. Resultados obtidos 107

4.5. Considerações finais 116

Capítulo 5 – Conclusões e perspectivas futuras

5.1 Conclusões 117

5.1.1 Conclusões relacionadas a comitês de máquinas 117

5.1.2 Conclusões relacionadas à Seleção de Variáveis 118

5.2 Perspectivas futuras 119

Anexo A

Auto-correlação e informação mútua para cálculo da janela de predição 123

Anexo B

Publicações vinculadas a esta dissertação 131

Referências Bibliográficas 133

Lista de Figuras

Capítulo 1

Figura 1.1:	Representação gráfica de uma série temporal.	3
Figura 1.2:	Formas de abordar o problema de predição de séries temporais.	6

Capítulo 2

Figura 2.1:	Arquitetura da rede MLP: acima, arquitetura completa com uma camada oculta; abaixo a representação do primeiro neurônio oculto e do único neurônio de saída com suas correspondentes funções de ativação tanh e *linear*.	20
Figura 2.2:	Um exemplo de aplicação do critério de parada em validação cruzada.	23
Figura 2.3:	Abordagem de construção do modelo para predição.	26
Figura 2.4:	Exemplo de uma série temporal que apresenta tendência linear crescente, sazonalidade e uma observação aberrante ou *outlier* (série *Clothing store*).	29
Figura 2.5:	Arquitetura da rede *NARMA recorrente*, proposta por CONNOR & MARTIN (1994).	33
Figura 2.6:	Resumo pelo número de casos favoráveis, não favoráveis e equivalentes das RNAs frente aos modelos tradicionais.	37
Figura 2.7:	Preparação dos dados para predição de um passo à frente com L=6 valores atrasados da série.	39
Figura 2.8:	Formas de separar os dados para compor os conjuntos de treinamento e validação, tomando por base a Tabela à direita na Figura 2.7. As porcentagens 50%, 25% e 25% representam apenas uma sugestão.	40
Figura 2.9:	Séries temporais testadas.	41

Capítulo 3

Figura 3.1:	Propostas de Comitês de Máquinas, (a) *ensemble* e (b) mistura de especialistas.	49
Figura 3.2:	Exemplo didático, aplicação do conceito de *ensemble* para reduzir a probabilidade de fracasso num canal de comunicações.	50
Figura 3.3:	Exemplo ilustrativo de ganho de desempenho empregando *ensemble* em predição de séries temporais.	54
Figura 3.4:	Exemplo ilustrativo do ganho de desempenho de um *ensemble* em problemas de classificação com múltiplas classes (POLIKAR, 2006).	55
Figura 3.5:	Etapas na construção de uma arquitetura para um *ensemble.*	59
Figura 3.6:	Geração de conjuntos de treinamento distintos via *bagging*.	62
Figura 3.7:	Combinação de componentes para tarefas de classificação (esquerda) e regressão (direita).	65
Figura 3.8:	Combinação de componentes em problemas de agrupamento.	66
Figura 3.9:	Gráficos *boxplot* obtidos a partir da Tabela 3.1.	71
Figura 3.10:	Número final de componentes selecionados em *Ensemble-Averaging* (esquerda) vs *Ensemble-Bagging* (direita), para cada série temporal testada.	72
Figura 3.11:	Exemplo de predição: série *Clothing store*, (a) única MLP, (b) *ensemble* total (50 componentes) e (c) *ensemble* com seleção (componentes selecionados: 8, 30, 21, 1, 23, 45, 10, 34).	72
Figura 3.12:	Estrutura típica de uma arquitetura de Mistura de Especialistas.	73
Figura 3.13:	Exemplo pictórico da decomposição do espaço de entrada via MEs.	74
Figura 3.14:	Exemplo de operação da rede *gating* para ponderar as saídas dos especialistas em função das entradas do problema.	75
Figura 3.15:	Aprendizado acoplado e simultâneo para mistura de especialistas.	79
Figura 3.16:	Aprendizado desacoplado para mistura de especialistas.	80
Figura 3.17:	Gráficos *boxplot* a partir da Tabela 3.3: MLP vs *Ensemble* vc MEs.	87

Capítulo 4

Figura 4.1:	Representação gráfica do espaço das variáveis relevantes e redundantes.	93
Figura 4.2:	Extração de características em predição de séries temporais.	94
Figura 4.3:	Seleção de variáveis em predição de séries temporais.	95
Figura 4.4:	Seleção de variáveis: Filtro.	96
Figura 4.5:	Seleção de variáveis: Envoltório.	97
Figura 4.6:	Seleção de variáveis: Embutido.	98
Figura 4.7:	Exemplo da execução da Seleção Progressiva, com L=12. Os dados correspondem à série *Furniture store* apresentada no Capítulo 2.	101
Figura 4.8:	Exemplo da execução de SBP, com L=12. Os dados correspondem à série *Furniture store* apresentada no Capítulo 2.	104
Figura 4.9:	Séries temporais consideradas.	106
Figura 4.10:	Formas de separar os dados para compor os conjuntos de treinamento e validação.	106
Figura 4.11:	Visualização gráfica dos resultados da Tabela 4.5 referentes ao conjunto de teste.	112
Figura 4.12:	Um exemplo gráfico na predição da série temporal *Clothing store*. Na esquerda: separação seqüencial e usando todos os atrasos, L=12 iniciais. Na direita: separação randômica e com os 5 atrasos obtidos via SP.	113

Capítulo 5

Figura 5.1:	A: visão tradicional e B: visão alternativa para predição de séries temporais.	120
Figura 5.2:	Visão atual de comitê de máquinas em predição de séries temporais.	121

Lista de Tabelas

Capítulo 2

Tabela 2.1:	Resultados comparativos entre a rede *NARMA recorrente* de CONNOR & MARTIN (1994), com uma MLP e uma rede totalmente recorrente. Valores numéricos entre parêntesis representam desvio padrão.	34
Tabela 2.2:	Acompanhamento de trabalhos comparativos entre RNAs e modelos tradicionais paramétricos empregados na predição de séries temporais.	35
Tabela 2.3:	Resultados de erro de predição MAE a partir do conjunto de teste das 5 séries temporais, inclui-se resultados de ZHANG & QI (2005).	44

Capítulo 3

Tabela 3.1:	Resultados comparativos empregando o conjunto de teste: única MLP, *Ensemble-Averaging* e *Ensemble-Bagging* com e sem Seleção.	69
Tabela 3.2:	Configurações de MEs testadas.	85
Tabela 3.3:	Resultados comparativos sobre o conjunto de teste: Única MLP, melhor *Ensemble* e MEs Heterogêneos.	86
Tabela 3.4:	Configurações mais adequadas para as séries temporais consideradas.	86

Capítulo 4

Tabela 4.1:	Possíveis subconjuntos para $L = 3$.	90
Tabela 4.2:	Número de subconjuntos candidatos para valores de L arbitrários.	91
Tabela 4.3:	Número de vezes em que cada atraso foi selecionado para compor o	108

subconjunto de variáveis de entrada do preditor. Foram executadas 30 vezes cada método, PBS e SP, para as 5 séries temporais testadas.

Tabela 4.4: Configuração final dos atrasos selecionados ao aplicar voto majoritário sobre os resultados da Tabela 4.3. 109

Tabela 4.5: Resultados para 30 execuções comparando “sem” (não uso de métodos de seleção de variáveis) vs “PBS” vs “SP”, cada um deles com separação de dados Seqüencial e Randômico. 111

Tabela 4.6: T-teste para análise de significância da diferença de desempenho entre os métodos PBS e SP. 114

Tabela 4.7: T-este para análise de significância da diferença de desempenho entre ausência de seleção e SP. 114

Tabela 4.8: T-este para análise de significância da diferença de desempenho entre ausência de seleção e PBS. 115

Tabela 4.9: T-este para análise de significância da diferença de desempenho entre separação seqüencial e randômica. 115

Lista de Abreviaturas

ACF	:	Função de Auto-correlação (*Auto-correlation Function*)
Adaboost	:	do inglês: *Ada*ptive *boost*ing
AR	:	Auto-regressivo
ARIMA	:	Auto-regressivo Médias Móveis Integrado (*Auto-regressive Integrated Moving Average*)
ARMA	:	Auto-regressivo Médias Móveis (*Auto-regressive Moving Average*)
Bagging	:	derivada de duas palavras: *B*ootstrap *agg*regat*ing*
CART	:	do inglês: *Classification and Regression Trees*
EQM	:	Error Quadratico Médio (**LMS -** *Least Mean Squares*)
EM	:	Maximização da Esperança (*Expectation Maximization*)
LME	:	Mistura de Especialistas Locias (*Local Mixture of Experts*)
NAR	:	Auto-regressivo não-linear (*Non-linear autoregressive*)
NMA	:	Médias móveis não-lineres (*Non-linear moving average*)
NARMA	:	Auto-geressivo médias móveis não lineares. (*Non-linear Auto-regressive Moving Average*)
MA	:	Médias móveis (*Moving Average*)
MAE		Erro Absoluto Médio (*Mean Absolute Error*)
MARS	:	do inglês: *Multivariate Adapative Regression Splines*

ME : Mistura de Especialistas (*Mixture of Experts*)

MI : Informação Mútua (*Mutual Informatin*)

MLP : Perceptron de multiplas camadas (*Multi-layer Perceptrons*)

OCR : Reconhecimento Ótico de Caracteres (*Optical Character Recognition*)

PAC : do inglês: *Probably Approximately Correct*

PCA : Analise de Componentes Principais (*Principal Component Analysis*)

RBF : Funções de Base Radial (*Radial Basis Function*)

RNAs : Redes Neurais Artificiais

SARIMA : Do inglês: *Seasonal-ARIMA*

SP : Seleção Progressiva

SBP : Seleção Baseada em Poda (*Sensitivity Based Pruning*)

STD : Desvio Padrão (*Standard Deviation*)

SVM : Maquinas de Vectores Suporte *(Support Vector Machines)*

TAR : do inglês: *Threshold auto-regressive*

TSP : Problema do Caixeiro Viajante (*Traveling Salesperson Problem*)

Lista de Símbolos

$\mathbf{X}$	:	Matriz das entradas dos padrões de treinamento, cada linha um padrão, cada coluna uma variável ou atributo ou característica.
$\mathbf{Y}$	:	Vetor das saídas dos padrões de treinamento, cada linha esta associada ao valor da linha correspondente em **X.**
χ	:	Conjunto dos padrões de treinamento, incluído a saída. Exclui a **X** e **Y**
w^a	:	Representam aos pesos das conexões antes da camada oculta numa RNA MLP com uma camada oculta.
w^d	:	Representam aos pesos das conexões depois da camada oculta numa RNA MLP com uma camada oculta.
$\hat{\mathbf{y}}$	:	Resposta da RNA MLP ante o ingresso de um padrão **x**.
E	:	Valor da somatória dos quadrados dos erros de um subconjunto de amostras.
θ_i	:	Vetor de parâmetros do especialista *i*.
$\mathbf{v}$	:	Vetor de parâmetros da rede *gating*.
Θ	:	Conjunto de parâmetros totais, especialistas e rede *gating*.
$P(\mathbf{y} \mid \mathbf{x}, \theta_i)$	:	Probabilidade do especialista *i* gerar a saída **y,** dado a entrada **x** e o vetor de seus parâmetros θ_i.
$P(i \mid \mathbf{x}, \mathbf{v}^0)$	:	Probabilidade de se escolher ao especialista *i*, dado a entrada **x** e o vetor dos parâmetros da rede *gating* **v**.

$g_i(\mathbf{x})$: Valor da saída *i* da rede *gating* ante um padrão de entrada **x**.

$l(\chi, \Theta)$: Função de verossimilhança que recebe ao conjunto de padrões de treinamento χ e ao conjunto de todos os parâmetros a ajustar Θ.

V : Conjunto que representa o espaço de variáveis.

$\hat{\mathbf{X}}$: Representa o subconjunto das variáveis de entrada como resultado do processo de seleção de variáveis.

E_c : Representa o erro de sensibilidade.

$\overline{x}_i$: Valor médio da variável $x_{i.}$.

$\hat{s}_t$: Valor estimado da série temporal no tempo *t*.

s : Representa uma série temporal.

Capítulo 1

Introdução

1.1 Escopo da dissertação

O escopo desta dissertação está relacionado ao estudo e uso de técnicas de aprendizado de máquina[1] para problemas de regressão não-linear de dados via comitê de máquinas (HAYKIN, 1999), empregando redes neurais artificiais (RNAs) com aprendizado supervisionado *off-line*. São tratados também dois métodos específicos de seleção de variáveis (GUYON & ELISSEEFF, 2003; KOHAVI & JOHN, 1997) empregando a abordagem de envoltório (do inglês *wrapper*). As aplicações das metodologias propostas estão voltadas para predição de séries temporais.

1.1.1 Séries temporais

Querer nos antecipar a um possível acontecimento é uma necessidade própria dos seres humanos. Desde muitos anos atrás, de forma empírica, tentou-se não só predizer, mas também explicar possíveis acontecimentos. Nas antigas culturas, somente seres com dons supostamente divinos podiam fazer isto em benefício da comunidade. Hoje, o científico superou o empírico, e muitos elementos dos modelos de regressão de séries temporais têm uma interpretação no mundo real, como tendências, sazonalidades, elasticidades, etc. Por

[1] Aprendizado de Máquina é conhecido na literatura em inglês como *Machine Learning.*

outra parte, a aplicação de técnicas matemáticas em séries temporais envolve a noção de predição[2] ou previsão.

> Predição: Ato ou efeito de predizer; profecia; vaticínio.
> Previsão : Ato ou efeito de prever; antevisão, presciência.
> Estudo ou exame feito com antecedência:
> Cautela, prevenção.

O resultado de uma predição auxilia na tomada de decisões em distintos campos do desenvolvimento humano. Dentre eles temos:

Decisões na produção em geral:

- Abastecimento antecipado;
- Atendimento a demandas;
- Índices de produção.

Decisões financeiras e comerciais:

- Compra / venda;
- Análise de investimentos.

Decisões atmosféricas, de meio ambiente e de saúde:

- Prognóstico do tempo;
- Precipitações e vazão de rios;
- Análise de índices de contaminação;
- Comportamento de infecções orgânicas;
- Reações de um organismo a medicamentos.

A Figura 1.1 mostra uma típica representação gráfica de uma série temporal, com o tempo no eixo horizontal e a magnitude da série no eixo vertical. Normalmente, dados históricos disponíveis da série temporal são utilizados para sintetizar um modelo preditor. Em seguida, este preditor é utilizado para realizar a predição de valores futuros da série.

[2] Definições extraídas do dicionário da língua portuguesa Aurélio. Neste documento, utilizaremos o termo "predição".

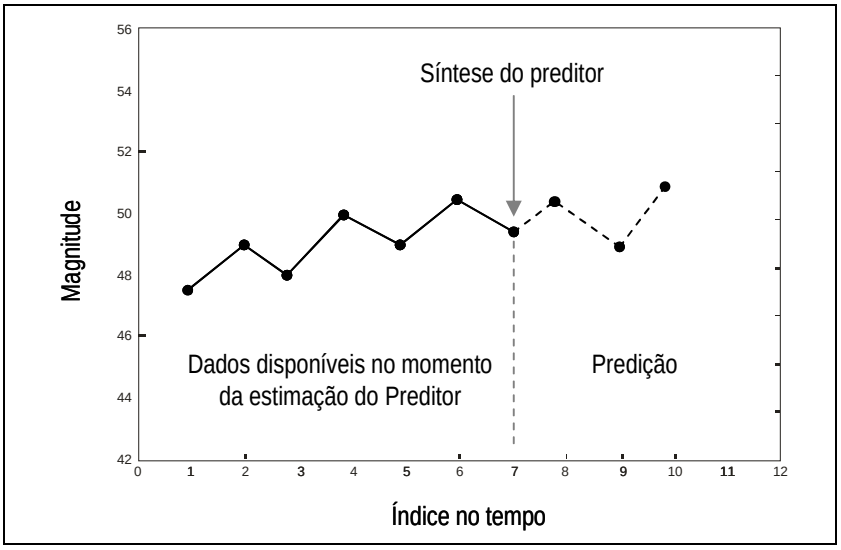


Figura 1.1- Representação gráfica de uma série temporal.

1.1.2 Redes neurais artificiais

Existem técnicas ou algoritmos computacionais que melhoram seu desempenho com a experiência e são aplicadas na resolução de problemas ligados à inteligência computacional. Tais técnicas estão inseridas no contexto de "aprendizado de máquina" (do inglês *machine learning*). Assim, na medida em que estas máquinas de aprendizado são submetidas a "novas experiências", seu desempenho tende a melhorar (MITCHELL, 1997).

As RNAs são claros exemplos de técnicas baseadas na abordagem de aprendizado de máquina, mas não são as únicas. Poderíamos mencionar também árvores de decisão, técnicas de computação evolutiva, máquinas de vetores suporte (do inglês *support vector machines - SVM*), entre outras. Todas elas têm uma característica em particular, que é a de aprender a partir dos dados ou amostras disponíveis (*learning from data*), descobrindo regras, relações ocultas, até chegar numa resposta. Segundo o tipo de aprendizagem, podem realizar tarefas de classificação, regressão ou agrupamento de dados. Existem outras tarefas que também são realizadas empregando RNAs, como memória associativa e otimização combinatória (HOPFIELD, 1982).

Identificação de sistemas do tipo caixa-preta é uma das áreas na qual as RNAs mostram-se altamente competitivas e está intimamente relacionada à aplicação deste trabalho. As razões principais para o uso de RNAs são as seguintes:

- Habilidade para aprender a partir dos dados;
- Capacidade de aproximação universal, incluindo aproximação de mapeamentos não-lineares;
- Capacidade de manipular múltiplas entradas e múltiplas saídas.

1.1.3 Comitê de máquinas

Comitê de máquinas é uma proposta baseada no princípio de "dividir-para-conquistar" (HAYKIN, 1999) e que busca superar o desempenho de uma máquina de aprendizado operando isoladamente. No caso dinâmico, o comitê é denominado mistura de especialistas, sendo que a divisão de tarefas e a síntese dos especialistas são realizadas simultaneamente e de forma automática pelo algoritmo de aprendizagem. No caso estático, o comitê é denominado *ensemble*, o qual busca agregar as respostas de múltiplos componentes sintetizados a priori (a síntese pode até ser independente), sendo que cada componente procura resolver todo o problema.

O ganho de desempenho com o comitê, quando comparado a uma única máquina de aprendizado, pode ser visto sob os seguintes ângulos:

- Melhor capacidade de generalização.
- Diminuição da variância do modelo;
- Maior tolerância a ruído nos dados;

1.1.4 Seleção de variáveis

Uma outra linha de pesquisa que visa auxiliar as propostas de aprendizado de máquina é seleção de variáveis. Recentemente, no trabalho de GUYON & ELISSEEFF (2003), foram classificadas três formas de seleção de variáveis: filtro, envoltório (do inglês *wrapper*) e embutido (do inglês *embbeding*). No filtro, a seleção das variáveis é realizada de forma independente do preditor (*model free*). Já em envoltório, as variáveis são selecionadas com o auxílio do preditor (*model based*). Na versão embutida, por sua vez, o processo de seleção de variáveis é intrínseco ou faz parte do algoritmo de treinamento do modelo preditor.

Os objetivos genéricos do uso de seleção de variáveis são os seguintes:

- Melhora na qualidade do resultado do preditor, como conseqüência da eliminação de variáveis não-relevantes e/ou redundantes;
- Diminuição do número de parâmetros do preditor, ao desconsiderar variáveis redundantes ou pouco informativas, e em conseqüência diminuição de memória e do esforço computacional requerido;
- Melhor entendimento do processo gerador dos dados.

1.2 Motivação da proposta

A motivação deste trabalho pode inicialmente ser sustentada pela proposta da Figura 1.2, onde mostram-se algumas formas de tratar o problema de predição de séries temporais, identificadas ao realizar uma revisão envolvendo a grande variedade de trabalhos da literatura.

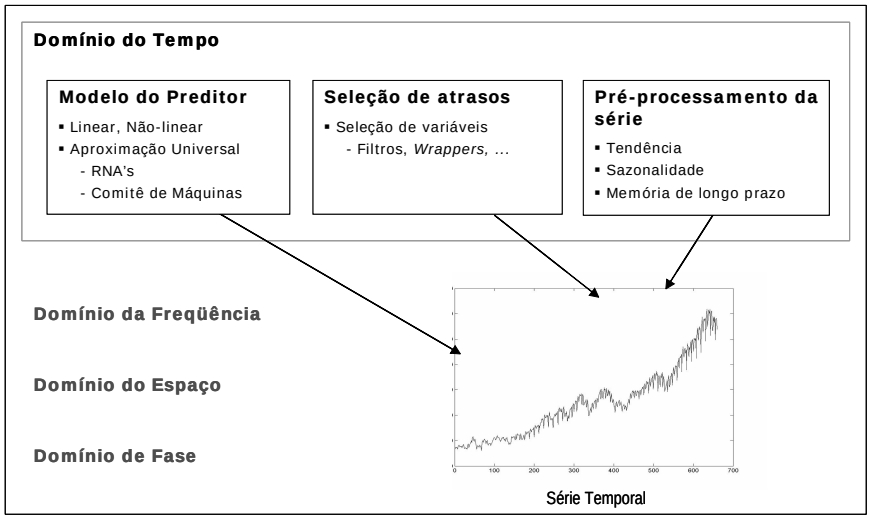


Figura 1.2 - Formas de abordar o problema de predição de séries temporais.

Restringindo a abordagem ao domínio do tempo, identificam-se três formas de tratar o problema: escolha do modelo do preditor, seleção de atrasos e pré-processamento da série temporal.

Assim, no caso da escolha do **modelo do preditor,** que é usado para detectar a correlação linear ou não-linear entre valores passados da série visando predizer valores futuros da mesma, chega-se a um problema de regressão de dados. Os preditores lineares foram amplamente estudados nos anos 1970 e inícios dos anos 1980, principalmente. Esses preditores lineares modelam correlações lineares entre os valores da série temporal. Por exemplo, um dos modelos clássicos na literatura é o ARIMA (BOX, *et al*., 1994). Por outro lado, preditores não-lineares surgiram como alternativa aos modelos lineares visando tratar séries temporais não-lineares.

Uma outra forma de abordar o problema de séries temporais refere-se à **seleção de atrasos**, que pode ser considerada como um problema de seleção de variáveis ou características (GUYON & ELISSEEFF, 2003; KOHAVI & JOHN, 1997). Assim, mostra-se de grande interesse prático investir em métodos que conduzam à identificação daqueles valores passados da série que mais benefícios tragam ao realizar a predição.

O **pré-processamento da série** é uma outra forma de abordar o problema de predição de séries temporais. Ele compreende o tratamento ou conjunto de transformações prévias a que a série pode ser submetida. Por exemplo, tratamentos de extração de componentes sazonal e tendência. Na maioria das vezes, as predições com modelos preditores lineares são antecedidas por este tipo de tratamento.

Nesta dissertação, por serem considerados preditores não-lineares, dar-se-á maior atenção às questões do modelo do preditor e da seleção de atrasos.

1.3 Objetivos

Os objetivos desta dissertação podem ser apresentados da seguinte forma:

A. Estudo de formas computacionais que permitam agrupar mais de uma máquina de aprendizado, visando superar o desempenho de qualquer uma delas quando atuando isoladamente. No caso, as máquinas de aprendizado serão representadas por uma RNA do tipo perceptron de múltiplas camadas ou simplesmente MLP (do inglês *Multi-layer Perceptron*) (RUMELHART & MCCLELLAND, 1986).

B. Estudo de métodos de seleção de variáveis visando atender a demanda por seleção de atrasos em problemas de predição de séries temporais.

Ambos os objetivos guardam relação com as formas de tratar o problema de predição de séries temporais apresentados na Figura 1.2. Consultando a literatura (NELSON *et al.*, 1999; ZHANG *et al*., 1998; ZHANG & QI, 2005), percebe-se que o sucesso na predição envolve: (*i*) a escolha de um modelo preditor flexível, possivelmente não-linear; (*ii*) identificação dos valores passados mais adequados para predizer valores futuros. Nesse sentido, nosso estudo concentra-se em modelos não-lineares, seja na síntese de comitês, seja na seleção de variáveis.

1.4 Metodologia

Visando alcançar os objetivos do trabalho, para a parte A, o estudo se concentrou nas propostas de comitê de máquinas: *ensemble* e mistura de especialistas. Um *ensemble* permite combinar as respostas de várias propostas de máquinas de aprendizado com o intuito de obter uma resposta global. Cada proposta atenderia o total do problema e o resultado seria uma espécie de consenso dos participantes. Para o caso de mistura de especialistas, além da combinação das propostas de máquinas de aprendizado, denominadas especialistas, resolvendo diretamente o problema, existe uma outra máquina que aprende a discriminar aspectos do problema e alocar para cada um deles um único ou um subconjunto de especialistas. Com essas duas abordagens, é possível obter melhores resultados em termos de desempenho, quando comparado ao melhor dos componentes do comitê, tomado isoladamente (HAYKIN, 1999).

Para atender a parte B dos objetivos, aborda-se um dos tipos de seleção de variáveis chamado envoltório (do inglês *wrapper*) (GUYON & ELISSEEFF, 2003), tendo como preditor uma rede neural MLP.

1.5 Contribuições

As contribuições desta dissertação estão diretamente relacionadas às motivações descritas na seção 1.2:

- Estudo e uso das propostas de comitê de máquinas: *ensemble* e mistura de especialistas como ferramentas de regressão, tendo como componentes as RNAs, no caso a MLP. Este ponto inclui uma revisão bibliográfica do estado da arte, assim como uma descrição das implementações realizadas. Finalmente, um estudo comparativo do desempenho alcançado pelas propostas: uma única MLP, *ensemble* de MLPs e mistura de especialistas na forma de MLPs, aplicados na modelagem e predição de cinco séries temporais reais de natureza financeira.

- Estudo e utilização de duas técnicas de seleção de variáveis que empregam a abordagem envoltório (*wrapper*), aplicadas ao problema de predição de séries temporais: Seleção Progressiva - SP (do inglês *Forward Selection*) e Poda Baseada em Sensibilidade - PBS (do inglês *Sensitivity Based Pruning*). Este ponto compreende uma revisão do estado da arte das abordagens de seleção de variáveis, descrição das técnicas específicas utilizadas, assim como resultados de um estudo comparativo entre SP e PBS.
- Comparação de duas técnicas de partição do conjunto de dados disponível para treinamento, responsáveis pela definição dos conjuntos de treinamento, validação e teste.

1.6 Organização e conteúdo da dissertação

O Capítulo 2 é dedicado a uma descrição das redes neurais artificiais aplicadas ao problema de predição de séries temporais, com ênfase na rede MLP. A formalização deste capítulo deve servir de base para o Capítulo 3, que trata de comitê de máquinas e são descritas algumas propostas da literatura para projetar um *ensemble* e uma mistura de especialistas, visando resolver problemas de regressão de dados. Incluem-se também simulações e resultados com ambas as abordagens.

O Capítulo 4, por sua vez, aborda o tema de seleção de variáveis, acompanhado pelo estado da arte e a descrição de dois métodos específicos de seleção, aplicados ao problema de predição de séries temporais.

O Capítulo 5 apresenta as conclusões do trabalho, assim como algumas considerações que apontam para trabalhos futuros.

Capítulo 2

Redes neurais artificiais em predição de séries temporais

Resumo: Este capítulo procura destacar aspectos conceituais e técnicos que justificam a aplicação de redes neurais artificiais na predição de séries temporais. O problema de predição será visto como um problema de regressão de dados, para o qual será adotado o modelo perceptron de múltiplas camadas (MLP, do inglês *Multi-layer Perceptron*). Apresentam-se as principais características das séries temporais, assim como as considerações principais para o emprego adequado de uma MLP como preditor. Finalmente, são realizadas predições de cinco séries temporais financeiras, com os resultados sendo comparados com propostas alternativas presentes na literatura.

2.1 Introdução às redes neurais artificiais

Redes Neurais Artificiais (RNAs) podem ser entendidas como dispositivos de processamento de informação caracterizados pela interconexão de unidades elementares de processamento, simples e similares entre si (HAYKIN, 1999). Trata-se de uma iniciativa de modelagem matemática de algumas propriedades do cérebro, sendo atribuído ao padrão de conexões e aos valores dos pesos sinápticos o papel de moldar o comportamento de entrada-saída da rede neural, ou seja, a forma como a rede neural irá responder a certos estímulos de entrada.

Algoritmos de aprendizado devem, portanto, ser concebidos de modo a promover o ajuste das conexões sinápticas, supondo já definido o padrão de conexões ou a arquitetura

da rede neural, visando obter o comportamento de entrada-saída desejado. O aprendizado pode então ser interpretado como um processo de condicionamento da rede neural, capaz de reproduzir em computador associações entre estímulos e respostas, algo que o cérebro realiza com grande eficácia (KOHONEN, 1988).

Particularmente no contexto de aprendizado supervisionado, situação em que se conhece a resposta desejada a um subconjunto finito de estímulos de entrada, é explorada a capacidade de aproximação universal das redes neurais artificiais (HORNIK *et al.*, 1989), a qual afirma que existe um padrão de conexões em camadas e valores para os pesos sinápticos que permitem reproduzir, com um grau de precisão arbitrário, qualquer mapeamento contínuo e restrito a uma região compacta do espaço de entrada. Apesar de oferecer um enorme potencial de solução de problemas vinculados à síntese de mapeamentos de entrada-saída, esta propriedade existencial, no entanto, não aponta como deve se dar o processo de aprendizado, vinculado à necessidade de maximizar a capacidade de generalização (BISHOP, 1995). Dado que o conjunto de amostras de entrada-saída é finito, existem infinitos mapeamentos que respondem bem a este conjunto de amostras, ou seja, existem infinitas configurações de redes neurais que respondem igualmente bem aos estímulos contidos no conjunto de treinamento. Logo, qual dentre as possíveis propostas de redes neurais artificiais deveria ser escolhida? A resposta está na escolha daquela, cujo mapeamento responda melhor a estímulos não apresentados durante o processo de treinamento, também chamados de amostras de validação.

2.1.1 Taxonomia

As redes neurais artificiais podem ser classificadas segundo os critérios explicitados ao longo das próximas sub-seções.

2.1.1.1 Tipo de associação entre as informações de entrada e saída

Auto-associativas:

A rede armazena certos padrões recebidos no processo de treinamento, por ajuste de sinapses. Quando se lhe apresenta uma informação incompleta ou com ruído, ela realizará uma associação e responderá com o padrão mais parecido dentre os já armazenados. Exemplos de redes auto-associativas mais conhecidas na literatura: Rede de Hopfield (HOPFIELD, 1982), ART (CARPENTER & GROSSBERG, 1988), Mapas Auto-Organizáveis de Kohonen (KOHONEN, 1988).

Hetero-associativas:

A rede armazena certas associações de entrada-saída recebidas no processo de treinamento, por ajuste de sinapses. Assim, quando se lhe apresenta um certo estímulo de entrada, ela deverá responder gerando a correspondente saída. Exemplos de redes hetero-associativas mais conhecidas na literatura: rede MLP (RUMELHART & MCCLELLAND, 1986), rede RBF (do inglês *Radial Basis Function*) (BROOMHEAD & LOWE, 1988).

2.1.1.2 Tipo de arquitetura

Pelo número de camadas:

Redes de uma camada, onde cada um dos neurônios recebe a entrada e produz a saída final. Este tipo de rede, geralmente está associado a tarefas auto-associativas, por exemplo, reconstruir padrões incompletos ou com ruído.

Redes de várias camadas, onde os neurônios estão dispostos em vários níveis ou camadas: *camada de entrada*, que recebe os padrões de entrada; uma ou várias camadas ocultas, que geralmente realizam o mapeamento de classes ou regressão; e a *camada de saída*, que nos casos auto-associativos realiza um processamento de associação, mas nos casos hetero-associativos compõe a saída combinando a informação proveniente da última camada oculta.

Pelo tipo de conexões:

Feedforward, onde os sentidos das conexões são para a frente ou diretas. Por exemplo: MLP, RBF.

Feedforward/feedback, onde existem conexões de realimentação, além das diretas. Por exemplo: redes recorrentes (CONNOR & MARTIN, 1994). Essas redes podem ser totalmente ou parcialmente recorrentes.

2.1.1.3 Tipo de mecanismo de aprendizagem

A aprendizagem é um processo no qual a RNA modifica seus pesos (conexões sinápticas) em função de:

- Informação de entrada;
- Informação de entrada associada a uma saída desejada.

Análogo ao caso biológico, onde os estímulos recebidos promovem modificações nas intensidades das sinapses, os mecanismos de aprendizado em redes neurais artificiais buscam ajustar as conexões ou pesos sinápticos em resposta aos estímulos recebidos.

O fundamental no processo de aprendizagem é definir como tais pesos serão alterados quando se requer que a rede aprenda uma nova informação. Tais critérios podem ser classificados como *supervisionados* ou *não-supervisionados*, e vai depender do problema a resolver. A principal diferença entre estes dois tipos de aprendizagem, como já mencionado, está na existência ou não de um agente externo ou supervisor que controla o processo de aprendizagem.

Supervisionado:

Nesta forma de aprendizado, um supervisor é quem determina a resposta que a rede deverá dar para uma entrada determinada. Este mecanismo está fortemente associado a redes de tipo hetero-associativas. O supervisor verifica a saída da rede e, caso ela não coincida com a saída desejada, fará um ajuste nos pesos das conexões visando minimizar esta diferença. Por exemplo:

- *Por correção do erro*: onde o ajuste está em função do erro cometido, o qual pode ser calculado como a diferença entre os valores desejados e os obtidos pela rede. O algoritmo mais conhecido relacionado a este tipo de aprendizado é o *backpropagation* (WERBOS, 1974) para redes *feedforward*, e extensões deste algoritmo para redes *feedforward/feedback* ou redes recorrentes (PINEDA, 1987; ALMEIDA, 1987).
- *Por reforço*: variante com um grau menor de supervisão que a anterior, no qual não se dispõe de um exemplo completo do comportamento desejado. Esta forma de treinamento é análoga ao agir de um crítico, que em vez de corrigir baseado na diferença relativa entre a resposta desejada e a obtida, ele fornece apenas um indicativo de nível de sucesso ou fracasso vinculado a uma seqüência de ações da rede neural. Alguns exemplos deste tipo de algoritmo: *Linear Reward-Penalty* (NARENDRA & THATHACHER, 1974), *Associative Reward-Penalty* (BARTO & ANANDAN, 1985).

Não-supervisionado:

Para este tipo de aprendizado, não existe supervisor nem crítico. É conhecido também como aprendizado auto-supervisionado, não requerendo indicativos de comportamento desejado para a rede neural. Com isso, interpreta-se o processo de ajuste de conexões como resultado de um processo de auto-organização. Aplica-se em redes auto-associativas e pode ser de dois tipos:

- *Hebbiano*: (HEBB, 1949): conexões associadas a neurônios que se encontram ativos simultaneamente tendem a serem fortalecidas, enquanto que conexões associadas a neurônios que sofrem ativações em instantes descorrelacionados no tempo, tendem a serem enfraquecidas.
- *Competitivo e cooperativo*: nesta classe de algoritmos, os neurônios concorrem por representar as amostras de entrada e aquele que vence a competição tem seu vetor de pesos ajustado incrementalmente, na direção da amostra de entrada, assim como os neurônios vizinhos ao vencedor, mas estes últimos, com menor intensidade de ajuste.

2.1.1.4 Tipo de procedimento de ajuste das conexões sinápticas

Off-line:

O ajuste dos pesos se dá anteriormente à colocação em operação da rede neural. Quando em operação, os pesos sinápticos da rede neural são fixos.

On-line:

Neste modo, não há distinção de fases de treinamento e operação, pois os pesos variam incrementalmente sempre que se apresenta um novo estímulo de entrada e o comportamento a cada instante da rede neural depende dos valores atuais dos pesos sinápticos.

2.1.2 Motivações para o emprego de redes neurais artificiais

As principais motivações que levam ao emprego de redes neurais artificiais são as seguintes:

- *Habilidade de aprender a partir dos dados amostrados*: o aprendizado é realizado de forma incremental ou por condicionamento, a partir da exposição repetida a amostras de treinamento disponíveis. No caso de aprendizado supervisionado, as amostras são de entrada-saída, ou seja, há uma resposta desejada a cada estímulo de entrada. O ajuste das conexões sinápticas pode ser visto então como um problema de otimização cuja função-objetivo mede o erro na saída da rede neural. Já no caso de aprendizado não-supervisionado, as amostras são apenas de entrada e cabe à rede neural reproduzir em suas conexões sinápticas propriedades estatísticas presentes nas amostras de entrada. Isso se dá por mecanismos de auto-organização.
- *Capacidade de aproximação de mapeamentos estáticos não-lineares e multidimensionais:* o número de entradas e de saídas (estas últimas para o caso supervisionado) pode ser arbitrário, assim como o tipo de associação entre as

entradas e as saídas, desde que o mapeamento desejado seja contínuo e sua aproximação esteja restrita a uma região compacta do espaço de entrada.

- *Capacidade de aproximação de mapeamentos dinâmicos não-lineares e multidimensionais:* a identificação de sistemas dinâmicos pode se dar pela presença de conexões recorrentes e/ou pela presença de linhas de derivação de atraso (NARENDRA & PARTHASARATHY, 1990; NARENDRA & PARTHASARATHY, 1991)
- *Habilidade para implementação de memória associativa e para a solução de problemas combinatórios pela busca de pontos de equilíbrio em dinâmicas de relaxação:* essas propriedades estão presentes em redes recorrentes de Hopfield (HOPFIELD, 1982), as quais implementam sistemas dinâmicos não-lineares que podem apresentar uma multiplicidade de pontos de equilíbrio com funcionalidades específicas.

Nesta dissertação, serão exploradas basicamente as duas primeiras motivações, embora as outras duas também contribuam para a obtenção de preditores de séries temporais (NELSON *et al*. 1999; ZHANG *et al*., 1998; ZHANG & QI, 2005).

2.1.3 Principais aplicações

Nos últimos anos, as redes neurais artificiais vêm sendo utilizadas de forma bem-sucedida como ferramentas de solução para uma ampla gama de problemas, sendo que as sub-seções a seguir procuram levantar as principais classes de aplicações.

2.1.3.1 Problemas de reconhecimento de padrões

Agrupamento ou clusterização de dados:

- As amostras não estão associadas a classes, ou seja, não se encontram previamente rotuladas.
- O objetivo é descobrir agrupamentos com características em comum.

- Aplica-se aprendizado não-supervisionado.

  Exemplos:

  - ✓ Mineração de dados e de textos;
  - ✓ Síntese e quantização de informação.

Classificação de padrões:

- Cada amostra está associada a uma classe, ou seja, as amostras estão previamente rotuladas.
- O objetivo é sintetizar um classificador capaz de generalizar corretamente, ou seja, classificar novas amostras ainda não-rotuladas.
- Aplica-se aprendizado supervisionado.

  Exemplos:

  - ✓ Reconhecimento de pessoas pelo rosto, voz ou digitais.
  - ✓ Reconhecimento e classificação de textos, reconhecimento óptico de caracteres (OCR, do inglês *Optical Character Recognition*).

2.1.3.2 Problemas de regressão de dados

Dentre os problemas que envolvem regressão de dados tem-se:

- Aproximação de funções (VON ZUBEN, 1996);
- Identificação de sistemas (JAIN & MAO, 1996);
- Controle de processos (NØRGAARD *et al*., 2000);
- Filtragem adaptativa (WIDROW & STERNS, 1985);
- Predição de séries temporais (ZHANG *et al*., 1998).

2.1.3.3 Problemas de otimização combinatória

Problemas de otimização combinatória podem ser resolvidos via RNAs, por exemplo, problemas de caixeiro viajante (TSP, do inglês *Traveling Salesperson Problem*) e problemas de roteamento de veículos. Para tanto, podem ser empregadas redes neurais de Hopfield (HOPFIELD, 1982) e redes neurais auto-organizáveis (GOMES & VON ZUBEN, 2002).

2.1.3.4 Outras aplicações

- Encriptação de dados (LI, 2004);
- Redução da dimensionalidade de dados (HINTON & SALAKHUTDINOV, 2006).

Evidentemente aqui a lista não é exaustiva, apenas ilustrativa.

2.1.4 Descrição do perceptron de múltiplas camadas - MLP

A partir da taxonomia das RNAs apresentada anteriormente, a rede neural MLP (RUMELHART & MCCLELLAND, 1986) é uma rede hetero-associativa, que pode apresentar várias camadas ocultas. No entanto, para ser um aproximador universal, é preciso apenas uma camada oculta com funções de ativação sigmoidais: logística ou tangente hiperbólica. O aprendizado obedece a um processo supervisionado por correção do erro.

Na Figura 2.1, apresenta-se a arquitetura de uma rede neural MLP com uma saída, sendo que, no caso geral, podem existir múltiplas saídas.

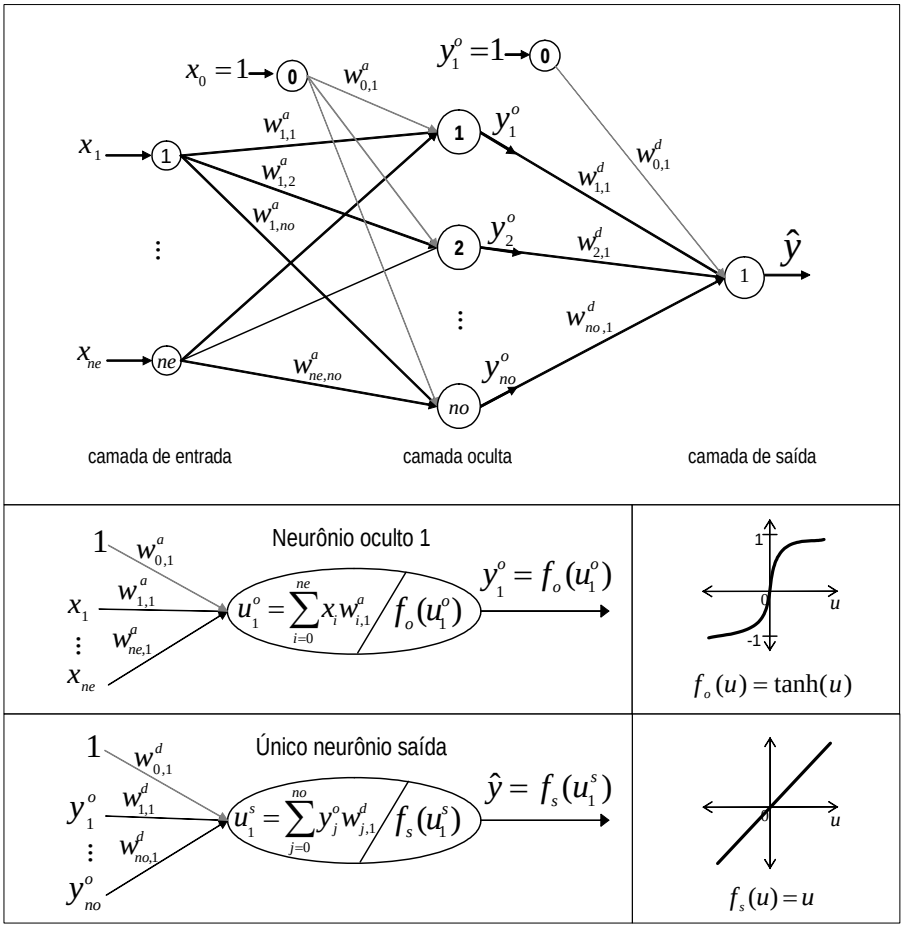


Figura 2.1- Arquitetura da rede MLP: acima, arquitetura completa com uma camada oculta; abaixo a representação do primeiro neurônio oculto e do único neurônio de saída com suas correspondentes funções de ativação *tanh* e *linear*.

É dado destaque também nesta Figura 2.1 para a representação de dois neurônios: ao Neurônio 1 da camada oculta e ao único neurônio da camada de saída da rede. Repare que cada neurônio da camada oculta e da camada de saída realizam uma combinação linear entre suas entradas e os respectivos pesos sinápticos $w^a_{i,j}$ e $w^d_{j,1}$ ($i = 0,1,2,...,ne$ e $j = 0,1,2,...,no$), produzindo as ativações internas u^o_i (camada oculta) e u^s_1 (camada de saída). Após estas combinações, estas ativações internas serão aplicadas nas funções de ativação f_o e f_s dos neurônios das camada oculta e de saída, respectivamente, para formar a saída final de cada neurônio. As funções f_o são do tipo sigmoidal e f_s do tipo linear.

Esta configuração é comumente empregada em problemas de regressão de dados, incluindo o problema de predição de séries temporais. Vale indicar que não existem ativações internas nos neurônios da camada de entrada (aqueles numerados de 0 a *ne*) dado que não há pesos sinápticos antes dela e as suas saídas correspondem ao mesmo valor que das suas entradas $x_0, x_1, x_2, ...x_{ne}$.

As saídas y_j^o ($j = 0,1,2,..., no$) dos neurônios da camada oculta podem ser representadas pela equação (2.1), onde ne é o número de entradas, e $w_{i,j}^a$ ($i = 0,1,2,..., ne$ e $j = 0,1,2,..., no$) os pesos das conexões sinápticas antes da camada oculta.

Algo que será útil posteriormente é levar em conta que o valor das funções de ativação f_o e f_s, de cada neurônio (camada oculta e de saída), dependem dos valores dos pesos das conexões que conduzem sinais até aquele neurônio. Como indicado nas equações (2.1) e (2.2).

$$y_j^o = f_o(\sum_{i=0}^{ne} x_i w_{i,j}^a) . \tag{2.1}$$

Na equação (2.2), $w_{j,1}^d$ ($j = 0,1,2,..., no$) são os pesos das conexões sinápticas depois da camada oculta e $\hat{y}$ corresponde à saída do único neurônio da camada de saída e, por conseqüência, à saída global da rede.

$$\hat{y} = f_s(\sum_{j=0}^{no} y_j^o w_{j,1}^d) . \tag{2.2}$$

2.1.4.1 Descrição do algoritmo de treinamento

O treinamento de uma MLP consiste em encontrar valores adequados para todos os pesos das conexões $w_{i,j}^a$ e $w_{j,1}^d$ (pesos antes e depois da camada oculta, respectivamente), de forma que, ao ingressar com um padrão de entrada, a saída da rede seja o valor mais próximo ao valor desejado associado a tal entrada. Para isto, precisa-se de:

- *Dados de treinamento*: chamados de amostras de treinamento, onde cada amostra consta de valor(es) de entrada (X) associados com valor(es) de saída desejado(s) (Y).
- *Uma função-objetivo*: por exemplo, a *função somatória dos quadrados dos erros* (*E*), apresentada na equação (2.3), onde o erro é entendido ser a diferença entre o valor de saída obtido pela rede $\hat{y}^p$ e o valor de saída desejado y^p, com N sendo o número de padrões de treinamento e $p = 1,2,...,N$.

$$E = \frac{1}{2}\sum_{p=1}^{N}\left(\hat{y}^p - y^p\right)^2 \tag{2.3}$$

- *Algoritmo de ajuste dos pesos*: trata-se de um algoritmo para otimização não-linear irrestrita. São várias as possibilidades para a MLP. O mais básico dos algoritmos de otimização emprega informação de primeira ordem e corresponde ao método do gradiente. Algoritmos que empregam informação de 2a. ordem também podem ser utilizados (BATTITI, 1992). Qualquer que seja o método de otimização, será necessário empregar o algoritmo de retropropagação, também denominado de *backpropagation* (WERBOS, 1974) para se obter o vetor gradiente.
- *Procedimento para evitar sobre-ajuste (over-fitting)*: também conhecido como sobre-treinamento, acontece quando a rede parece estar representando o problema cada vez melhor, ou seja, o erro junto ao conjunto de treinamento vai diminuindo. A questão é que, em algum ponto deste processo, a capacidade de responder adequadamente a um novo conjunto de dados, também denominada de capacidade de generalização, começa a piorar (PRECHELT, 1997). Dentre as formas empíricas de se evitar o sobre-ajuste (BISHOP, 1995), destacam-se: (*i*) validação cruzada; (*ii*) incorporação de um termo de penalidade na função-objetivo; e (*iii*) inserção de ruído nos dados de treinamento.

A *validação cruzada* (PRECHELT, 1997; PRECHELT, 1998), consiste em separar os padrões em três conjuntos: treinamento, validação e teste. O conjunto de treinamento é usado para ajustar os pesos sinápticos. Após cada época de ajuste de pesos, calcula-se o erro junto ao conjunto de validação. Assim, quando este valor apresentar uma tendência

definida de aumento, será um indicativo de sobre-ajuste e o treinamento deverá ser interrompido. O conjunto de teste, por sua vez, é utilizado para indicar como ficaria o desempenho da rede neural em operação. É suposto que os três conjuntos contêm amostras independentes e são todos capazes de representar bem o problema que está sendo abordado. Por exemplo, espera-se que um bom desempenho junto ao conjunto de validação implique em um bom desempenho junto ao conjunto de teste. Na Figura 2.2, ilustra-se este método.

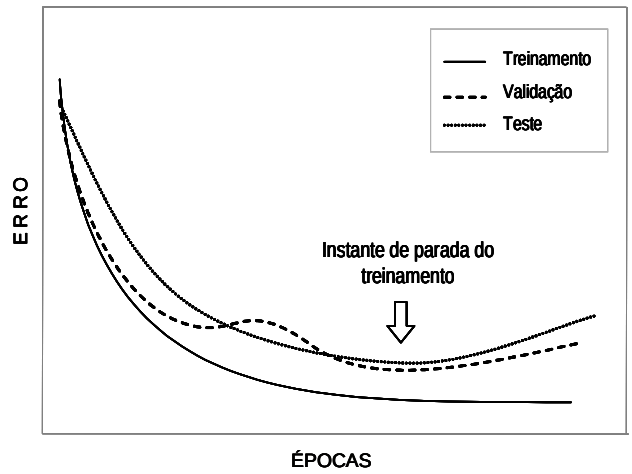


Figura 2.2 – Um exemplo de aplicação do critério de parada em validação cruzada.

A função-objetivo calcula o erro cometido entre a saída estimada pela rede e a saída desejada. Porém, o objetivo é minimizar a função em relação aos pesos sinápticos[1] $\mathbf{w}^a$ e $\mathbf{w}^d$, que correspondem às variáveis de otimização. Mas, na função-objetivo dada na equação (2.3) não aparecem explícitos os pesos sinápticos $\mathbf{w}^a$ e $\mathbf{w}^d$. Mas, realizando substituições a partir da saída final da rede (saída do único neurônio da camada de saída) representada na equação (2.3) e utilizando a equação (2.1) para o calculo das saídas y_j^o dos neurônios ocultos consegue-se obter a função-objetivo em função dos pesos sinápticos $\mathbf{w}^a$ e

[1] $\mathbf{w}^a$ e $\mathbf{w}^d$ são as representações em notação matricial dos pesos sinápticos $w_{i,j}^a$ e $w_{j,1}^d$, respectivamente.

$\mathbf{w}^d$. O gradiente desta função será produzido, então, empregando-se a *regra da cadeia* do cálculo diferencial.

2.2 Introdução às séries temporais

Um estudo aprofundado de séries temporais foge ao escopo desta dissertação, seja por envolver um elenco amplo de temas multi-disciplinares, seja por possuir um longo histórico de investigação, particularmente ao longo do último século. Neste sentido, a introdução a ser apresentada nas próximas seções visa levantar aspectos técnicos, definições básicas e exemplos simples.

Uma série temporal pode ser entendida como uma variável aleatória indexada no tempo, como segue:

$$\text{Série temporal: } s_1, s_2, s_3, s_4, s_5, ..., s_N \tag{2.4}$$

De uma maneira mais formal, uma série temporal é uma realização de um processo estocástico. Ela difere de um processo *iid* (independente e identicamente distribuído) já que existe uma dependência entre seus valores. Se não existisse, não se poderia realizar predição de valores futuros. Nesta dissertação, as séries temporais a serem tratadas têm seus valores definidos em intervalos de tempo igualmente espaçados.

Existe uma classe de processos estocásticos chamados de estacionários e estão baseados na premissa de estarem num estado particular de equilíbrio estatístico. Podem ser estritamente estacionários (ou de estacionariedade forte) quando a distribuição de probabilidade conjunta é a mesma em relação ao tempo, tomando como referência blocos de observações de mesmo tamanho em diferentes instantes de tempo. Estacionariedade no sentido amplo (ou de estacionariedade fraca) significa que a média, variância e covariância não variam no tempo.

O interesse em estudar séries temporais abrange os seguintes aspectos:

- *Análise* e *modelagem* da série temporal: descrever a série, identificar suas características relevantes e possíveis relações com outras séries.
- *Predição* de valores futuros da série: a partir de valores passados da série (às vezes, também, a partir de valores de outras séries), predizer um valor futuro s_{t+k}, onde s_t é o valor atual da série e k representa o horizonte de predição. Por exemplo, para predição um passo à frente, o objetivo seria predizer s_{t+1}.

No contexto de predição, a existência de dependência entre os valores da série é um requisito necessário, já que é justamente essa dependência que permite gerar predições. Sob a condição de independência dos valores da série, as predições não apresentariam correlação com os valores correspondentes da série temporal. Existem diferentes graus de dependência nas séries temporais, deixando a tarefa de predição com diferentes níveis de dificuldade. Por exemplo, tende a ser mais fácil predizer valores futuros de uma série de temperaturas ambientais médias mensais do que uma série de índices mensais da inflação.

Também é sabido que não existe um método de predição que seja bom para todas as séries, pois o seu desempenho está vinculado às características de cada série. Além disso, pode-se afirmar que o nível de incerteza aumenta com o horizonte de predição, ou seja, quanto maior o horizonte de predição, maior tende a ser a incerteza associada à predição.

Uma forma bem conhecida de construir um modelo para predição está baseada na metodologia de Box & Jenkins (BOX *et al.*, 1994). Na Figura 2.3, apresentam-se os passos desta metodologia. Trata-se de um procedimento iterativo, onde se parte da escolha de uma classe de modelos, depois realiza-se a definição da estrutura paramétrica do modelo, seguida do ajuste dos parâmetros, o que geralmente envolve a minimização de uma função-objetivo. Em seguida, executa-se uma etapa de diagnóstico ou validação do modelo e, finalmente, o modelo se encontra pronto para o seu uso em predição.

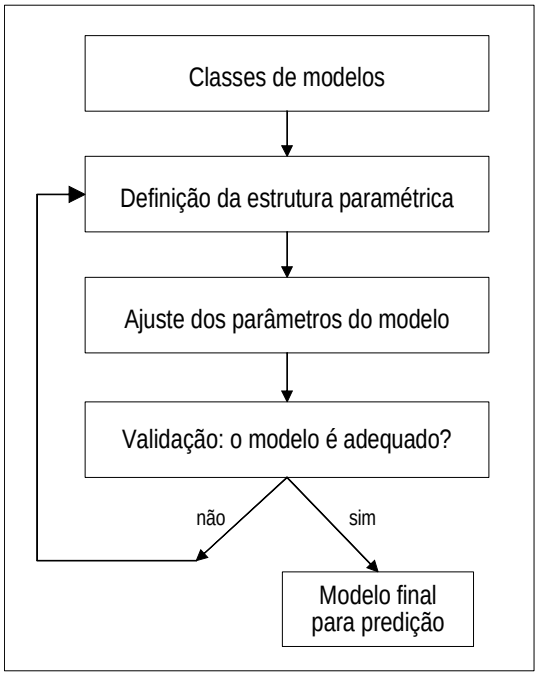


Figura 2.3 – Abordagem de construção do modelo para predição.

2.2.1 Principais características das séries temporais

A definição de série temporal pode ser simples e geral, mas a escolha de um modelo para predição que melhor explore as dependências entre os valores da série se constitui em uma tarefa desafiadora pela presença de: *tendência*, *sazonalidade*, *observações aberrantes* (*outliers*), *heterocedasticidade*, *não-linearidade*. Uma série temporal pode apresentar mais de uma dessas características, aumentando ainda mais o desafio na escolha e ajuste do modelo.

2.2.1.1 Tendência

Indica o comportamento de "longo prazo" do valor médio da série. Por exemplo, na Figura 2.4 apresenta-se uma série temporal com tendência linear crescente. Trata-se de uma série financeira com observações de vendas mensais de roupas nos Estados Unidos.

As tendências fazem com que a série seja não-estacionária e, para torná-la estacionária, alguns tratamentos da série são indicados. Um deles envolve a aplicação de *diferenciação*, que consiste em criar uma nova série temporal como resultado da subtração de valores consecutivos da série:

$$s_t^{nova} = s_t - s_{t-1}.$$

A nova série terá N–1 valores, sendo N o número de valores da série original. Para o caso da tendência quadrática, é preciso aplicar diferenciação duas vezes, e assim por diante. Com a diferenciação, espera-se obter uma série estacionária em relação à tendência central. No entanto, a variância pode sofrer incrementos no tempo, de modo que novas transformações sejam necessárias para casos em que a diferenciação não estabilize a variância (BOX *et al.*, 1994; MCCLEARY & HAY, 1980). É comum usar uma transformação logarítmica, não sem antes re-escalonar a série de tal forma que não apareçam valores menores ou iguais a zero.

Visando estabilizar a variância, eliminar assimetria e tornar a distribuição da série aproximadamente normal, são geralmente empregadas as transformações de Box-Cox (BOX *et al*., 1994).

2.2.1.2 Sazonalidade

A sazonalidade é caracterizada pela repetição de um padrão (ciclo) num período de tempo fixo. Por exemplo: temperatura ambiente e vazões de rios tendem a apresentar ciclos anuais. A série temporal da Figura 2.4 apresenta também sazonalidade, sendo que a cada doze meses repete-se um padrão. A observação aberrante em torno do instante 60 será tratada na próxima subseção.

A sazonalidade pode tornar à série temporal como não-estacionária. Em vista disso, alguns métodos propostos na literatura já buscam modelar e extrair da série a sua sazonalidade, sendo possível citar métodos como SARIMA (BROCKWELL & DAVIS, 1991) e X-12-ARIMA (FINDLEY *et al.*, 1998).

Na literatura, existem trabalhos que comparam empiricamente os diversos métodos para lidar com a sazonalidade (ALBERTSON & AYLEN, 1996; FRANSES & DIJK, 2005; KULENDRAN & KING, 1997; NOAKES *et al.*, 1985), sendo que os resultados foram diversos, dependem da classe de modelos e da natureza da série e não houve um consenso em se definir as condições nas quais um dos métodos seja preferido.

2.2.1.3 Observações Aberrantes

Observações aberrantes são associadas a distorções observadas na série ou observações que são altamente inconsistentes com outros comportamentos já observados da série. São caracterizadas, por exemplo, pela ocorrência *outliers* (uma ou mais observações não esperadas). Também podem envolver uma mudança do regime da série. Por exemplo, uma série que, a partir de um certo momento começa a apresentar sazonalidade. A série temporal da Figura 2.4 apresenta *outliers* em torno da observação 60, impedindo assim a ocorrência de um pico esperado.

Caso a série temporal sofra mudanças de regime, alguns tratamentos mais sofisticados para síntese do preditor devem ser considerados. Alguns exemplos de métodos na literatura que lidam com mudança de regime numa série temporal podem ser encontrados em PUMA-VILLANUEVA *et al.* (2005) e WEIGEND *et al.* (1995), ambos utilizando mistura de especialistas, sendo os especialistas redes neurais artificiais.

Já para o caso de *outliers*, por definição, não existe um algoritmo capaz de predizê-los (CRYER, 1986). Para identificá-los, o que se recomenda é examinar a série a partir da representação gráfica (antes e depois de algum ajuste ou transformação ser realizado) e realizar uma identificação visual, embora possam ser aplicados procedimentos automáticos para este fim.

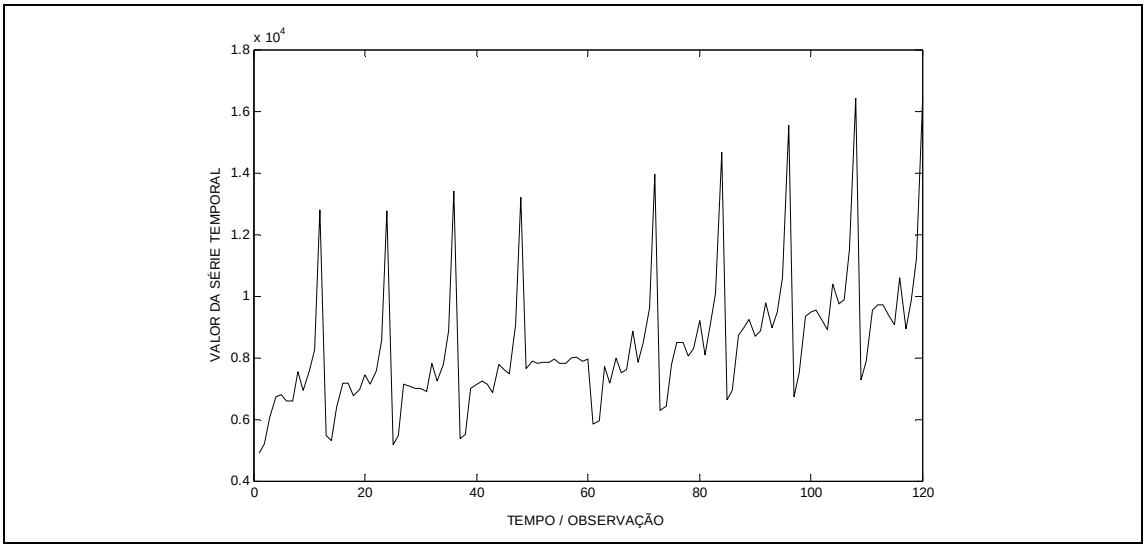


Figura 2.4 - Exemplo de uma série temporal que apresenta tendência linear crescente, sazonalidade e observações aberrantes ou *outliers* em torno da observação 60 (série *Clothing store*).

2.2.1.4 Heterocedasticidade

Quando a série temporal é tal que a variância não é constante para todas os diversos regimes da série, então, caracteriza-se um comportamento heterocedástico, refletindo isto na variância dos resíduos do modelo.

Por exemplo, é mais fácil estimar o gasto com alimentação como função da renda para uma família de baixa renda do que para uma família de alta renda. Logo, na medida em que a renda aumenta, a variância do erro de estimação aumenta também.

Séries com heterocedasticidade condicional podem ser tratadas com modelos auto-regressivos com heterocedasticidade condicional *ARCH* (ENGLE, 1982), ou com uma versão mais generalizada como o *GARCH* (BOLLERSLEV *et al.*, 1994; TAYLOR, 1987).

2.2.1.5 Não-linearidade

De forma geral e até, talvez, informal, a não-linearidade pode ser definida como qualquer comportamento que não seja linear. No contexto de séries temporais, implica que

a relação de dependência entre os valores da série é não-linear, ou dito de outra forma, a dependência entre as observações passadas da série e os valores a serem preditos é não-linear.

O estudo de modelos não-lineares para séries temporais, na literatura, é menos freqüente que o estudo de modelos lineares e geralmente está baseado em extensões dos modelos ARMA. No entanto, supor linearidade nas relações de dependência foi e é considerada uma estratégia válida e poderosa. Mas em finais dos anos 1970 e início dos anos 1980, modelos lineares mostraram-se insuficientes em muitas aplicações práticas.

Dentre as abordagens não-lineares que permitem modelar séries temporais, serão destacadas algumas abordagens a seguir. Atribuiu-se a VOLTERRA (1930) a primeira iniciativa em se analisar séries temporais não-lineares, e mostrou-se que qualquer função não-linear no tempo pode ser aproximada por uma série finita de Volterra. TONG (1983, 1990) propôs o TAR (*Threshold Auto-Regressive*) e o SETAR (*Self-Exciting TAR*), que buscam aproximar a não-linearidade via linearidade por partes e permitem o chaveamento entre regimes. O Modelo bi-linear (POSKITT & TREMAYNE, 1986) já emprega uma função quadrática. As redes neurais artificiais (RNAs) representam uma proposta bem sucedida para modelagem de séries temporais não-lineares, justamente por serem aproximadores universais e, assim, apresentarem flexibilidade suficiente para mapear qualquer relação de entrada-saída presente nos dados, desde que essa relação esteja restrita a uma região compacta do espaço de entrada e possa ser expressa por um mapeamento contínuo (HORNIK *et al*, 1989). Na literatura, existe um número expressivo de trabalhos aplicando RNAs em predição, muitos deles comparando-as com modelos lineares e outros modelos tradicionais (descritos na Seção 2.2.2 a seguir).

2.2.2 Modelos básicos para predição de séries temporais

A predição de séries temporais consiste em encontrar um modelo que descreva o comportamento da série e, a partir dele, buscar realizar predições de valores futuros. O ponto de vista tradicional é o seguinte:

- Propor um modelo paramétrico e realizar pré-processamentos junto à série temporal para aumentar a capacidade de predição do modelo proposto. Restrições geralmente adotadas são: a linearidade do modelo e a estacionariedade da série temporal.

A abordagem dessa dissertação difere desta perspectiva, sendo que procura-se atribuir ao modelo de predição flexibilidade suficiente para explorar as particularidades de cada série temporal a ser predita, com ou sem pré-processamento da mesma. O modelo de predição não só é não-linear como também semi-paramétrico. Com esta nova perspectiva, aumenta significativamente o grau de complexidade ao ajustar os parâmetros do modelo, que no caso de redes neurais artificiais correspondem aos pesos sinápticos. Perde-se, por exemplo, a garantia de otimização global dos parâmetros e entra em cena a questão da capacidade de generalização.

A seguir, é feita uma breve revisão dos modelos básicos de predição, para só então tratar da abordagem via RNAs.

2.2.2.1 Modelo auto-regressivo – AR(p)

Um modelo auto-regressivo supõe que a série temporal pode ser modelada a partir dos valores atrasados da série. Este modelo é ilustrado na equação (2.5), onde e_t representa ruído branco e os coeficientes $\phi_1,...,\phi_p$ representam os parâmetros a estimar. Como se trata de um caso linear, os coeficientes $\phi_1,...,\phi_p$ podem ser estimados empregando métodos baseados na minimização dos erros quadrados médios ou outros um tanto mais sofisticados, como o algoritmo Durbin-Levinson (BROCKWELL & DAVIS, 1991).

$$s_t = \phi_1 s_{t-1} + ... + \phi_p s_{t-p} + e_t \tag{2.5}$$

2.2.2.2 Modelo de médias móveis – MA(q)

Neste caso, o modelo assume que há uma dependência entre os valores futuros da série e os erros cometidos. Na equação (2.6), vemos que as diferenças frente ao modelo AR(p)

estão na ausência de valores passados da série e presença dos erros de predição: por exemplo, $e_{t-1} = \hat{s}_{t-1} - s_{t-1}$, onde $\hat{s}_{t-1}$ representa o valor estimado pelo modelo. Os coeficientes $\theta_1,...,\theta_q$ podem ser estimados via o algoritmo de inovação (*Innovations Algorithm*) (BROCKWELL & DAVIS, 1991).

$$s_t = e_t + \theta_1 e_{t-1} + ... + \theta_q e_{t-q} \tag{2.6}$$

2.2.2.3 Modelo ARMA(*p*,*q*)

O modelo ARMA combina ambas as propostas anteriores, resultando num preditor auto-regressivo com médias móveis. Neste caso, o modelo assume a forma indicada na equação (2.7). Dentre os algoritmos de ajuste dos coeficientes comumente utilizados temos Maximização da Verossimilhança (*Maximum Likelihood* - *ML*) e Quadrados Mínimos (BROCKWELL & DAVIS, 1991).

$$s_t = e_t + \phi_1 s_{t-1} + ... + \phi_p s_{t-p} + \theta_1 e_{t-1} + ... + \theta_q e_{t-q} \tag{2.7}$$

2.2.2.4 Modelo ARIMA(*p*,*d*,*q*)

Os modelos ARIMA(*p*,*d*,*q*) (*Auto-regressive Integrated Moving Average*) podem aproximar uma série temporal com raiz unitária. São uma generalização dos modelos ARMA, os quais unicamente representam séries estacionárias. Na abordagem ARIMA, depois de aplicar um processo de diferenciação da série em um número finito de vezes (*d*), reduz-se ao caso de modelos ARMA. Geralmente, basta realizar uma diferenciação (*d*=1) para tornar estacionárias séries com tendência linear, e duas (*d*=2), se a tendência for quadrática.

Séries temporais com componente sazonal são mais bem modeladas com modelos SARIMA (*Seasonal*-ARIMA) (BROCKWELL & DAVIS, 1991).

2.3 Predição de séries temporais via RNAs

As RNAs vêm sendo amplamente utilizadas em predição de séries temporais, principalmente por serem aproximadores universais. Interpreta-se o problema de síntese do preditor como aquele da síntese de um mapeamento não-linear de entrada-saída, sendo que o vetor de entrada será composto pelas informações consideradas relevantes para auxiliar no processo de predição. Logo, é possível empregar redes neurais artificiais para sintetizar modelos não-lineares que representam extensões daqueles apresentados nas sub-seções anteriores, resultando em modelos NAR, NMA, NARMA, onde o N inicial indica não-linearidade no modelo.

Na Figura 2.5, apresenta-se uma arquitetura de RNA que atende aos modelos NARMA(*p*,*q*), proposta por CONNOR & MARTIN (1994) e chamada de *rede NARMA recorrente*, onde os neurônios de entrada recebem: *p* entradas correspondentes a valores atrasados da série (a parte de AR(*p*)) e *q* entradas correspondentes aos erros de predição (a parte MA(*q*)).

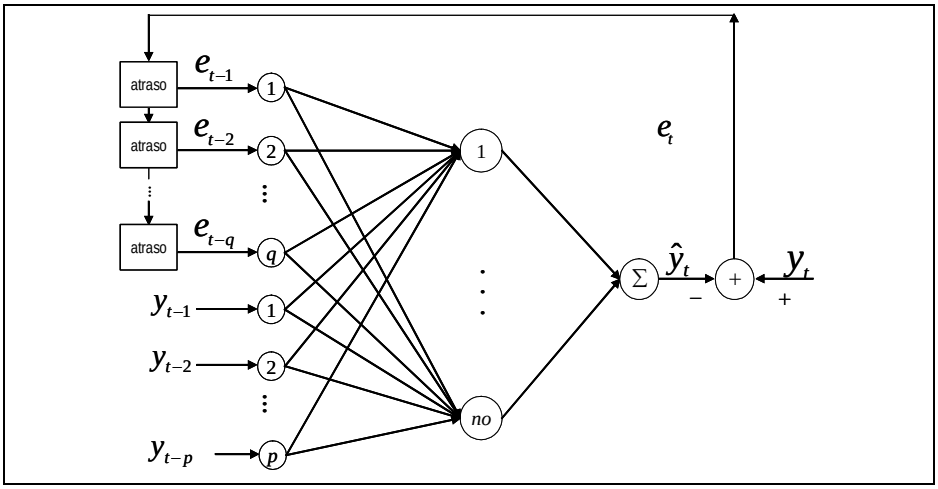


Figura 2.5 - Arquitetura da *rede NARMA recorrente*, proposta por CONNOR & MARTIN (1994).

CONNOR & MARTIN (1994) apresentaram um experimento com esta arquitetura com uma série temporal artificial do tipo NARMA(1,1): $y_t = e_t + 0.5 e_{t-1} y_{t-1}$, e compararam os resultados com aqueles produzidos por: (*i*) uma rede neural MLP implementando um

modelo NAR, e (*ii*) uma rede totalmente recorrente (recorrência total na camada oculta). Os resultados estão na Tabela 2.1, onde o mais relevante é notar que uma rede MLP também consegue modelar séries temporais do tipo NARMA, e a diferença perante as propostas de arquiteturas mais complexas, que envolvem recorrências, pode não ser significativa ou poderá depender do problema.

Tabela 2.1 - Resultados comparativos entre a rede NARMA recorrente de CONNOR & MARTIN (1994), com uma MLP e uma rede totalmente recorrente. Valores numéricos entre parênteses representam desvio padrão.

RNAs	*p*	*q*	No. Neurônios ocultos	EQM Treinamento	EQM Teste
MLP (NAR)	1	0	7	1,17	1,48 (0,028)
	2	0	8	1,11	1,35 (0,026)
	3	0	6	1,15	1,50 (0,030)
	4	0	4	1,14	1,45 (0,028)
	5	0	4	1,07	1,50 (0,029)
"NARMA Recorrente"	0	1	2	1,49	1,79 (0,035)
	1	1	5	1,11	1,28 (0,025)
Totalmente recorrente	1		5	1,03	1,11 (0,022)

Nesta dissertação, foram utilizados modelos de RNAs do tipo NAR, no caso, uma MLP, tanto nos resultados apresentados neste capítulo como no Capítulo 3, que trata de comitê de máquinas, onde o preditor é composto por mais de um modelo NAR. E no Capítulo 4, de igual forma, serão empregados modelos NAR para selecionar os atrasos mais relevantes para a predição das séries temporais.

2.3.1 Histórico do uso de RNAs em predição de séries temporais

Na literatura, o uso e o estudo de RNAs em predição de séries temporais podem ser descritos pela ocorrência de duas inquietudes maiores por parte de comunidade de redes neurais:

- RNAs são melhores opções do que os modelos tradicionais paramétricos?
- RNAs produzem melhores resultados ao se extraírem os componentes de tendência e sazonalidade da série temporal?

Atendendo a primeira questão, apresenta-se na Tabela 2.2 um acompanhamento de trabalhos de caráter comparativo entre RNAs e modelos tradicionais paramétricos empregados na predição de séries temporais. A coluna etiquetada como **Avaliação** indica com um símbolo "+" quando as RNAs se mostraram superiores, com "–" se inferiores, e "=" se tanto modelos paramétricos quanto RNAs tiveram desempenhos equivalentes na predição.

Tabela 2.2 - Acompanhamento de trabalhos comparativos entre RNAs e modelos tradicionais paramétricos empregados na predição de séries temporais.

Avaliação	Autor / ano	Séries testadas	Descrição
–	FISHWICK, 1989.	Séries de trajetória balística.	RNAs foram piores do que regressão linear e modelo resposta de superfície.
+ –	DULIBA, 1991.	Séries de transporte.	RNAs superam em séries com especificação de efeitos randômicos, não no fixo.
+	KANG, 1991.	50 séries de M-competition.	A melhor RNA sempre supera modelos Box & Jenkins, e melhora em relação ao horizonte de predição. RNAs necessitam menos pontos para igualar a predição oferecida por modelos ARIMA.
+ =	TANG *et al*., 1991.	3 séries negócio-econômicas mensais.	Em séries com memória de curto prazo, RNAs foram melhores e em séries com memória de longo prazo, RNAs foram equivalentes ao modelo ARIMA.
–	BRACE *et al*., 1991.	8 séries de carga elétrica diária.	RNAs não foram tão boas quanto modelos Box & Jenkins.
=	DE GROOT & Wurt, 1991.	Sunspots – anual.	RNAs foram comparáveis com métodos de Box & Jenkins e Holt-Winters.
+	CAIRE *et al*., 1992.	1 série de consumo elétrico diário.	RNAs melhor em 1 passo à frente e melhor ainda em vários passos à frente.
+	CHAKRABORTY *et al*., 1992.	1 série financeira mensal.	RNAs foram superiores a modelos de regressão linear.
+	WEIGEND *et al*., 1992.	Série Sunspot.	RNAs foram melhores do que TAR e modelos Bi-lineares.

+	WEIGEND *et al*., 1992.	Série de taxa de câmbio, diária.	RNAs foram significativamente melhores do que Random Walk.
–	FOSTER *et al*., 1992.	384 séries econômicas e demográficas trimestral e anual.	RNAs significativamente inferiores a regressão linear e a Exponential Smoothing Method.
=	MARQUEZ *et al*., 1992.	Série simulada por 3 métodos de regressão.	Desempenhos equivalentes.
=	SHARDA & PATIL, 1990; SHARDA & PATIL, 1992.	111 séries de M-competition, 75 séries mensais, trimestrais e anuais.	RNAs foram equivalentes a modelos de Box & Jenkins.
+	REFENES, 1993.	1 série de taxa de câmbio, horária.	RNAs muito melhor do que Exponential Smoothing e ARIMA.
+	TANG & FISHWICK, 1993.	14 séries de M-competition, 2 séries de negócios mensais & trimestrais.	RNAs se mostram melhores, e ainda mais quando o horizonte de predição aumenta.
+	SRINIVASAN *et al*., 1994.	1 série de carga elétrica.	RNAs foram melhores do que modelos de regressão e ARMA.
=	GORR *et al*., 1994.	Séries pontuações médias de estudantes de graduação.	A melhora não é significativa com respeito à regressão linear.
+ =	DENTON, 1995.	Várias séries sintéticas.	RNAs superiores em condições ideais e, caso contrário, equivalentes.
+	LACHTERMACHER & FULLER, 1995.	4 séries estacionárias de vazamento de rio e 4 séries não-estacionárias de carga elétrica anual.	Nas séries estacionárias, as RNAs foram ligeiramente superiores e nas séries não-estacionárias, sempre foram muito melhor do que ARIMA.
+	NAM & SCHAEFER, 1995.	Série Airpass mensal.	RNAs melhores do que regressão e Exponential Smoothing.
+ =	HANN & STEURER, 1996.	Série de taxa de câmbio semanal e mensal.	Supera a regressão linear nas séries semanais, com desempenho equivalente para as séries mensais.
+	KOHZADI *et al*., 1996.	Séries de preço de gado e trigo mensal.	RNAs considerável e consistentemente melhores.

No intuito de resumir os resultados da avaliação, a Figura 2.6 apresenta um diagrama de Venn de onde podemos concluir que, dos 22 casos comparativos, em 15 casos as RNAs se mostraram superiores e em 4 inferiores. As interseções são entendidas da seguinte forma:

dos 7 casos em que os resultados foram equivalentes, em 3 deles as RNAs foram superiores em alguns pontos específicos, os quais são detalhados na coluna de “Descrição” da Tabela 2.2 e nos outros 4 casos os resultados foram equivalentes.

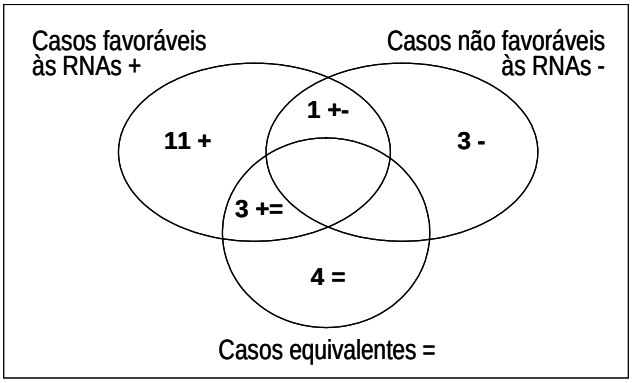


Figura 2.6 - Resumo pelo número de casos favoráveis, não-favoráveis e equivalentes das RNAs frente aos modelos tradicionais.

Curiosamente, a primeira inquietude da comunidade de RNAs deu origem a uma outra: se as RNAs produzem melhores resultados ao extrair os componentes de tendência e sazonalidade. Nos estudos comparativos citados anteriormente, muitos deles retiravam os componentes de tendência e sazonalidade da série. Tentando resolver esta nova inquietude, foram realizados vários trabalhos comparativos, dos quais destacam-se NELSON *et al.* (1999) e ZHANG & QI (2005). Eles concluíram que as RNAs na prática conseguem melhorar seus desempenhos ao serem retirados da série temporal os componentes de tendência e sazonalidade. Em adição, em muitos dos estudos comparativos utilizaram-se séries curtas da competição-M (*M-competition*) (MAKRIDAKIS *et al.*, 1982) e o risco de sobreajuste compromete a propriedade de aproximação universal. Resumindo, o uso de RNAs exige cuidados na qualidade e quantidade dos dados, acondicionamento dos dados e seleção do modelo no processo de treinamento. Estes cuidados valem também para a modelagem clássica de séries temporais, por exemplo, a de Box & Jenkins (BOX *et al.*, 1994).

2.3.2 Tratamento dos dados para o treinamento

Nesta seção, descrevem-se as técnicas de pré-processamento dos dados da série temporal ao usar RNAs, no caso uma MLP. Os procedimentos são basicamente dois: (*i*) o acondicionamento dos valores da série temporal; e (*ii*) a construção dos conjuntos de treinamento, validação e teste.

2.3.2.1 Acondicionamento

Como o modelo de RNA escolhido foi uma MLP com função de ativação tangente hiperbólica, é recomendado que os valores da série excursionem dentro da faixa entre −1 e +1, reduzindo as chances de produzir neurônios saturados durante o treinamento da rede neural. Trata-se apenas de uma mudança na escala dos valores da série temporal original, de modo que possíveis componentes como tendência e/ou sazonalidade não sejam alterados. É preciso reverter esta mudança de escala para calcular o erro de predição da série na escala original da mesma. É comum utilizar outras formas de normalização baseadas no desvio padrão e no logaritmo dos valores da série temporal, no primeiro caso, gera-se uma nova série temporal onde cada novo valor (s_i^{novo}) desta série é calculado da seguinte forma:

$$s_i^{novo} = (s_i - \bar{s}) / \sigma_s , \tag{2.8}$$

sendo $\bar{s}$ e σ_s o valor médio e o desvio padrão da série temporal original. Por outra parte, a normalização via extração do logaritmo da série temporal é particularmente empregada em séries temporais financeiras.

2.3.2.2 Construção dos conjuntos de treinamento

Como a série temporal já foi acondicionada na etapa anterior, então resta montar os conjuntos de treinamento, validação e teste. Como primeira etapa, devem-se preparar os

dados para realizar predição de valores futuros, por exemplo, um valor à frente. A Figura 2.7 descreve esta etapa, onde o valor de L é calculado usando a função de auto-correlação – ACF (BOX *et al.*, 1994) ou via Informação Mútua – MI (CELLUCCI *et al.*, 2005). No Anexo-A, apresenta-se o tratamento das séries com essas funções que conduzem a uma estimativa do valor de L. Uma vez definido o valor de L, o número de possíveis subconjuntos de atrasos (colunas de **X** para sintetizar um preditor) apresenta crescimento exponencial e obedece à fórmula 2^L-1. Isto será tratado de forma mais detalhada no Capítulo 4 de Seleção de Variáveis.

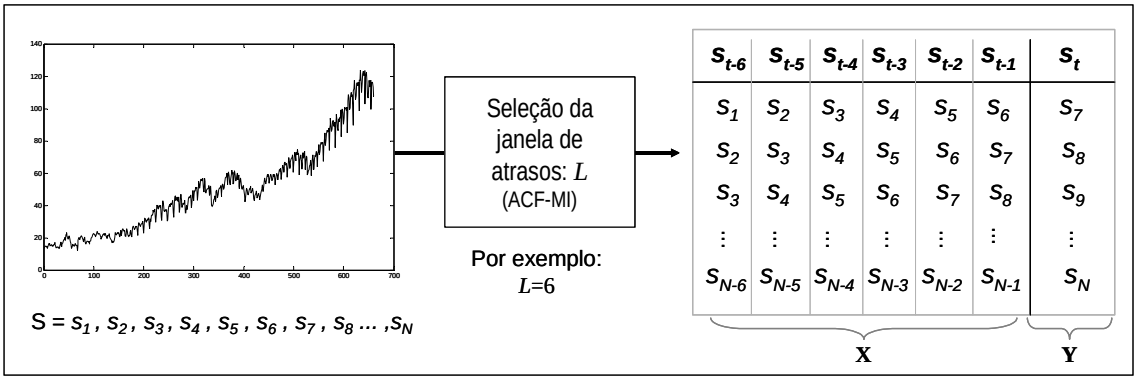


Figura 2.7 - Preparação dos dados para predição de um passo à frente com L=6 valores atrasados da série.

Da Figura 2.7 desprendem-se as clássicas matrizes **X** e **Y** que representam os padrões de treinamento da RNA, onde cada valor de **Y** é a saída desejada associada à sua correspondente linha da matriz **X**. Na Figura 2.8, apresentam-se duas formas de separar os conjuntos de treinamento, validação e teste. O conjunto de teste geralmente é tomado como uma parcela dos últimos valores contíguos da série temporal. Para os conjuntos de treinamento e validação, dadas as porcentagens previamente estabelecidas para cada um, a priori, pode haver duas opções: (*i*) separar de forma seqüencial ou (*ii*) separar de forma randômica. Em PUMA-VILLANUEVA *et al.* (2006), mostra-se que melhores resultados podem ser obtidos pela separação randômica. A separação randômica é sempre possível tomando-se aleatoriamente, e com distribuição de probabilidade uniforme, linhas da tabela à direita na Figura 2.7.

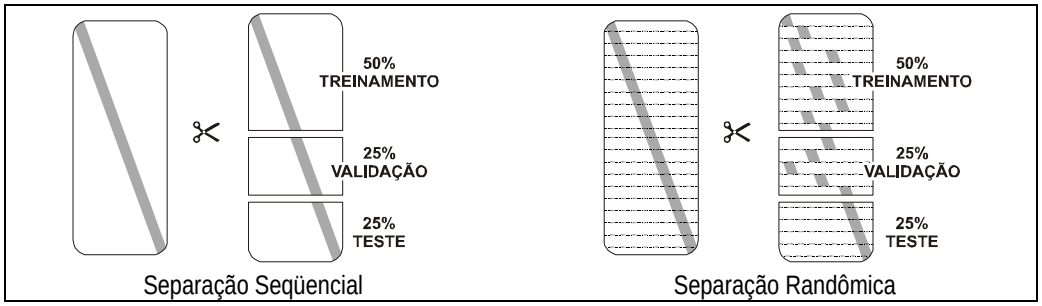


Figura 2.8 - Formas de separar os dados para compor os conjuntos de treinamento, validação e teste, tomando por base a Tabela à direita na Figura 2.7. As porcentagens 50%, 25% e 25% representam apenas uma sugestão.

2.3.3 Simulações e resultados

Como uma etapa preliminar deste trabalho, realizaram-se modelagem e predição de cinco séries temporais reais via uma RNA do tipo MLP. No Capítulo 3, estes resultados serão contrastados com os obtidos via abordagens de Comitê de Máquinas (*Ensemble* e Mistura de Especialistas) no Capítulo 3.

As quatro primeiras séries contam com 120 observações e a última conta com 660. São todas de natureza financeira de valores mensais médios de: venda de livros (*Book store*), roupa (*Clothing store*), móveis (*Furniture store*), hardware (*hardware store*) e a última de bens de consumo duráveis (*Durable consumer goods*).

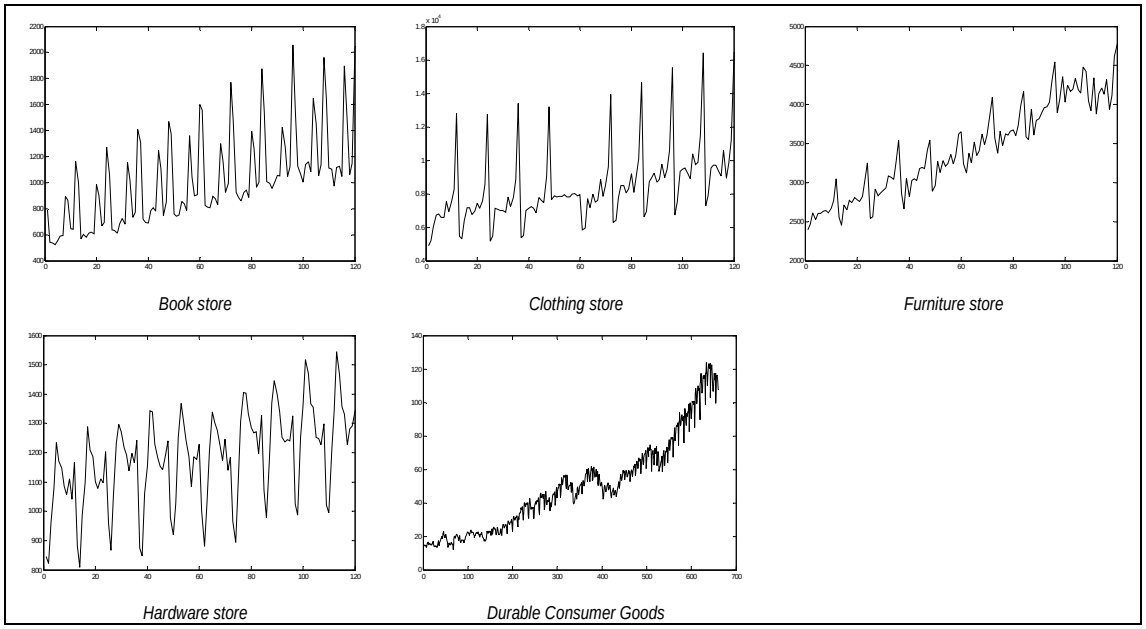


Figura 2.9 - Séries temporais testadas.

Estas mesmas séries foram utilizadas por ZHANG & QI (2005), que procuraram mostrar que realizando um pré-processamento prévio da série temporal consegue-se melhorar o erro de predição. Os resultados obtidos em ZHANG & QI (2005) foram adicionados aos obtidos nesta dissertação via uma rede neural MLP, para fins apenas informativos e não comparativos, dado que a configuração de parâmetros não é a mesma. Isto será explicado em detalhe na seção seguinte.

2.3.3.1 Configuração dos parâmetros

São muitos os parâmetros que devem ser definidos no caso de predição via RNAs, com alguns deles dependentes do problema de predição específico. Segue uma descrição da configuração de todos os parâmetros empregados:

- *Série temporal*: o único pré-processamento realizado sobre a série temporal foi um simples escalonamento linear das observações para a faixa entre –1 e +1, visando acondicionar os dados de treinamento.

- *A construção dos conjuntos de treinamento*: o número de atrasos empregados foi doze (L=12) e são consecutivos, obtido com auxílio de Informação Mútua (ver Anexo-A). A separação dos conjuntos de *treinamento*, *validação* e *teste* obedeceu as porcentagens de 50, 25 e 25%, respectivamente. Os dois primeiros conjuntos foram definidos de forma *randômica* e o conjunto de *teste* foi formado pelas últimas amostras de cada série, tal como ilustrado na Figura 2.8.
- *MLP*: O número de neurônios na camada de entrada coincide com os atrasos da série, já os ocultos foram fixados em 30 e cada um deles com função de ativação tangente hiperbólica. Um único neurônio compõe a camada de saída, por se tratar de predição um passo à frente com função de ativação linear. O treinamento segue os delineamentos da Seção 2.1.4, empregando o método do gradiente para ajuste dos pesos da rede. Esses pesos foram inicializados com valores reais aleatórios no intervalo [-0,01; +0,01]. Aplicou-se validação cruzada como forma de seleção do modelo visando evitar sobre-ajuste, o que corresponde a uma estratégia de parada antecipada para deter o treinamento antes das 500 épocas fixadas, caso a rede já tenha convergido.

2.3.3.2 Resultados obtidos

A Tabela 2.3 apresenta os resultados obtidos na coluna 3 usando o índice de erro MAE, os resultados publicados por ZHANG & QI (2005) nas últimas 5 colunas para estas mesmas séries temporais. E visando calcular estatísticas como média e desvio padrão (*STD-Standard Deviation*), máximo, mínimo, foram realizadas 30 execuções da MLP.

O objetivo de ZHANG & QI (2005) era provar que, realizando um pré-processamento mais elaborado da série temporal antes de ser modelada por uma RNA (MLP), podia-se conseguir ganhos significativos na predição. Assim, utilizou técnicas para retirar a tendência e a sazonalidade da série. Os resultados consideram o conjunto de teste, e o valor que aparece na tabela corresponde à medida de erro MAE (erro absoluto médio).

É importante ter em consideração os parâmetros utilizados por ZHANG & QI (2005):

- *Série temporal*: como já indicado, empregaram-se técnicas de extração da tendência e/ou sazonalidade (X-12-ARIMA).
- *A construção dos conjuntos de treinamento*: o número de atrasos empregados foram diferentes, mas para o caso em que se tratou a série original sem pré-processamento, utilizaram os seguintes: *t*-1, ..., *t*-4, *t*-12, *t*-13, *t*-14, *t*-24, *t*-25 e *t*-36. A separação dos conjuntos de *treinamento*, *validação* e *teste* foi feito de modo seqüencial, sendo que as últimas 12 observações para *teste*, as 12 antecedentes do *teste* para *validação* e o restante para *treinamento*.
- MLP: os neurônios na camada oculta não foram mais que 14, a função de ativação dos neurônios da camada oculta foi a *logística*. O algoritmo de ajuste dos parâmetros empregou informação de segunda ordem: *fast Levenberg & Marquardt*. Empregaram validação cruzada para selecionar o modelo tendo como máximo 1000 épocas de treinamento. Foram realizadas 5 execuções do algoritmo partindo de condições iniciais distintas para os pesos da rede neural. Em seguida, foi escolhida aquela com menor erro de validação.

Pelo descrito anteriormente, é impossível tentar comparar resultados de ambas as propostas, porque o conjunto de teste, embora corresponda às últimas observações da série, não foram iguais: enquanto ZHANG & QI (2005) usaram apenas os 12 últimos pontos das séries temporais, nesta dissertação foi empregado 25% do total, 29 pontos para as 4 primeiras séries e 165 para a última série. As dimensões das redes neurais e as funções de ativação também foram diferentes em ambos os trabalhos, assim como a quantidade de amostras do subconjunto de validação, também foram diferentes em ambos os trabalhos.

Na coluna 4 da Tabela 2.3, é apresentado o erro de predição MAE utilizando uma rede MLP com as séries originais, na coluna 5 a MLP opera com as séries apenas sem tendência, na coluna 6 unicamente a sazonalidade foi retirada, e na coluna 7 ambas foram retiradas, a tendência e a sazonalidade das séries temporais. Na última coluna, é usado um modelo ARIMA com as séries temporais originais (sem retirar tendência, nem sazonalidade).

Tabela 2.3 - Resultados de erro de predição MAE a partir do conjunto de teste das 5 séries temporais, inclui-se resultados de ZHANG & QI (2005).

SÉRIE	Estatísticas	MLP	Resultados de Zhang & Qi (2005) para as mesmas séries temporais				
			MLP - série original	MLP - série com tendêndecia retirada	MLP - série com sazonalidade retirada	MLP - série com tendêndecia e sazonalidade retiradas	ARIMA
Book store	**Média**	**84,82**	**305,61**	**105,96**	**85,06**	**43,23**	**51,06**
	STD	14,95					
	Max	121,85					
	Min	69,08					
Clothing store	**Média**	**781,35**	**1662,36**	**691,46**	**866,82**	**206,52**	**405,53**
	STD	437,18					
	Max	1767,90					
	Min	477,71					
Furniture store	**Média**	**176,44**	**391,00**	**193,12**	**109,46**	**76,23**	**103,82**
	STD	32,77					
	Max	310,36					
	Min	155,78					
Hardware store	**Média**	**42,68**	**167,30**	**84,55**	**131,33**	**41,24**	**89,40**
	STD	8,89					
	Max	73,92					
	Min	34,12					
Durable Consumer Goods	**Média**	**4,11**	**4,89**	**4,52**	**2,95**	**2,66**	**4,47**
	STD	0,32					
	Max	4,74					
	Min	3,57					

Capítulo 3

Comitê de Máquinas

Resumo: Neste capítulo, apresentam-se duas abordagens de comitê de máquinas que visam produzir ganhos de desempenho frente a propostas de soluções isoladas, ou seja, que não formam comitês. Essas abordagens são *ensemble* e mistura de especialistas (MEs), ambas guiadas pelo princípio dividir-para-conquistar. Um comitê de máquinas é uma forma de aprendizado de máquina que pode ser aplicada tanto para problemas de classificação quanto regressão, sendo que neste capítulo será considerado o problema de predição de séries temporais, um problema típico de regressão, mas que pode envolver etapas de classificação. Cada abordagem será apresentada em detalhe e seus resultados serão contrastados com os resultados obtidos por um único modelo preditor, no caso, uma rede neural MLP.

3.1 Introdução

Realizando uma breve revisão de fatos importantes ao longo da história de "aprendizado de máquina", cabe destacar quatro eventos, sendo que no último surgiu a idéia de comitê de máquinas:

- ***Evento 1*:** Ao final dos anos 1950, iniciou-se a construção da primeira máquina de aprendizado, conhecida como "*Perceptron* de Rosenblatt" (ROSENBLATT, 1958). Alguns anos depois, foi provado que esta máquina possuía algumas limitações teóricas (MINSKY & PAPERT, 1969). No entanto, a proposta do perceptron não deixa de ser relevante, pois marca o início de uma das vertentes na linha de Inteligência Artificial: o aprendizado de máquina.

- ***Evento 2***: No final dos anos 1960 e início dos anos 1970, foram elaborados os fundamentos básicos da Teoria do Aprendizado. Esses fundamentos são ainda hoje empregados no desenvolvimento e/ou aprimoramento de máquinas de aprendizado (VAPNIK & CHERVONENKIS, 1971).
- ***Evento 3***: Nos anos 1980, as limitações da proposta de Rosenblatt foram superadas com a contribuição de RUMELHART & MCCLELLAND (1986) para o treinamento de redes MLP (*Multi-layered Perceptrons*), empregando o algoritmo de retro-propagação do erro ou *backpropagation* (WERBOS, 1974). A partir de então, foram propostas várias outras arquiteturas de redes neurais artificiais (RNAs) e surgiram vários algoritmos de treinamento para RNAs com o objetivo de atender a problemas específicos.
- ***Evento 4***: Nos anos 1990, em busca de uma maior maturidade conceitual, foram propostas formas alternativas às RNAs, visando superar algumas dificuldade identificadas. Assim, surgiram quase que em paralelo as propostas de comitês de máquinas (HAYKIN, 1999) e máquinas de vetores-suporte – SVM (VAPNIK, 1998).

A partir de então, houve uma mudança de paradigma na área de aprendizado de máquina. SVMs (do inglês *support vector machines*) e outros métodos baseados em *kernel* são métodos estatísticos que buscam maximizar a capacidade de generalização do modelo preditor. Por outra parte, comitês de máquinas buscam agregar, de alguma forma, o conhecimento adquirido pelos modelos que o compõem para chegar a uma solução global que apresente melhor desempenho frente à obtida por qualquer um dos componentes atuando isoladamente. As propostas para comitês de máquinas são *ensemble* e mistura de especialistas (MEs).

No decorrer deste capítulo, serão feitas distinções de nomenclatura para os comitês de máquinas: os modelos preditores que compõem um *ensemble* serão chamados de componentes, e os modelos preditores que compõem uma mistura de especialistas serão chamados de especialistas. Nas próximas seções, será apresentada uma visão geral de *ensemble* e de mistura de especialistas, visando destacar as diferenças que existem entre eles e a sua aplicabilidade ao problema de predição de séries temporais.

3.1.1 Razões que levaram ao surgimento de comitê de máquinas

Se conseguíssemos obter um modelo com máxima capacidade de generalização, não haveria lugar para a técnica de comitê de máquinas. No entanto, isto não é possível, e dentre as causas esta o fato de que as amostras de treinamento utilizadas para o ajuste do modelo são em número finito e estão sujeitas a ruído e *outliers*.

No contexto de RNAs, as razões que levaram ao estudo e à implementação de comitê de máquinas estão diretamente relacionadas aos aspectos desfavoráveis que as RNA's apresentam (BISHOP, 1995):

- **Convergência do algoritmo de treinamento para um mínimo local**: a forma da função-objetivo que guia o processo de ajuste dos pesos sinápticos das RNAs na maioria dos casos é não-linear e multimodal. Adicionalmente, a dependência entre os estímulos de entrada ou entrada-saída, associados ao problema, faz parte também desta função-objetivo, em conseqüência o risco de cair num mínimo local não-desejado é não-desprezível.
- **Risco de sobre-ajuste ou sub-ajuste**: como apresentado no Capítulo 2, o risco do treinamento da RNA levar a um sobre-ajuste junto ao conjunto de treinamento é tratado por técnicas empíricas, as quais não garantem o melhor desempenho de generalização. Se existe sobre-ajuste, então pode existir sub-ajuste, que é quando o treinamento é concluído antes da RNA atingir um nível satisfatório de aproximação do mapeamento de entrada-saída associado ao problema. Se um ou ambos os casos acontecem, comete-se um *erro de estimação*.
- **Complexidade do problema a ser solucionado**: embora as RNAs sejam consideradas aproximadores universais, o seu desempenho efetivo vai depender fortemente de dois aspectos. O primeiro está relacionado ao erro de estimação visto no item anterior. O segundo aspecto tem a ver com a estrutura da RNA. A eleição de um número adequado de neurônios ocultos é fundamental para garantir que hipoteticamente a RNA conte com a flexibilidade necessária para modelar o problema. É evidente que, dependendo do tipo de mapeamento a ser aproximado, um

número maior ou menor de neurônios ocultos será requisitado. Caso o modelo não satisfaça esta hipótese de flexibilidade compatível com a complexidade inerente do problema, comete-se um *erro de aproximação*.

O sucesso da generalização da RNA vai depender de ambos os erros: estimação e aproximação. Com base nos aspectos levantados acima, se justifica a busca de alternativas para contornar as dificuldades das RNAs, incluindo os comitês de máquinas.

A utilização de comitês de máquinas pode trazer os seguintes benefícios (HAYKIN, 1999; BISHOP, 1995):

- Redução na variância do modelo final;
- Melhora na tolerância a ruído nos dados;
- Melhora na capacidade de generalização.

Além disso, pode existir uma outra vantagem prática: cada componente do ensemble e cada especialista da mistura não precisam necessariamente apresentar uma flexibilidade compatível com a complexidade do problema, visto que não serão considerados isoladamente, mas sim em comitê.

3.1.2 Estruturas: Estática e Dinâmica

HAYKIN (1999) propôs uma classificação de comitês de máquinas em estruturas estáticas e dinâmicas. *Na estrutura estática*, colocou a abordagem *ensemble*, onde os componentes são treinados para resolver todo o problema de forma separada e, possivelmente, independente. Neste caso, cada componente representa isoladamente uma solução candidata para o problema, seja de classificação, regressão e até de agrupamento, podendo cada solução ter sido obtida por meios distintos e independentes entre si. A Figura 3.1(a) apresenta uma estrutura típica de um *ensemble*.

Mistura de especialistas foi colocada como *estrutura dinâmica*, porque a tarefa é decomposta em várias sub-tarefas, e a solução final da tarefa requer a contribuição de todos

ou um subconjunto dos especialistas. Logo, um especialista isolado não representa uma solução candidata para o problema como um todo, sendo fundamental a composição dos especialistas. Esta composição é realizada via a rede *gating*, a qual aprende, de acordo com as características individuais de cada componente, a alocar dinamicamente uma determinada área do espaço de entrada, durante o treinamento. A Figura 3.1(b) apresenta uma estrutura típica de mistura de especialistas.

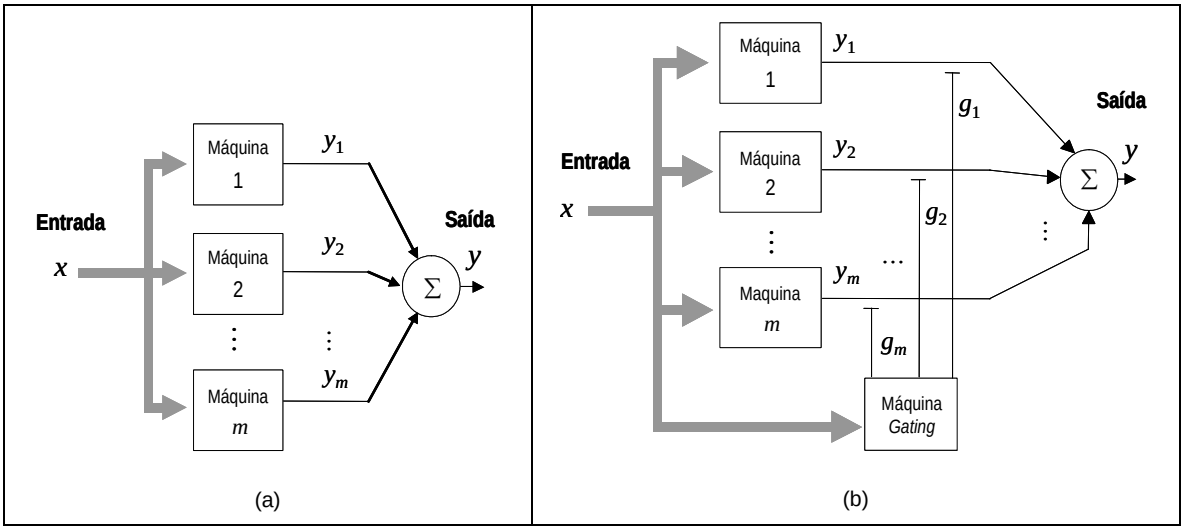


Figura 3.1: Propostas de Comitês de Máquinas, (a) *ensemble* e (b) mistura de especialistas.

3.2 *Ensemble*

Ensemble é a combinação das saídas de dois ou mais componentes, os quais resolvem o total de uma determinada tarefa de forma isolada, mas não necessariamente independente. Podem ser utilizados em aprendizado supervisionado, como, por exemplo, regressão e classificação de dados, ou em aprendizado não-supervisionado, como agrupamento de dados.

Trata-se de uma das principais linhas de pesquisa em aprendizado de máquina (DIETTERICH, 2000; KITTER *et al.*, 1998; KUNCHEVA, 2004). Algumas contribuições que buscam explicar o seu comportamento e características são: ALLEWEIN *et al.* (2000)

interpretou a capacidade de generalização no contexto de classificadores de grande margem, KLEINBERG (2000) no contexto da Teoria de Discriminação Estocástica e BREIMAN (2000) no contexto de análise bias-variância, utilizando ferramentas clássicas de estatística.

Um dos requisitos fundamentais para o sucesso ao utilizar *ensemble* é que os componentes generalizem de forma diferente. Não faz sentido combinar modelos que adotam os mesmos procedimentos e hipóteses para a solução de um problema. É necessário que os erros cometidos pelos componentes sejam descorrelacionados (PERRONE & COOPER, 1993).

Um exemplo que pode ajudar a entender a idéia de agrupar e combinar, as quais são as idéias fundamentais por trás de um *ensemble*, seria o seguinte: "Transmissão de um bit por um canal de comunicações":

Considere, um canal de comunicação em que a probabilidade de fracassar ao enviar um bit é $P(fracasso) = 10^{-3}$, em conseqüência $P(sucesso) = 1\text{-}10^{-3}$ (ver Figura 3.2). Visando diminuir a probabilidade de fracasso, assume-se uma nova política de envio, cada bit será enviado 3 vezes e, na chegada, será aplicada a estratégia de "voto majoritário" para combinar os resultados dos 3 envios e saber o valor do bit resultante. Assim, os envios 000, 001, 010, e 100 significam 0, e os envios 111, 110, 101, e 011 significam 1. A pergunta é: ante a nova política de envio-recepção, qual é a nova probabilidade de fracasso, $P(fracasso)$?

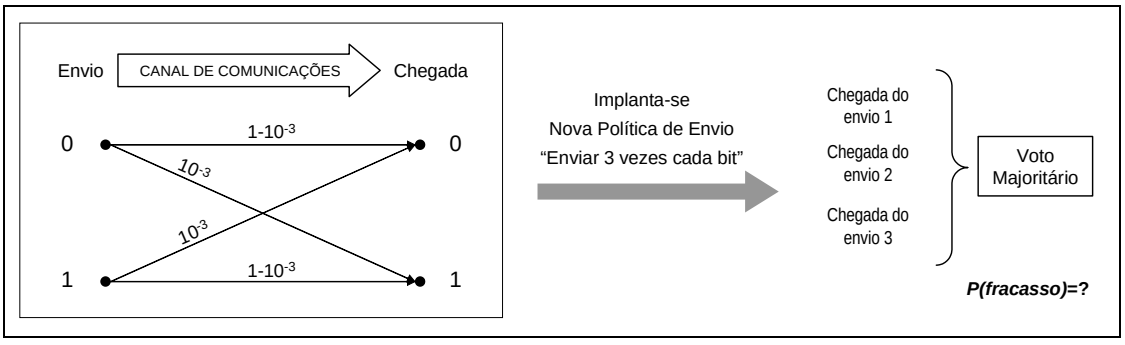


Figura 3.2 – Exemplo didático, aplicação do conceito de *ensemble* para reduzir a probabilidade de fracasso num canal de comunicações.

A nova probabilidade de fracasso *P*(*fracasso*) = *P*(acontecerem 2 ou mais fracassos parciais), quer dizer, considera-se que houve fracasso se, após de envio dos 3 bits, em 2 ou em 3 envios houve fracasso ($P_3(2), P_3(3)$ respectivamente). Utilizando expansão binomial, pode-se calcular a nova probabilidade de fracasso:

$$P(\textit{fracasso}) = P_3(2) + P_3(3) = \binom{3}{2}(10^{-3})^2(1-10^{-3}) + \binom{3}{3}(10^{-3})^3(1-10^{-3})^0 \cong 3\times10^{-6}$$

Do exemplo anterior conclui-se que, repetindo 3 vezes um experimento que envolve certo grau de incerteza, pode-se diminuir a probabilidade de fracasso de 10^{-3} para aproximadamente 3×10^{-6}, obtendo um consenso final a partir do voto majoritário envolvendo os resultados dos 3 experimentos.

3.2.1 Motivações para usar *ensemble*

No exemplo anterior, a transmissão de um bit por um canal de comunicação mostrou os benefícios advindos da combinação de respostas individuais. No caso, utilizou-se voto majoritário. Surge então a questão: quando os componentes são máquinas de aprendizado, por exemplo, RNAs, será que combinar seus resultados traz ganhos de desempenho, como a diminuição do erro de generalização? As seguintes motivações buscam sustentar uma resposta afirmativa para esta questão.

- **Motivações estatísticas**: no contexto de RNAs, considere alguns aspectos relacionados ao treinamento e à seleção do modelos: (*i*) obter um baixo valor do erro para o conjunto de treinamento não implica necessariamente que o erro de generalização seja baixo também; (*ii*) componentes com índice de desempenho similar no treinamento (por exemplo, valor acumulado do erro de treinamento) podem ter desempenhos de generalização diferentes; (*iii*) se o conjunto de validação usado para selecionar o modelo (parar o treinamento antes de acontecer sobre-ajuste) não for suficientemente representativo, aumenta a possibilidade de que a generalização não seja bem sucedida. Esses aspectos estão presentes também em modelos estatísticos,

não se restringindo ao caso de redes neurais artificiais. Por exemplo, em predição de séries temporais, os primeiros trabalhos a defenderem que a combinação de modelos pode trazer ganhos na predição foram BATES & GRANGER (1969) e NEWBOLD & GRANGER (1974), ao combinar métodos de Box & Jenkins, Holt-Winters e auto-regressivos por partes. O simples exemplo de transmitir um bit por um canal de comunicações, citado na seção anterior, se encaixa nesta motivação.

- **Grande volume de dados**: em algumas aplicações, a quantidade de dados a ser analisada pode ser de grande porte e um único componente pode não modelar o total do problema de forma eficiente, levando à ocorrência de *erro de estimação*. Usualmente, pode-se particionar os dados em subconjuntos menores, e ajustar componentes usando cada um dos subconjuntos por separado, seguido da combinação de suas saídas. Esta forma tende a ser uma abordagem mais efetiva.
- **Pequeno volume de dados**: a qualidade e a quantidade de dados são fatores decisivos para o desempenho de máquinas de aprendizado. Na ausência de uma amostragem adequada para treinamento, técnicas de re-amostragem podem ser usadas para gerar subconjuntos aleatórios com sobreposição dos dados disponíveis. Esses subconjuntos parciais podem ser usados para treinarem componentes, criando finalmente um *ensemble*.
- **Complexidade do problema**: independente da quantidade de dados disponível, alguns problemas são muito difíceis de serem tratados por apenas uma componente. No caso de classificação, a fronteira de decisão que separa os dados de classes diferentes pode ser muito complexa e difícil de reproduzir via um único modelo. A combinação de modelos isolados pode possibilitar uma melhor aproximação. As Figuras 3.3 e 3.4 ilustram graficamente este aspecto.
- **Custo Computacional**: é melhor computacionalmente treinar vários modelos simples do que um único modelo maior e com muitos parâmetros a serem ajustados. Visto de outra forma, o custo computacional para escolher uma RNA (testar diversas arquiteturas, algoritmos de treinamento que também têm parâmetros próprios e limiares, etc) pode ser mais alto que aquele associado a uma abordagem de *ensemble*.

- **Fusão de dados**: a fusão ocorre quando dados provenientes de diferentes fontes de entrada são combinados para produzir uma única saída. Sendo assim, os princípios de operação são similares e as vantagens apontadas para técnicas de fusão de dados (HALL & LLINAS, 2001) podem, em princípio, ser estendidas para *ensembles*.

Há muitos outros cenários nos quais a utilização de *ensemble* pode ser muito benéfica. Entretanto, a discussão desses cenários mais específicos requer um profundo entendimento das condições que devem estar presentes para que uma proposta de *ensemble* seja bem sucedida. Essas condições podem não ser triviais de identificar a priori, e uma sugestão prática é aplicar *ensemble* e comparar o desempenho obtido com aquele produzido por soluções isoladas.

3.2.2 Desempenho esperado para um e*nsemble*

Antes de apresentar o formalismo por trás de um *ensemble*, seguem dois exemplos didáticos que visam ilustrar o ganho em desempenho esperado, associado a problemas de predição de séries temporais e classificação de padrões.

Na Figura 3.3, ilustra-se o caso do problema de predição de séries temporais, supondo os componentes na forma de RNAs. Na Figura 3.3, há 3 componentes com desempenhos diferentes para o conjunto de teste (predição). Olhando mais no detalhe, podemos perceber que os dois primeiros conseguiram aproximar melhor os maiores picos da série, e que o último teve melhor desempenho nos valores baixos. Se combinarmos essas predições utilizando a média simples, obteremos o resultado apresentado ao lado direito da Figura 3.3. Conforme pode ser observado, o resultado obtido via *ensemble* apresenta um desempenho superior a qualquer um dos componentes tomados de forma isolada.

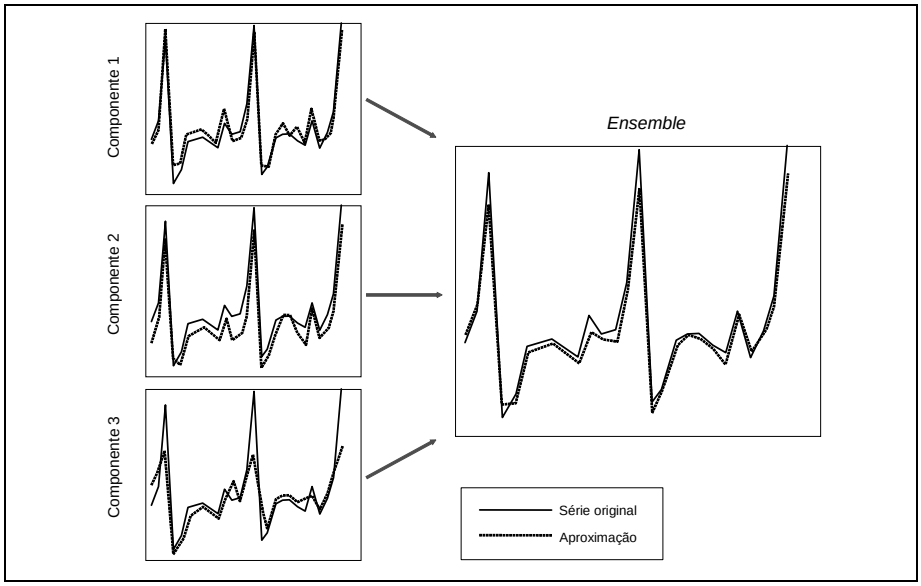


Figura 3.3 – Exemplo ilustrativo de ganho de desempenho empregando *ensemble* em predição de séries temporais.

Na Figura 3.4, é apresentado, com fins didáticos (POLIKAR, 2006), um problema de classificação com 3 classes. Também foram utilizados 3 componentes e as fronteiras de decisão de cada um são apresentadas nas Figuras 3.4(a)(b)(c). Na Figura 3.4(d), todas as 3 fronteiras de decisão são apresentadas simultaneamente. Na figura 3.4(e), o resultado da combinação, utilizando voto majoritário, é apresentado. Observe que, ao realizar a combinação, obtivemos 100% de classificação correta, enquanto todos os componentes apresentavam algum erro de classificação. É evidente que o que se espera na prática é apenas um ganho de desempenho frente ao melhor classificador disponível, quando tomado isoladamente, o que não necessariamente produzirá taxas de acerto elevadas (próximas de 100%).

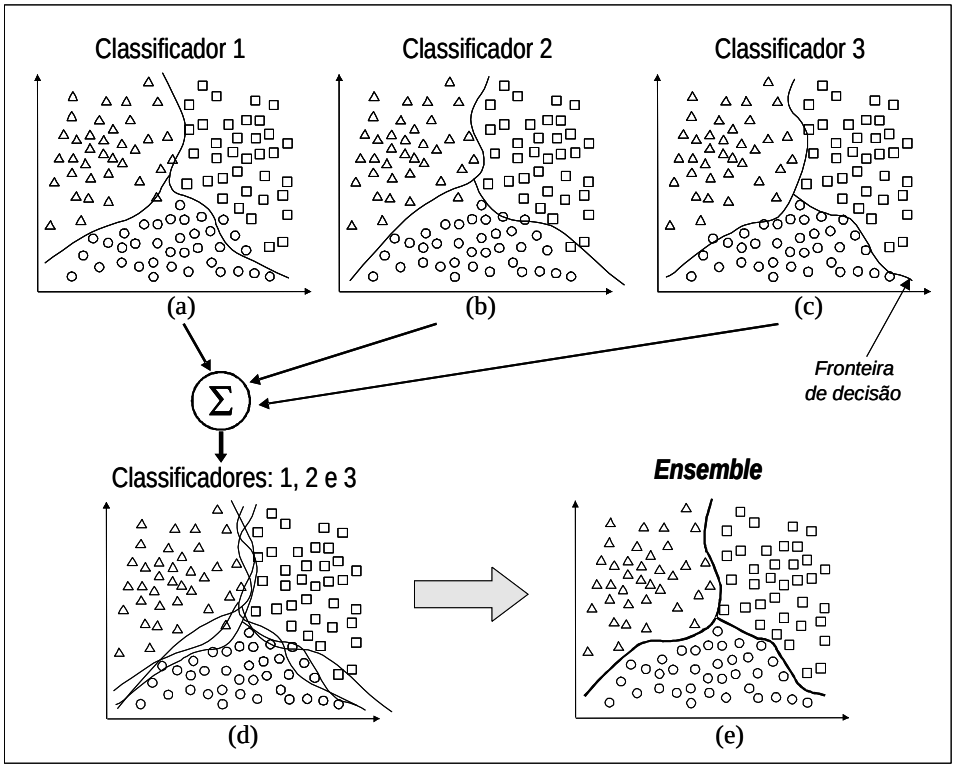


Figura 3.4 Exemplo ilustrativo do ganho de desempenho de um ensemble em problemas de classificação com múltiplas classes (POLIKAR, 2006).

Agora, um tanto mais formal e para indicar que o desempenho do ensemble pode ser superior ao do melhor componente atuando de forma isolada, apresenta-se o seguinte fundamento teórico (BISHOP, 1995):

Considere que $y_i(\mathbf{x})$ representa a saída do componente i, para o padrão de treinamento $\mathbf{x}\in\mathbf{X}$, onde $\mathbf{X}$ é o espaço de entrada e i=1,2,3,...,M, sendo M o número de componentes treinados. Suponha que $f(\cdot):\mathbf{X}\rightarrow\Re$ seja uma função desconhecida que se deseja aproximar. Pode-se escrever a função de mapeamento que cada componente realizará da seguinte forma:

$$y_i(\mathbf{x}) = f(\mathbf{x}) + e_i(\mathbf{x}),\ \mathbf{x}\in\mathbf{X}. \tag{3.1}$$

O erro quadrático médio para o componente i (EQM_i) pode ser calculado da seguinte forma:

$$EQM_i = E\left[\{y_i(\mathbf{x}) - f(\mathbf{x})\}^2\right] = E\left[e_i^2(\mathbf{x})\right], \tag{3.2}$$

onde $E[\ ]$ representa a esperança matemática, e corresponde a uma integral sobre **x** ponderada pela densidade de **x** tal que:

$$E\left[e_i^2\right] \equiv \int e_i^2(\mathbf{x}) p(\mathbf{x}) d\mathrm{x} \,. \tag{3.3}$$

A partir da equação (3.2), o erro quadrático médio cometido pelos *M* componentes (EQM_{me}) atuando individualmente é:

$$EQM_{me} = \frac{1}{M}\sum_{i=1}^{M} EQM_i = \frac{1}{M}\sum_{i=1}^{M} E\left[e_i^2\right]. \tag{3.4}$$

Considere agora a abordagem *ensemble* utilizando média simples como forma de combinação. Logo, a saída do *ensemble*, $y_{ensemble}(\mathbf{x})$, seria:

$$y_{ensemble}(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M} y_i(\mathbf{x}) \,. \tag{3.5}$$

O erro cometido pelo *ensemble*, $EQM_{ensemble}$, é dado por:

$$EQM_{ensemble} = E\left[\left(\frac{1}{M}\sum_{i=1}^{M} y_i(\mathbf{x}) - f(\mathbf{x})\right)^2\right] = E\left[\left(\frac{1}{M}\sum_{i=1}^{M} e_i\right)^2\right]. \tag{3.6}$$

Supondo que os erros $e_i(\mathbf{x})$ possuem média zero e são não correlacionados, resulta:

$$E\left[e_i\right] = 0 \,, \quad E\left[e_i e_j\right] = 0 \ \text{ se } \ j \neq i \,. \tag{3.7}$$

Da equação (3.4) e da equação (3.6), e levando-se em conta os resultados da equação (3.7), é possível obter (BISHOP, 1995):

$$EQM_{ensemble} = \frac{1}{M^2}\sum_{i=1}^{M} E\left[e_i^2\right] = \frac{1}{M} EQM_{me}\,. \tag{3.8}$$

Este resultado é bastante interessante, pois diz que a somatória do erro quadrático pode ser reduzido por um fator *M*, simplesmente tirando a média das saídas dos *M* componentes. No entanto, na prática a redução do erro é geralmente muito menor, porque o erro $e_i(x)$ de modelos diferentes são freqüentemente correlacionados. Logo, a suposição feita na equação (3.7) não é verdadeira. Portanto, é necessário gerar componentes com *EQM* baixo (caso contrário, o EQM_{me} será alto) e que sejam pouco correlacionados com os outros componentes. Para tanto, uma estratégia possível é aquela proposta por YAO & LIU (1998) em que os componentes são treinados seqüencialmente, visando diminuir a correlação dos componentes que estão sendo gerados com aqueles que já foram treinados.

Pode-se mostrar, também, que o *ensemble* não produz um incremento no erro quadrático médio. Utilizando as desigualdades de Cauchy (BISHOP, 1995), tem-se:

$$\left(\sum_{i=1}^{M} e_i\right)^2 \leq M\sum_{i=1}^{M} e_i^2\,, \tag{3.9}$$

da qual resulta:

$$EQM_{ensemble} \leq EQM_{me}\,. \tag{3.10}$$

3.2.3 Um breve histórico dos *ensembles*

Talvez um dos primeiros trabalhos sobre *ensemble* na comunidade de RNAs é o artigo de DASARATHY & SHEELA (1979), em que os autores discutem o particionamento do espaço de características usando dois ou mais classificadores. HANSEN & SALAMON (1990) mostraram que o desempenho de generalização de uma RNA pode ser melhorado usando um *ensemble* de RNAs configuradas com o mesmo número de neurônios. SCHAPIRE (1990) provou que um classificador *forte* no sentido de aprendizado provavelmente aproximadamente correto (PAC do inglês *probably approximately correct*) pode ser gerado pela combinação de classificadores fracos através do algoritmo *boosting* (SCHAPIRE, 1990), o antecessor do algoritmo *AdaBoost* (FREUND & SCHAPIRE, 1996).

Após esses trabalhos iniciais, a pesquisa em ensemble vem se expandido rapidamente, aparecendo freqüentemente na literatura, com várias denominações e variantes criativas. Entre elas, podem ser citadas: composição de sistemas classificadores (DASARATHY & SHEELA, 1979), generalização empilhada (WOLPERT, 1992), agregação de consenso (BENEDIKTSSON & SWAIN, 1992), combinação de vários classificadores (XU *et al.*, 1992; HO *et al.*, 1994, ROGOVA, 1994), seleção dinâmica de classificadores (WOODS *et al.*, 1997), fusão de classificação (CHO *et al.*, 1995; KUNCHEVA, 2001), agrupamento de redes neurais (DRUCKER *et al*. 1994), reservatório de classificador (BATTITI & COLLA, 1994), ensemble de classificadores (DRUCKER *et al*. 1994; KUNCHEVA, 2004), sistema pandemonium de agentes reflexivos (SMIEJA, 1996), SVMs diversos como componentes de um *ensemble* (LIMA *et al.*, 2004a), entre muitos outros.

Os paradigmas desta abordagem diferem uns dos outros com relação ao procedimento usado para *gerar* os componentes individuais e/ou estratégia empregada na *combinação* e/ou *seleção* de componentes. Na próxima seção, apresentam-se as etapas a serem seguidas para a construção de um *ensemble*: geração, combinação e seleção.

3.3 Etapas no projeto de um *ensemble*

As etapas no projeto de um ensemble são as seguintes: geração, combinação e seleção. Na Figura 3.5, são apresentadas de forma gráfica as fases que compreendem a construção de uma arquitetura para um *ensemble*.

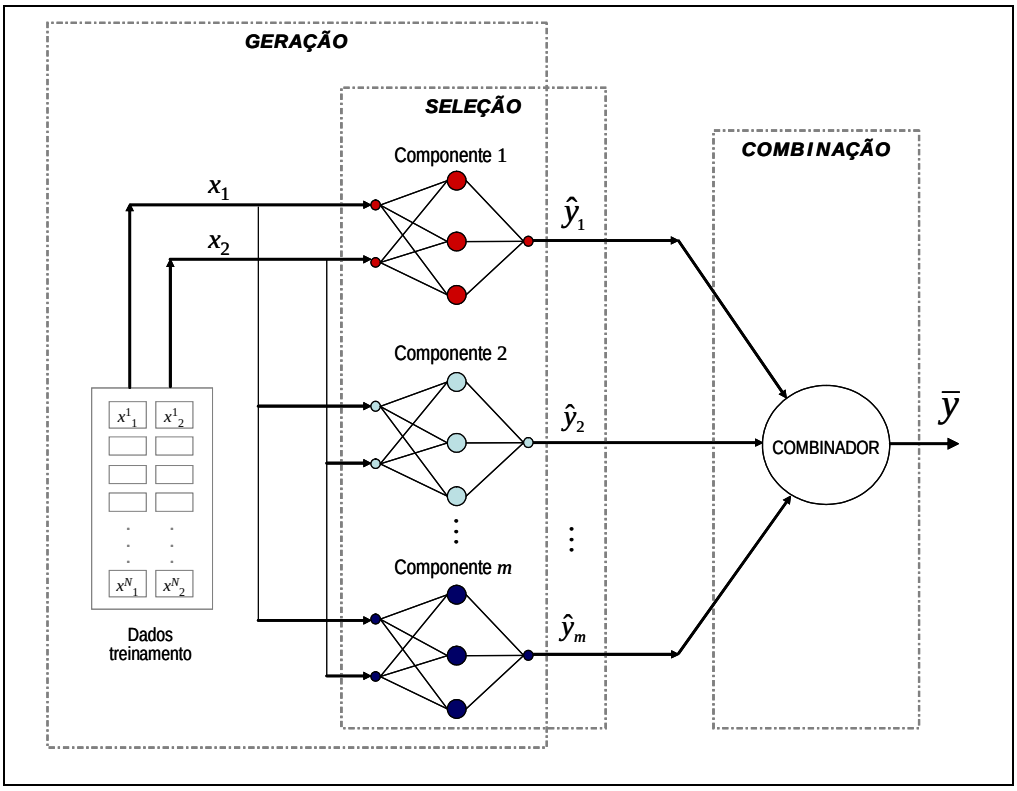


Figura 3.5 – Etapas na construção de uma arquitetura para um *ensemble.*

3.3.1 Etapa de geração de componentes

A etapa de geração de componentes candidatos a participarem do *ensemble* deverá ser guiada tendo em mente dois requisitos fundamentais: o baixo erro de treinamento e a descorrelação dos perfis de erro dos componentes. Está implícito que a geração de componentes envolve múltiplas iniciativas de solução do problema, uma para cada candidato.

No caso de RNAs como componentes do *ensemble*, há várias formas de tentar alcançar diversidade. Dentre elas, podemos citar as seguintes:

- **Inicialização aleatória dos pesos das conexões de cada RNA**: com isto espera-se que o processo de treinamento das redes convirja para mínimos locais distintos e, por conseqüência, cria-se a possibilidade de que elas generalizem de forma diferente.
- **Variação da arquitetura da RNA**: pode-se realizar variação na arquitetura da seguinte forma: modificando o número de camadas e/ou o número de neurônios na camada escondida e/ou a função de ativação.
- **Variação do algoritmo de treinamento**: utilização de algoritmos de treinamento baseados em informação de primeira e segunda ordem, dentre outras opções. Este ponto pode também incluir o uso de distintas funções-objetivo, as quais guiam o processo de otimização. Uma alternativa aqui também é o emprego de algoritmos de otimização que buscam múltiplas propostas de otimização simultaneamente, fornecendo assim várias redes neurais artificiais ao final do processo de otimização.
- **Re-amostragem dos dados**. Técnicas de re-amostragem como *Bagging* (BREIMAN, 1996), *Boosting* (FRIEDMAN *et al.*, 2000) introduzem, também, diversidade nos componentes, pois cada um vai ser treinado com um conjunto de treinamento distinto.

É possível considerar mais de uma destas formas de geração para inserir diversidade nos componentes do *ensemble*, assim como é possível gerar um *ensemble* heterogêneo, em que os componentes são modelos distintos, por exemplo, redes neurais MLP e RBF, máquinas de vetores-suporte e sistemas nebulosos.

Na literatura, encontramos vários trabalhos relacionados a esta etapa de geração do ensemble, dentre os quais podemos citar:

- HAMPSHIRE & WAIBEL (1990) utilizaram funções de erro distintas para treinar cada componente;
- CHERKAUER (1996) adotou cada componente com um número distinto de neurônios na camada oculta;

- MACLIN & SHAVLIK (1995) utilizaram formas distintas de inicialização dos pesos sinápticos na RNA;
- OPITZ & SHAVLIK (1996) e YAO & LIU (1998) empregaram algoritmos genéticos para sintetizar componentes com comportamentos distintos;
- COELHO (2006) usou algoritmos imunológicos artificiais para buscar múltiplas propostas de otimização simultaneamente, visando preservar a diversidade de comportamento dos componentes.

3.3.1.1 Geração via *bagging*

Bagging vem de ***B**ootstrap **ag**regat**ing*** ou agregação *bootstrap* (BREIMAN, 1996). Trata-se de uma técnica de geração de conjuntos distintos de treinamento a partir de um único conjunto, via re-amostragem com reposição. Estes conjuntos de treinamento serão usados para treinar cada rede neural do tipo MLP que compõe o *ensemble*. A Figura 3.6 mostra graficamente este processo. Considere que se parta de um único conjunto de treinamento, conjunto fonte com N amostras, para treinar m componentes do tipo MLP. O objetivo é construir m conjuntos de treinamento com N amostras cada um. Supondo uma distribuição de probabilidade uniforme para seleção das amostras a partir do conjunto fonte, a existência de reposição de amostras selecionadas implica que algumas amostras podem ser selecionadas mais de uma vez. Com isso, apenas um subconjunto das amostras do conjunto fonte estará em cada um dos m conjuntos, o que implica na existência de amostras repetidas, já que o total de amostras é N.

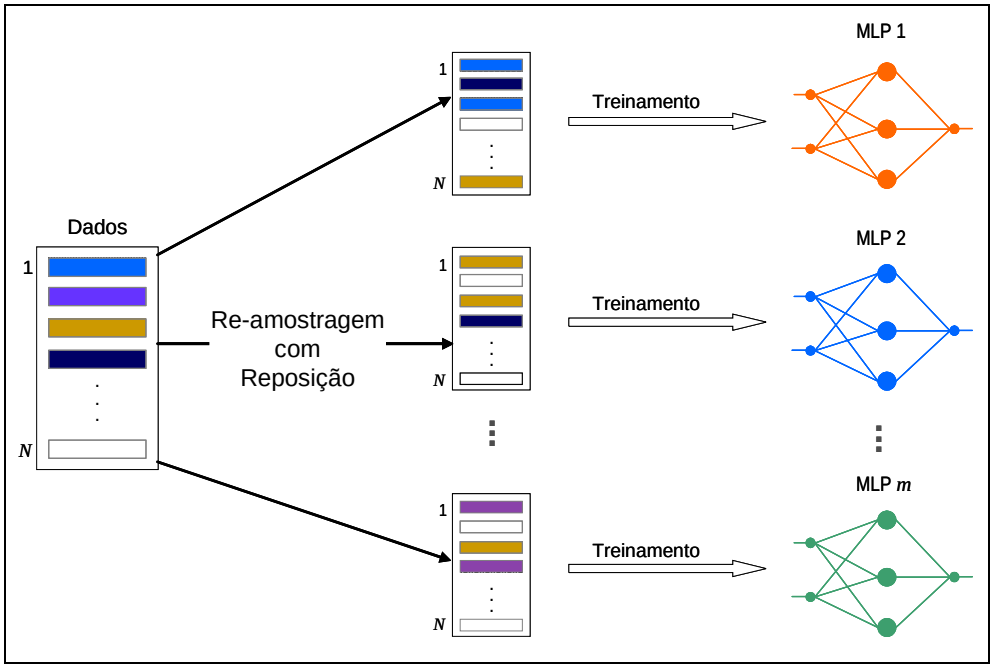


Figura 3.6 – Geração de conjuntos de treinamento distintos via *bagging*.

Esta estratégia de inserir diversidade nos componentes requer que eles produzam comportamento distinto sempre que são submetidos a conjuntos de treinamento distintos. Tanto as RNAs quanto os outros modelos de máquina de aprendizado obedecem a esta requisição, sendo assim denominados de modelos instáveis (BREIMAN, 1996).

É possível que os componentes não generalizem satisfatoriamente, mas a agregação deles tende a produzir uma generalização melhor. No entanto, sempre vai existir um compromisso entre a capacidade de generalização dos componentes e o desempenho obtido após a agregação (TANIGUCHI & TRESP, 1997). Além da re-amostragem com reposição, há na literatura mecanismos de adição de perturbação com ruído nos conjuntos de treinamento gerados (BREIMAN, 2000; DIETTERICH, 2000; RAVIV & INTRATOR, 1999; SHARKEY, 1999).

3.3.1.2 Variações do *bagging*

Random Forest: Uma variação do algoritmo *bagging* é o *Random Forest* (BREIMAN, 2001). Um *Random Forest* pode ser criado a partir de árvores de decisão, cujos parâmetros de treinamento variam randomicamente. Tais parâmetros podem ser réplicas dos dados de

treinamento, como no *bagging*, mas podem ser um subconjunto de características diferentes, como nos métodos de sub-espaço randômico.

Rotation Forest: RODRIGUEZ *et al.* (2006) propôs um método para geração de classificadores para ensemble baseado na extração de características. Para criar os dados de treinamento para os componentes, o conjunto de características é randomicamente separado em *K* subconjuntos (*K* é um parâmetro do algoritmo) e uma análise de componentes principais (PCA) é aplicada a cada subconjunto. Todas as direções principais obtidas via PCA são mantidas a fim de preservar a variabilidade nos dados. Então, *K* eixos de rotação são utilizados para formar as novas características dos componentes do ensemble-base. A idéia da abordagem de rotação é produzir precisão e diversidade dentro do ensemble. Diversidade é promovida através da extração de características para cada componente. Árvores de decisão foram escolhidas por serem sensíveis à rotação dos eixos de características, assim o nome "Forest". Precisão é obtida assegurando que todas as direções principais sejam consideradas e também usando todo o conjunto de dados para treinar cada componente.

Pasting Small Votes: assim como no *bagging*, *pasting small votes* é projetado para ser usado com grandes conjuntos de dados (BREIMAN, 1999). Um grande conjunto de dados é particionado em pequenos subconjuntos, chamados *bites*, cada um dos quais é usado para treinar um componente diferente. Duas variações do *pasting small votes* foram propostas: uma que cria conjuntos de dados randômicos, chamada *Rvotes*, e uma que cria subconjuntos consecutivos baseados na importância das instâncias, chamada *Ivotes*. A última abordagem é conhecida por proporcionar melhores resultados (CHAWLA *et al.*, 2002), e é similar à abordagem seguida pela algoritmo baseado em *boosting*, em que cada componente focaliza atenção nas instâncias mais importantes (ou mais informativas) para o componente atual.

3.3.1.3 Geração via *boosting*

SCHAPIRE (1990) mostrou que um preditor *fraco*, um algoritmo que gera classificadores que podem ser meramente melhor que a escolha da classe ao acaso, pode ser transformado

em um preditor *forte*, ou seja, que pode classificar uma fração arbitrária de exemplos corretamente (SCHAPIRE, 1990). Definições formais de preditor *fraco* e *forte* podem ser encontradas em SCHAPIRE (1990). SCHAPIRE (1990) propôs um algoritmo para transformar o desempenho de um preditor fraco em um preditor forte, e esse algoritmo foi denominado *boosting*. Este algoritmo é considerado como um dos desenvolvimentos mais importantes na história do aprendizado de máquina.

De forma similar ao *bagging*, o *boosting* também cria vários conjuntos de treinamento por re-amostragem dos dados. Entretanto, a similaridade entre essas duas abordagens é apenas esta. No *boosting*, a re-amostragem é estrategicamente gerada para fornecer dados de treinamento mais informativos para componentes que são gerados de forma consecutiva.

3.3.1.4 Geração *Adaboost*

FREUND & SCHAPIRE (1997) introduziram o *AdaBoost* com singular sucesso. Essa é uma versão mais geral do algoritmo *boosting* original. Conta com outras variações como *AdaBoost.M1* e *AdaBoost.R*, as quais são as mais comumente utilizadas, pois são capazes de atacar problemas com múltiplas classes e problemas de regressão.

3.3.2 Etapa de combinação de componentes

Depois de gerados e treinados os componentes, resta ver como eles serão combinados para produzir uma resposta única a partir das respostas individuais. Dependendo da tarefa a resolver, será aplicado um método de combinação específico. A seguir, apresentam-se formas de como combinar componentes em problemas de regressão, classificação e agrupamento de dados.

- *Em classificação de dados*: podem ser empregadas técnicas baseadas em votação a partir da saída simbólica dos componentes. O mais usado é o voto simples, onde o resultado é indicado pela metade mais um dos resultados (aqui supondo a existência de apenas duas classes), como mostrado na Figura 3.7(a).

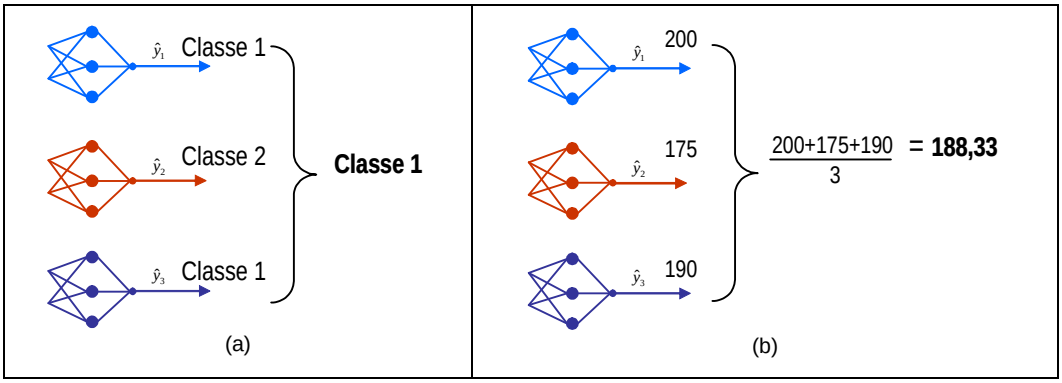
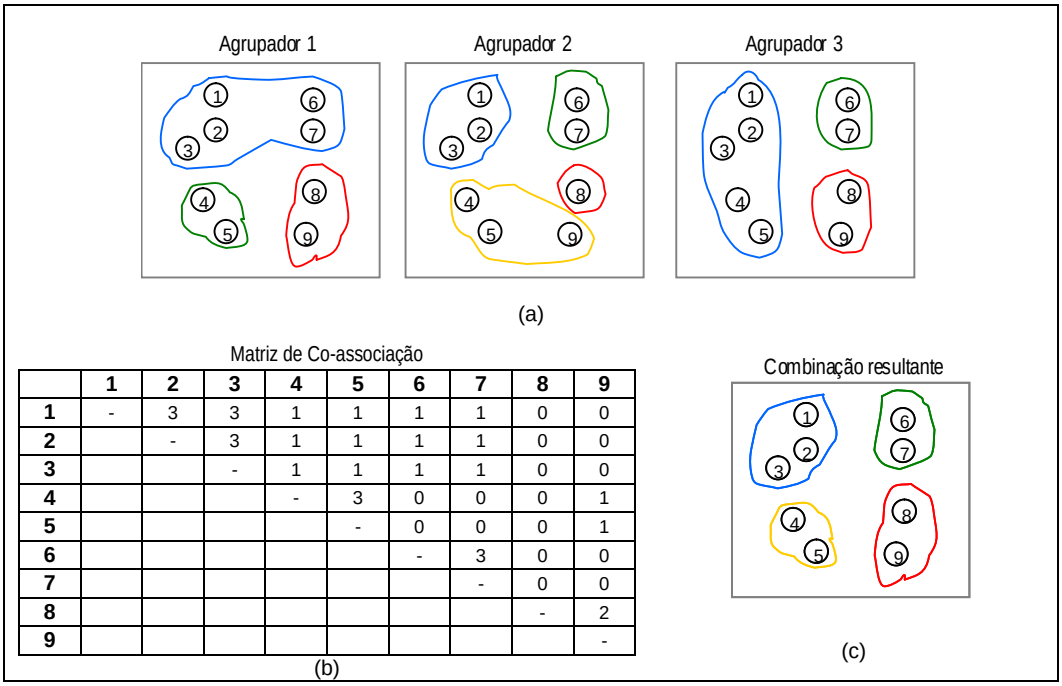


Figura 3.7 – Combinação de componentes para tarefas de classificação (esquerda) e regressão (direita).

- *Em regressão de dados*: pode aplicar-se a média simples ou a média ponderada das saídas numéricas dos componentes. Para o caso de média ponderada, requer-se um processo de ajuste dos coeficientes de ponderação, o qual pode ser aproximado via o método dos quadrados mínimos. A Figura 3.7(b) ilustra o caso de média simples. Em casos de várias saídas, uma abordagem é calcular as médias de cada saída de forma separada (HASHEM, 1997; NAFTALY & INTRATOR, 1997).

- *Em agrupamento de dados*: neste caso, a combinação pode envolver técnicas um tanto mais elaboradas que nos casos anteriores, devido a que cada componente apresenta como resposta um conjunto de grupos. Além disso, o número de grupos pode não ser o mesmo. Para conceber uma única resposta como produto do consenso de todos os agrupadores, pode-se empregar a matriz de co-associação. A Figura 3.8 ilustra um caso didático que envolve 3 soluções (item "a" da figura) de agrupamento de 9 amostras. Monta-se uma matriz quadrada e triangular superior (item "b" da figura), onde as linhas e colunas representam cada amostra. Logo, preenchem-se as posições da matriz com o número de vezes que ambas as amostras (linha, coluna) apareceram no mesmo grupo em cada proposta de agrupamento. Para formar o agrupamento final a partir desta matriz, pode-se aplicar uma estratégia de voto majoritário. Todos aqueles pares de amostras com valor maior que a metade do número de agrupamentos estarão no mesmo grupo. O resultado está na parte "c" da Figura 3.8.



	1	2	3	4	5	6	7	8	9
1	-	3	3	1	1	1	1	0	0
2		-	3	1	1	1	1	0	0
3			-	1	1	1	1	0	0
4				-	3	0	0	0	1
5					-	0	0	0	1
6						-	3	0	0
7							-	0	0
8								-	2
9									-

Figura 3.8 – Combinação de componentes em problemas de agrupamento.

Informações adicionais sobre esta matriz de co-associação podem ser encontradas em FRED & JAIN (2002). Outras formas mais sofisticadas de combinação de agrupamentos podem ser encontradas nos trabalhos de STREHL & GHOSH (2002a, 2002b), onde a combinação de agrupamentos é formalizada como um problema de otimização.

3.3.3 Etapa de seleção de componentes

A seleção pode ser considerada como uma etapa de refinamento do *ensemble* e, dependendo da estratégia adotada na etapa de geração, pode ser um passo fundamental visando melhorar o desempenho do *ensemble*. Uma vez gerados os componentes, usando uma ou algumas das estratégias descritas na seção 3.3.1, há a possibilidade de que nem todos os componentes contribuam para o desempenho global do *ensemble*, e o indicado seria identificar esses componentes e descartá-los do *ensemble* final. Resumindo, o objetivo é maximizar o desempenho de generalização pela seleção de um subconjunto de componentes dentre o total de candidatos gerados.

Deverá ser empregado algum critério para selecionar os componentes que irão compor o *ensemble* final, que pode ser uma medida de erro (por exemplo, o EQM) sobre um subconjunto dos dados, os quais serão separados exclusivamente para este propósito. Esse conjunto de dados será denotado *conjunto de seleção*, e o erro obtido corresponderá então ao *erro de seleção*.

Na literatura, existem várias formas de realizar seleção, dependendo do problema ser de classificação ou regressão. Dentre as propostas mais difundidas, encontram-se aquelas que seguem uma de duas heurísticas: *construtiva* e *de poda*.

3.3.3.1 Método construtivo

O método construtivo consiste em:

- **1º Passo:** Parte-se com o *ensemble* vazio (sem componentes). Em seguida, ordenam-se os componentes baseados em algum critério. Por exemplo, *erro de seleção*.
- **2º Passo:** Aquele com melhor valor do critério será o primeiro componente selecionado para compor o *ensemble*.
- **3º Passo:** Verifica-se se existe algum dos componentes restantes que, ao ser adicionado ao *ensemble*, produza melhoria no critério. Caso exista, adiciona-se o mesmo ao *ensemble*. Caso contrário, vá para o 5º Passo. Observe que esta última tarefa poderia ser realizada de duas formas, selecionando o primeiro que melhorar (*first improvement*), ou selecionando aquele que produz a melhora mais significativa (*best improvement*).
- **4º Passo:** Volte ao 3º passo.
- **5º Passo:** Fim do processo

3.3.3.2 Método de poda

O método de poda faz o caminho contrário do método construtivo. A diferença é que não se parte de um *ensemble* vazio e sim de um com todos os componentes gerados. Em

seguida, ordenam-se em ordem decrescente os componentes de acordo com algum critério, por exemplo, o *erro de seleção* individual. Verifica-se se a retirada do primeiro componente do *ensemble* causa uma redução do erro. Se isto acontecer, retira-se este componente. Se não, testa-se o próximo componente. Da mesma forma que no método construtivo, é possível também aplicar os critérios de "o primeiro que melhorar" ou "o que melhorar mais". Repete-se este processo até não existir componente que, quando retirado, produza uma melhora do critério ou do *erro de seleção* do *ensemble*.

Nesta dissertação, foram consideradas as seguintes estratégias de geração:

- *Ensemble* com critério de "geração por inicialização aleatória dos pesos das conexões das redes MLP" e combinação via média simples (*Ensemble-Averaging*);
- *Ensemble* com geração via reamostragem do tipo *bagging* (incluindo inicialização aleatória dos pesos das conexões das redes MLP) e critério de combinação média simples (*Ensemble-Bagging*).

3.4 Predição de séries temporais usando *ensemble*

Nesta seção, apresentam-se resultados comparativos entre *Ensemble-Averaging* e *Ensemble-Bagging*, usados em predição de séries temporais. As séries já foram apresentadas no Capítulo 2. É possível contrastar os resultados da abordagem ensemble com os resultados obtidos por uma única RNA, no caso, uma MLP.

A Tabela 3.1 apresenta os resultados para as 5 séries temporais, resultados colhidos a partir do conjunto de teste. Simularam-se as abordagens *Ensemble-Averaging*, *Ensemble-Bagging*, as quais são contrastadas com os resultados obtidos via uma única MLP. Realizaram-se 30 simulações de cada, com a finalidade de calcular estatísticas como média e desvio padrão, mínimo e máximo valor obtidos. Os valores correspondem ao erro MAE (*mean absolute error*) ou erro absoluto médio.

A configuração de parâmetros foi a seguinte:

MLP: 30 neurônios ocultos, 1000 épocas de treinamento, validação cruzada e parada antecipada, os últimos 25% das amostras foram separadas seqüencialmente para o teste, 50 e 25% das amostras restantes para treinamento e validação, respectivamente, ambos de forma aleatória.

Ensembles: O número de componentes inicial foi 50, para ambos os tipos de *ensemble*, todos eles representam MLPs com os mesmos parâmetros que no caso da única MLP. Apenas diferenciam-se na aplicação da etapa de seleção de componentes, em que se aplicou a estratégia construtiva e, para este fim, uma porcentagem das amostras teve que ser separada. Assim, as porcentagens foram: 50, 5 e 20% para treinamento, validação e seleção, respectivamente.

Tabela 3.1: Resultados comparativos empregando o conjunto de teste: única MLP, *Ensemble-Averaging* e *Ensemble-Bagging* com e sem Seleção.

Série Temporal	Estatísticas	MLP	*Ensemble-Averaging*: sem Seleção	*Ensemble-Averaging*: com Seleção	*Ensemble-Bagging*: sem Seleção	*Ensemble-Bagging*: com Seleção	No. comp. *Ens-Averaging*	No. comp. *Ens-Bagging*
Book store	**Média**	**84,82**	**82,73**	**77,38**	**87,96**	**78,10**	**4,13**	**11,30**
	STD	14,95	10,82	5,70	11,16	5,23	4,94	5,96
	Max	121,85	116,26	89,92	132,68	95,35	24,00	30,00
	Min	69,08	70,77	67,43	75,10	70,46	1,00	3,00
Clothing store	**Média**	**781,35**	**872,80**	**715,23**	**959,14**	**607,80**	**4,37**	**8,90**
	STD	437,18	398,40	311,01	233,68	201,03	6,06	6,75
	Max	1767,90	1743,40	1721,40	1573,20	1170,60	24	30
	Min	477,71	400,20	505,58	669,52	411,29	1	0
Furniture store	**Média**	**176,44**	**167,03**	**162,10**	**165,80**	**161,85**	**5,00**	**6,33**
	STD	32,77	15,78	7,46	17,50	7,07	7,07	9,11
	Max	310,36	234,32	178,40	232,43	181,34	24	30
	Min	155,78	153,18	147,51	148,57	143,67	1	0
Hardware store	**Média**	**42,68**	**42,56**	**40,33**	**42,91**	**39,66**	**8,30**	**11,23**
	STD	8,89	6,16	3,77	5,80	2,06	7,57	10,57
	Max	73,92	55,80	49,13	64,37	44,33	24	30
	Min	34,12	32,88	33,59	33,51	36,66	1	0
Durable Consumer Goods	**Média**	**4,11**	**3,79**	**3,65**	**3,89**	**3,59**	**2,77**	**3,80**
	STD	0,32	0,05	0,21	0,34	0,18	1,43	2,43
	Max	4,74	3,90	4,44	5,61	4,33	6,00	11,00
	Min	3,57	3,71	3,40	3,70	3,33	1,00	1,00

Na Figura 3.9, ilustra-se de forma gráfica os resultados da Tabela 3.1, utilizando o recurso *boxplot*. Os asteriscos representam o valor médio das 30 simulações realizadas e os

valores em porcentagens sobre as terceiras e quintas colunas representam o ganho em relação ao valor médio de uma única MLP. Pode-se concluir que, para as séries testadas, o uso de *ensemble* incluindo a etapa de seleção foi em todos os casos superior a uma única MLP. E que o desempenho de *Ensemble-Bagging* foi em apenas um caso (série *Book store*) inferior ao desempenho do *Ensemble-Averaging*.

Vale notar que a aplicação da etapa de seleção foi fundamental, já que as estratégias adotadas no processo de geração não foram capazes de gerar sempre componentes com boa capacidade de predição e suficientemente diversos entre si.

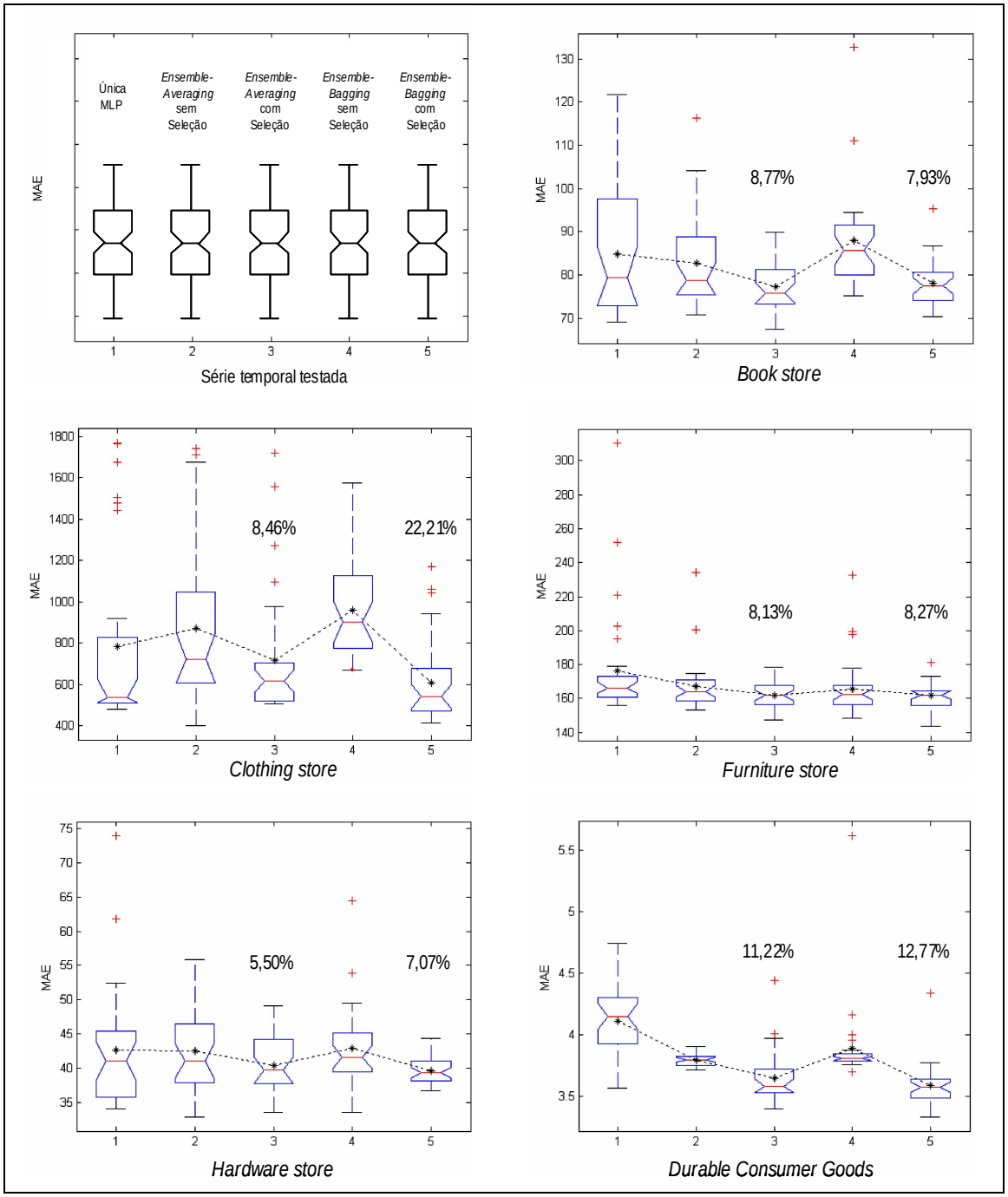


Figura 3.9 – Gráficos *boxplot* obtidos a partir da Tabela 3.1.

A Figura 3.10 mostra o número de componentes selecionados após aplicar a etapa de seleção construtiva descrita na seção 3.3.3.1. Inicialmente, foram gerados 50 componentes via *Ensemble-Averaging* e *Ensemble-Bagging*. Do gráfico, desprende-se que a geração

utilizando a técnica de *bagging* produz maior diversidade nos componentes, de tal forma que o número de componentes selecionados via esta estratégia foi superior àquele da geração por simples inicialização aleatória dos parâmetros dos componentes. Isto explica por que os melhores resultados foram obtidos via *Ensemble-Bagging*.

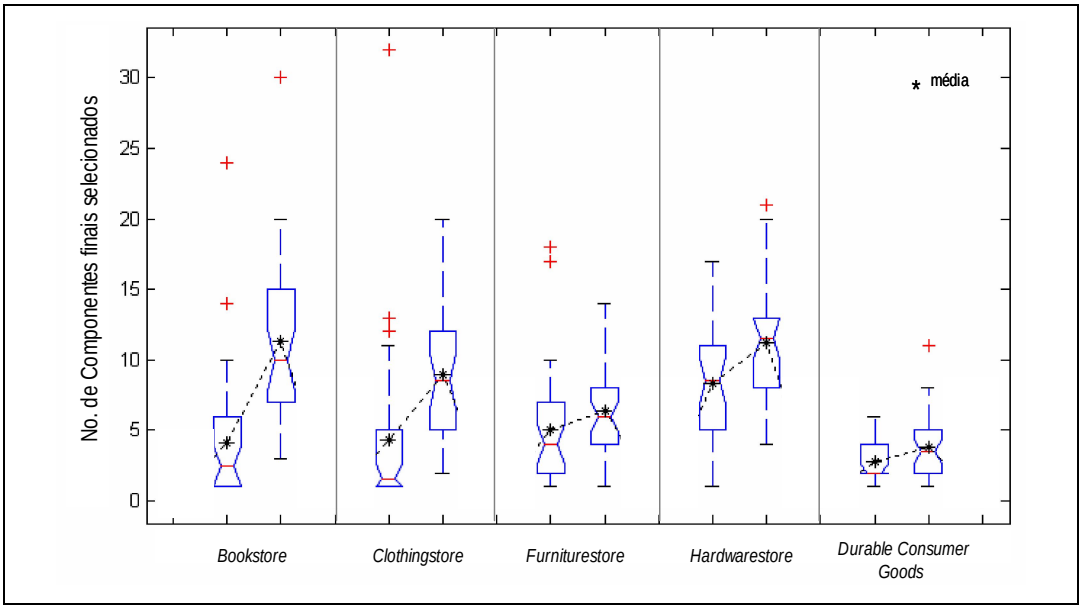


Figura 3.10 – Número final de componentes selecionados em *Ensemble-Averaging* (esquerda) vs *Ensemble-Bagging* (direita), para cada série temporal testada.

Na Figura 3.11, ilustra-se um exemplo para a predição da série *Clothing store*: (a) usando uma única MLP; (b) ensemble combinando os 50 componentes gerados; e (c) ensemble aplicando seleção, onde apenas os componentes 8, 30, 21, 1, 23, 45, 10, 34 foram selecionados para compor o ensemble final, nesta ordem.

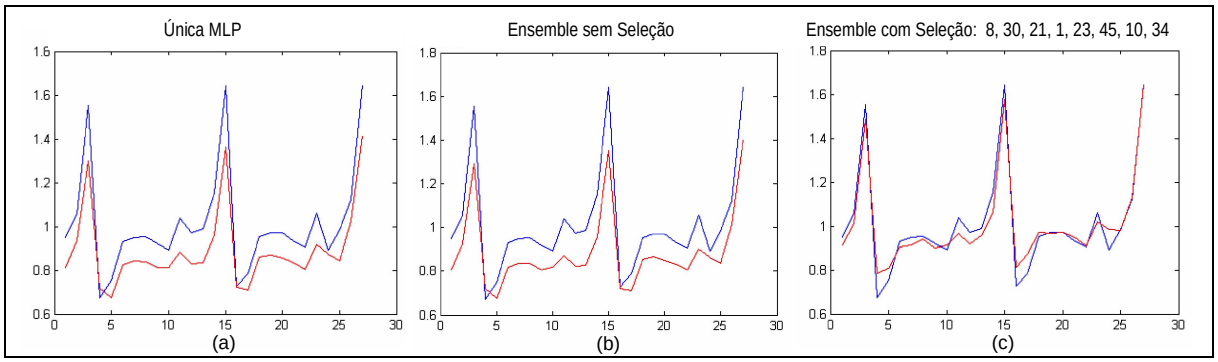


Figura 3.11 – Exemplo de predição: série *Clothing store*, (a) única MLP, (b) ensemble total (50 componentes) e (c) ensemble com seleção (componentes selecionados: 8, 30, 21, 1, 23, 45, 10, 34).

3.5 Mistura de Especialistas

Mistura de Especialistas (ME) é uma proposta de comitê de máquinas na qual o espaço de entrada é automaticamente dividido em regiões durante o treinamento e, para cada região existe um único ou um subconjunto de especialistas mais indicados para atuar, os quais são também concebidos durante o processo de treinamento. A divisão deste espaço pode ser linear ou não-linear. Pode também ser gradual e contemplar sobreposições de regiões. Isto se consegue a partir da implementação de uma "rede *gating*" (ver Figura 3.12) que define os coeficientes (g_1,...,g_m) da combinação convexa envolvendo a saída de cada especialista (y_1,...,y_m). Esses coeficientes devem ser sempre não-negativos e somar na unidade.

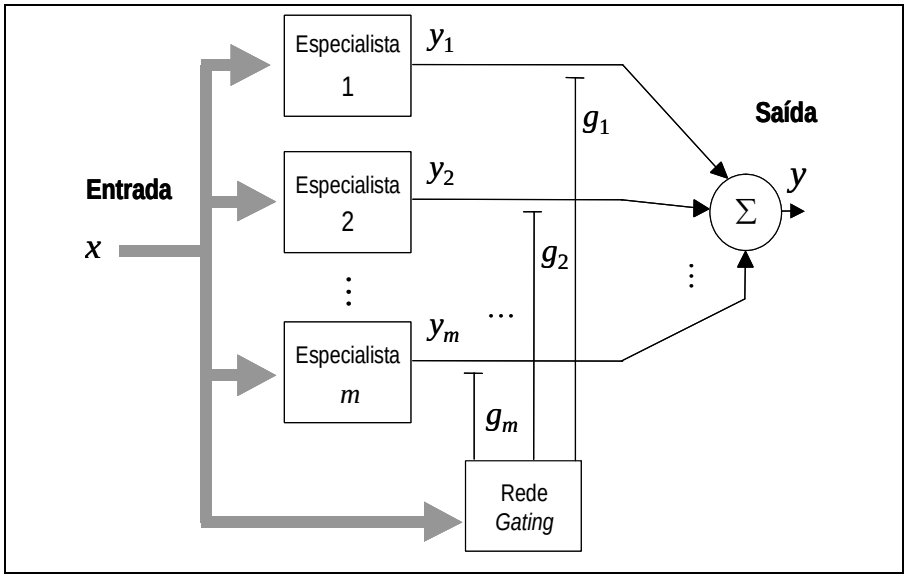


Figura 3. 12 – Estrutura típica de uma arquitetura de Mistura de Especialistas.

Continuando com a Figura 3.12, nela pode-se identificar a existência de *m* especialistas, os quais seriam propostas de máquinas de aprendizado, por exemplo RNAs, SVMs, modelos lineares, etc. A diferença para *ensembles* está no fato de que em MEs a rede *gating*, responsável pela combinação convexa das propostas de saída dos especialistas, recebe também a entrada ***x***, e a utiliza para decompor o espaço de entrada em diferentes

regiões. Com isso, é possível indicar automaticamente a região de atuação de cada especialista.

O caráter dinâmico de MEs (HAYKIN, 1999) deve-se ao fato de que as regiões de atuação a serem alocadas para os especialistas não são definidas a priori. Elas são implementadas de forma interativa e com garantia de convergência para um mínimo local.

No Capítulo 2, foi visto que a função-objetivo que guia o processo de treinamento de uma MLP comumente é a função somatória dos erros quadráticos, sendo que o objetivo é minimizar esta função. No caso de MEs, a função-objetivo é baseada na interpretação de MEs como modelos de mistura (MCLACHLAN & BASFORD, 1988; LIMA, 2004), o que implica numa função de verossimilhança, a qual deverá ser maximizada. Cada especialista terá uma função densidade de probabilidade condicional associada, com as saídas da rede *gating* desempenhando o papel de coeficientes da mistura. A Figura 3.13 ilustra o papel da rede *gating* na definição de um modelo de mistura, empregado na interpretação de MEs. Observe que o espaço de entrada foi decomposto em regiões, as quais foram alocadas a cada especialista.

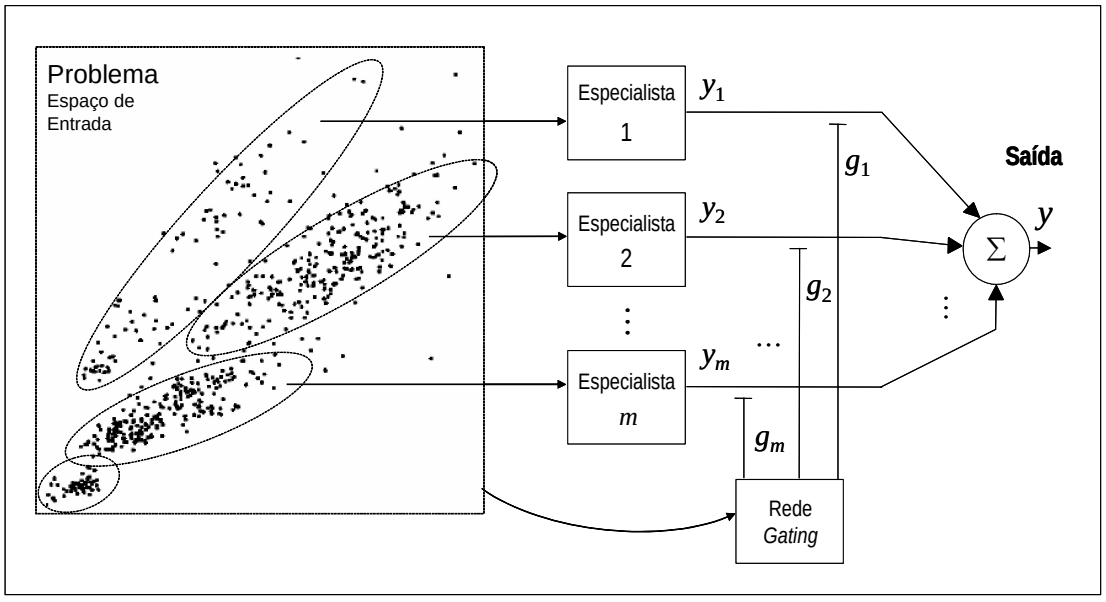


Figura 3.13 – Exemplo pictórico da decomposição do espaço de entrada via MEs.

Desta forma, cada especialista buscará se especializar no reconhecimento de padrões com algum tipo de associação em comum. Um exemplo apenas ilustrativo é o apresentado na Figura 3.14, onde mostra-se a operação da rede *gating* considerando apenas dois especialistas. No lado direito, vemos a arquitetura de ME empregada, g_1 e g_2 são as saídas da rede *gating* em resposta a um padrão do espaço de entrada. No lado esquerdo, acima, ilustra-se a saída desejada (problema de identificação de sistemas) e embaixo os valores das saídas g_1 e g_2 para cada valor de entrada do problema. Nota-se que a rede *gating* atribui de modo geral os picos positivos ao Especialista 1 e os picos negativos ao Especialista 2. Vale salientar que as saídas g_1 e g_2 devem sempre somar a unidade.

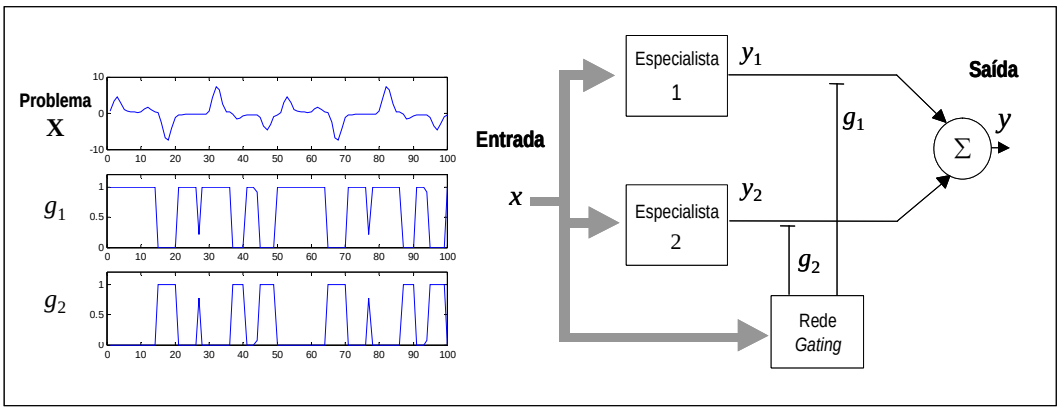


Figura 3.14 – Exemplo de operação da rede *gating* para ponderar as saídas dos especialistas em função das entradas do problema.

3.5.1 Propostas concorrentes à mistura de especialistas

O princípio de dividir-para-conquistar não foi unicamente adotado pela comunidade de RNAs. Também os estatísticos a adotaram, com propostas como TAR (*Threshold auto-regressive*) de TONG (1983, 1990), onde são empregados modelos lineares por partes. O algoritmo CART (*Classification and Regression Trees*) proposto por BREIMAN *et al.* (1984), o algoritmo MARS (*Multivariate Adapative Regression Splines*) proposto por FRIEDMAN (1991) e o algoritmo ID3 proposto por QUINLAN (1986) são exemplos bem conhecidos. Esses algoritmos definem o mapeamento de entrada-saída dividindo o espaço

de entrada explicitamente em sub-regiões e ajustando superfícies simples dentro destas sub-regiões. Esses algoritmos têm tempo de convergência que é freqüentemente uma ordem de magnitude mais rápido que o algoritmo baseado no gradiente para RNAs.

3.5.2 Arquitetura de Mistura de Especialistas

A arquitetura a ser considerada foi apresentada na Figura 3.12, sendo composta por *m* módulos referidos como redes especialistas, cada um implementando uma função parametrizada $y_i = f_i(\theta_i, \mathbf{x})$ da entrada **x** para a saída y_i, onde θ_i é o vetor de parâmetros do especialista *i*. As saídas de cada uma das redes especialistas recebem uma interpretação probabilística, considerando que o especialista i gera a saída y_i com probabilidade $P(\mathbf{y} \mid \mathbf{x}, \theta_i)$, onde y_i é a esperança matemática da variável aleatória **y**.

Considerando que diferentes redes especialistas são apropriadas para diferentes regiões do espaço de entrada, a arquitetura requer um mecanismo capaz de identificar, para cada entrada **x**, que especialista (ou combinação deles) é mais capaz de produzir a saída correta, em termos probabilísticos. Isto é realizado pela rede *gating*.

A interpretação probabilística da rede *gating* é de um sistema que calcula, para cada especialista, a probabilidade dele gerar a saída desejada, com base apenas no conhecimento de uma entrada **x**. Essas probabilidades são expressas pelos coeficientes g_i (i=1,...,m), de modo que estes devem ser não-negativos e devem produzir sempre o valor unitário quando somados, para cada **x**. Estes coeficientes não são constantes fixas, mas variam em função da entrada **x**. Caso os coeficientes g_i (i=1,...,m) sejam constantes e as redes especialistas atuem junto a todos os aspectos do problema, resultaria uma abordagem *ensemble*, conforme descrito na Seção 3.2.

Há muitas formas de garantir que os coeficientes g_i (i=1,...,m) atendam as restrições acima. Uma abordagem é utilizar a função *softmax* (JACOBS *et al.*, 1991). A função *softmax* define um conjunto de variáveis intermediárias ξ_i (i=1,...,m) como funções da entrada **x** e de um vetor de parâmetros $\mathbf{v}_i$ (i=1,...,m) na forma:

$$\xi_i = \xi_i(\mathbf{x}, \mathbf{v}_i)\,. \tag{3.11}$$

Com isso, os coeficientes g_i (i=1,...,m) podem ser definidos em termos de ξ_i (i=1,...,m) como segue:

$$g_i = \frac{\exp(\xi_i)}{\sum_{j=1}^{m} \exp(\xi_j)} \tag{3. 12}$$

A partir desta definição, os coeficientes g_i (i=1,...,m) passam a respeitar as restrições impostas, isto é, precisam ser não-negativos e, somados, produzirem sempre o valor unitário, para cada **x**. Uma interpretação probabilística para as variáveis intermediárias ξ_i (i=1,...,m) é que elas pertencem a uma família exponencial de distribuições de probabilidade (JORDAN & JACOBS, 1994).

A seguir, será especificado o modelo de probabilidade adotado para a arquitetura de mistura de especialistas. Considere que o conjunto de treinamento $\chi = \{(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})\}_{t=1}^{N}$ é gerado da seguinte forma: dada uma entrada **x**, um especialista i é escolhido com probabilidade $P(i \mid \mathbf{x}, \mathbf{v}^0)$ (onde o sobrescrito "0" será usado para distinguir os valores reais dos parâmetros do modelo de probabilidade adotado daqueles estimados pela rede *gating* ou pela rede especialista). Dada a escolha do especialista e dada a entrada, a saída desejada y é suposta ser gerada de acordo com a probabilidade $P(\mathbf{y} \mid \mathbf{x}, \theta_i^0)$. Cada um dos pares de entrada-saída é suposto ser gerado independentemente.

Observe que uma dada saída pode ser gerada de m formas diferentes, correspondendo aos m especialistas. Assim, a probabilidade total de geração de **y** a partir de **x** é dada pela soma sobre i, na forma:

$$P(\mathbf{y} \mid \mathbf{x}, \Theta^0) = \sum_{i=1}^{m} P(i \mid \mathbf{x}, \mathbf{v}^0) P(\mathbf{y} \mid \mathbf{x}, \theta_i^0) \tag{3. 13}$$

onde Θ^0 denota o vetor contendo todos os parâmetros, na forma $\Theta = [\theta_1^0, \theta_2^0, \ldots, \theta_m^0, \mathbf{v}^0]^T$. A função densidade de probabilidade na equação (3.13) é conhecida como *mistura de densidade* ou *função de verossimilhança*. É uma mistura de densidade no espaço de saída,

condicionada à escolha da entrada, onde $P(i \mid \mathbf{x}, \mathbf{v}^0)$ é a probabilidade de se escolher o especialista *i*, dada a entrada **x** e o parâmetro $\mathbf{v}^0$ da rede *gating*, e $P(\mathbf{y} \mid \mathbf{x}, \theta_i^0)$ é a probabilidade do especialista *i* gerar a saída **y**, dada a entrada **x** e seu vetor de parâmetros θ_i^0.

É tarefa da rede *gating* modelar as probabilidades $P(i \mid \mathbf{x}, \mathbf{v}^0)$, *i*=1,...,*m*. É possível parametrizar esta probabilidade via equações (3.11) e (3.12).

A saída da mistura de densidade pode ser calculada através da média condicional. A média condicional $\mathbf{y} = \mathrm{E}\left[P(\mathbf{y} \mid \mathbf{x}, \Theta^0)\right]$ é obtida tomando o valor esperado da equação (3.13):

$$\mathbf{y} = \sum_{i=1}^{m} g_i y_i \qquad (3.\ 14)$$

onde y_i é a média condicional associada à distribuição de probabilidade $P(\mathbf{y} \mid \mathbf{x}, \theta_i^0)$.

3.5.3 Formas de aprendizado em mistura de especialistas

O mecanismo de aprendizado de MEs pode ser realizado de várias formas. No entanto, a função-objetivo é sempre a *função de verossimilhança*, a qual foi apresentada na equação (3.13) e representa a função-base que guiará o processo de aprendizado. Em termos de otimização de parâmetros, seria a função-objetivo a ser maximizada.

Uma vez definida a função-objetivo, resta definir como será realizado o ajuste dos parâmetros, tanto dos especialistas quanto da rede *gating*. Na literatura, há duas estratégias propostas: a primeira pode ser vista como uma tentativa inicial e foi proposta por JACOB *et al.* (1991). Ela diz respeito à forma de ajuste de parâmetros e pode ser vista como um ajuste *acoplado* e/ou *simultâneo*, como ilustrado na Figura 3.15, onde todos os parâmetros, tanto

dos especialistas como da rede *gating*, são ajustados ao mesmo tempo. Esta forma de ajuste aumenta o risco de convergência para um mínimo local indesejado.

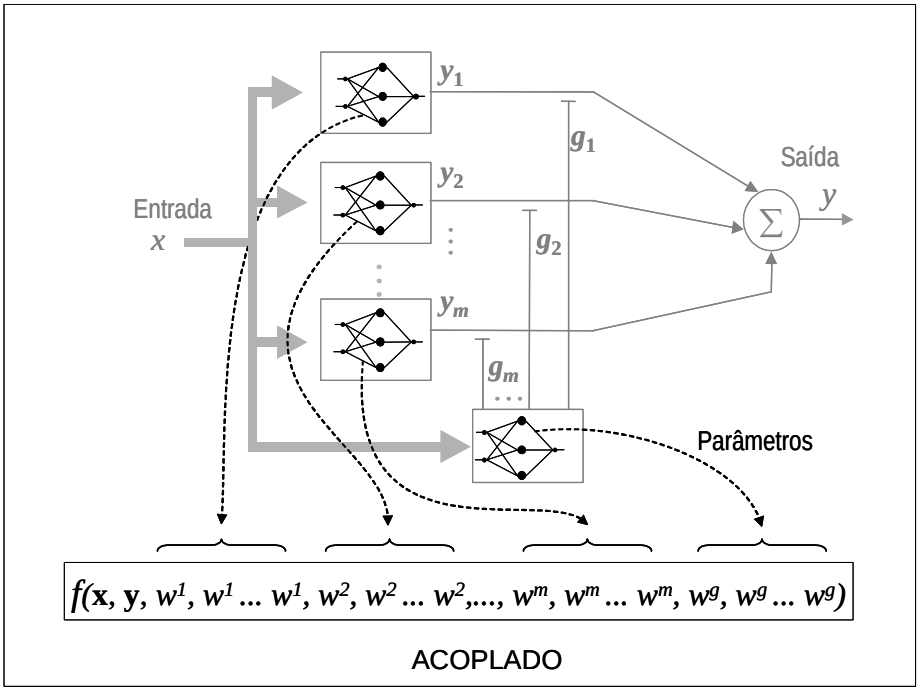


Figura 3.15 - Aprendizado acoplado e simultâneo para mistura de especialistas.

A segunda forma para o ajuste dos parâmetros de MEs é mais sofisticada e foi proposta por JORDAN & JACOBS (1994), sendo conhecida como método *desacoplado*. Para realizar o desacoplamento entre os especialistas e a rede *gating*, é utilizado o algoritmo de maximização da esperança (EM, do inglês *Expectation Maximization*) para MEs, e que foi inicialmente proposto por DEMPSTER *et al.* (1977) em outro contexto, voltado para aprendizado não-supervisionando. O algoritmo EM permite desacoplar o ajuste dos parâmetros do modelo de MEs. Dessa forma, o processo de treinamento, tanto dos especialistas quanto da rede *gating* pode ser realizado independentemente, a cada passo incremental. Uma conseqüência direta é a possibilidade de explorar de forma mais efetiva o espaço de busca das soluções, conseqüentemente diminuindo o risco de cair num mínimo local indesejado, quando comparado ao método acoplado. A Figura 3.16 ilustrar esta forma de ajuste de parâmetros. Observe que, neste caso, a *função* de *verossimilhança* dada pela equação (3.13) pode ser decomposta em soma de funções, as quais dependem

individualmente dos parâmetros de cada especialista e da rede *gating*. Logo, a maximização da *função de verossimilhança* pode ser realizada maximizando-se essas funções individuais, de forma incremental e iterativa.

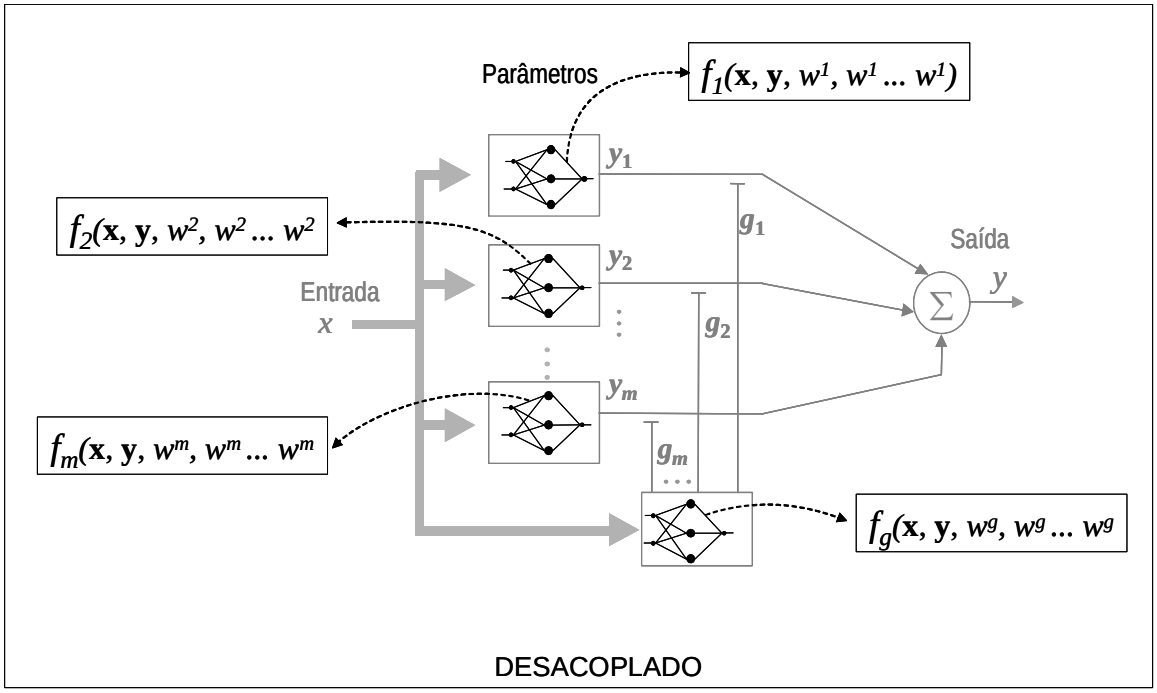


Figura 3.16 - Aprendizado desacoplado para mistura de especialistas.

Nas sub-seções seguintes, será apresentada em detalhes cada uma dessas formas de aprendizado para a abordagem ME.

3.5.3.1 Treinamento acoplado: baseado no método do gradiente simples

Para desenvolver este método de ajuste de parâmetros, parte-se do princípio de maximização da função de verossimilhança, definida na equação (3.13). Como é comumente adotado em estatística, é mais conveniente trabalhar com o logaritmo da verossimilhança que com a própria verossimilhança. Tomando o logaritmo de *m* densidades na forma da equação (3.13), chega-se à seguinte medida de verossimilhança:

$$l(\chi, \Theta) = \sum_{t=1}^{N} \log \sum_{i=1}^{m} P(i \mid \mathbf{x}^{(t)}, \mathbf{v}) P(\mathbf{y}^{(t)} \mid \mathbf{x}^{(t)}, \theta_i), \tag{3.15}$$

sendo N o número de padrões de treinamento.

Uma abordagem para maximizar o logaritmo da verossimilhança é usar o método do gradiente ascendente. Calculando o gradiente de $l(\cdot,\cdot)$ com respeito a y_i e ξ_i, resultam:

$$\frac{\partial l}{\partial y_i} = \sum_{t=1}^{N} h_i^{(t)} \frac{\partial}{\partial y_i} \log P(\mathbf{y}^{(t)} |, \mathbf{x}^{(t)}, \theta_i) \tag{3.16}$$

e

$$\frac{\partial l}{\partial \xi_i} = \sum_{t=1}^{N} \left(h_i^{(t)} - g_i^{(t)} \right), \tag{3. 17}$$

onde $h_i^{(t)}$ é definida como $P(i\,|\mathbf{x}^{(t)},\mathbf{y}^{(t)})$. Na derivação deste resultado, foi empregada a regra de Bayes:

$$P(i\,|, \mathbf{x}^{(t)}, \mathbf{y}^{(t)}) = \frac{P(i \mid \mathbf{x}^{(t)}) P(\mathbf{y}^{(t)} \mid \mathbf{x}^{(t)}, \theta_i)}{\sum_{j=1}^{m} P(j \mid \mathbf{x}^{(t)}) P(\mathbf{y}^{(t)} \mid \mathbf{x}^{(t)}, \theta_j)}. \tag{3.18}$$

Isto sugere que $h_i^{(t)}$ seja definida como a probabilidade *a posteriori* de escolha do *i*-ésimo especialista, condicionada à entrada $\mathbf{x}^{(t)}$ e à saída $\mathbf{y}^{(t)}$. De modo equivalente, a probabilidade $g_i^{(t)}$ pode ser interpretada como a probabilidade *a priori* $P(i,\mathbf{x}^{(t)})$, ou seja, a probabilidade da rede *gating* escolher o *i*-ésimo especialista, dada somente a entrada $\mathbf{x}^{(t)}$. Com essas definições, a equação (3.17) pode ser interpretada com uma forma de aproximar a probabilidade *a posteriori* utilizando a probabilidade *a priori*.

Um caso especial interessante é uma arquitetura na qual as redes especialistas e a rede *gating* são modelos lineares e a densidade de probabilidade associada com os especialistas é uma gaussiana com matriz de covariância igual à identidade (válido somente para problemas de regressão). Neste caso, a equação produz o seguinte algoritmo de aprendizado *on-line* (*on-line* implica apenas na ausência do somatório sobre *t*) (JORDAN & JACOBS, 1994):

$$\theta_i^{(k+1)} = \theta_i^{(k)} + \rho h_i^{(t)} (\mathbf{y}^{(t)} - y_i^{(t)}) \mathbf{x}^{(t)^T} \tag{3. 19}$$

$$\mathbf{v}_i^{(k+1)} = \mathbf{v}_i^{(k)} + \rho(h_i^{(t)} - g_i^{(t)})\mathbf{x}^{(t)^T} \tag{3.20}$$

onde ρ é a taxa de aprendizado e o índice k indica a iteração do processo de ajuste dos valores de θ_i e $\mathbf{v}_i$. Observe que ambas as equações acima têm a forma da regra dos quadrados mínimos (LS, do inglês *Least Squares*) clássica (WIDROW & STEARNS, 1985), com a atualização dos especialistas na equação (3.19) sendo modulada pela sua probabilidade *a posteriori* ($h_i^{(t)}$).

É também de interesse examinar a expressão para a probabilidade *a posteriori* no caso em que $P(\mathbf{y}^{(t)} \mid \mathbf{x}^{(t)}, \theta_i)$ é gaussiana (JORDAN & JACOBS, 1994):

$$h_i^{(t)} = \frac{g_i^{(t)} \exp\left\{-\frac{1}{2}\left(\mathbf{y}^{(t)} - y_i^{(t)}\right)^T \left(\mathbf{y}^{(t)} - y_i^{(t)}\right)\right\}}{\sum_{j=1}^{m} g_j^{(t)} \exp\left\{-\frac{1}{2}\left(\mathbf{y}^{(t)} - y_j^{(t)}\right)^T \left(\mathbf{y}^{(t)} - y_j^{(t)}\right)\right\}}. \tag{3.21}$$

Esta é uma medida da distância normalizada que reflete a magnitude relativa dos resíduos $\left(\mathbf{y}^{(t)} - y_i^{(t)}\right)$. Se o resíduo para o especialista *i* é pequeno em relação aos resíduos dos demais especialistas, então $h_i^{(t)}$ é grande. Caso contrário, $h_i^{(t)}$ é pequeno. Observe que, além disto, os $h_i^{(t)}$ (i=1,...,m) são não-negativos e somados produzem sempre o valor unitário, para cada $\mathbf{x}^{(t)}$. Isto implica que o crédito é distribuído para os especialistas de uma maneira competitiva.

3.5.3.2 Treinamento desacoplado via o método EM (*Expectation-Maximization*)

Partindo da equação (3.15) é possível utilizar o algoritmo de maximização da esperança (EM). Nesta abordagem, o ajuste dos parâmetros da rede *gating* e dos especialistas compreende dois passos bem definidos e engloba todo o conjunto de treinamento. A idéia é que a maximização da função de verossimilhança pode ser simplificada se cada padrão puder ser associado a exatamente um único especialista (indicado por variáveis chamadas de variáveis ausentes, que serão iguais ao valor de um para um especialista e zero para os

demais). As variáveis ausentes serão importantes para simplificação da otimização do logaritmo da verossimilhança. Os passos são os seguintes:

Passo E: É calculado o valor esperado para as variáveis ausentes (considerando que os parâmetros de todos os especialistas são conhecidos):

$$h_i^{(k)}(\mathbf{y}^{(t)} \mid \mathbf{x}^{(t)}) = \frac{g_i^{(k)}(\mathbf{x}^{(t)}, \mathbf{v})\phi_i(\mathbf{y}^{(t)} \mid \mathbf{x}^{(t)})}{\sum_{j=1}^{m} g_j^{(k)}(\mathbf{x}^{(t)}, \mathbf{v})\phi_j(\mathbf{y}^{(t)}, \mathbf{x}^{(t)})}, \tag{3.22}$$

onde $h_i^{(k)}(\mathbf{y}^{(t)} \mid \mathbf{x}^{(t)})$ são os valores das variáveis ausentes para o *t*-ésimo padrão de treinamento e o índice *k* indica o *k*-ésimo Passo E.

Passo M: O termo a ser maximizado pela rede *gating* será:

$$E_{gate} = \sum_{t=1}^{N}\sum_{i=1}^{m} h_i^{(k)}(\mathbf{y}^{(t)}, \mathbf{x}^{(t)}) \log(g_i(\mathbf{x}^{(t)})), \tag{3.23}$$

e para o especialista *i* será:

$$E_{expert} = \sum_{t=1}^{N}\sum_{i=1}^{m} h_i(\mathbf{y}^{(t)}, \mathbf{x}^{(t)}) \log(\phi_i(\mathbf{y}^{(t)}, \mathbf{x}^{(t)})), \tag{3.24}$$

onde ϕ é a densidade de probabilidade condicional:

$$\phi_i(\mathbf{y}^{(t)} \mid \mathbf{x}^{(t)}) = \frac{1}{(2\pi)^{d/2}\sigma_i} \exp\left\{-\frac{(\mathbf{y}^{(t)} - \mathbf{y}_i^{(t)})^T(\mathbf{y}^{(t)} - \mathbf{y}_i^{(t)})}{2\sigma_i^2}\right\}, \tag{3.25}$$

d é a dimensão de $\mathbf{y}^{(t)}$, e $\mathbf{y}_i^{(t)}$ é a saída do especialista *i* para o *t*-ésimo padrão de treinamento.

As *m* saídas da rede *gating* são dadas pela função *softmax*, da equação (3.12) com $\xi_i = g_i$.

Ao invés de usar uma rede *gating* com função de ativação *softmax* (conforme definido acima), uma outra abordagem é utilizar funções gaussianas normalizadas, cada uma centrada na região de atuação de um especialista. São conhecidas como MEs locais (LME, do inglês *local mixture of experts*), pois o espaço de entrada é dividido por hiper-elipsóides, facilitando a contribuição de vários especialistas para uma sub-região. As saídas (α_i e α_j) da rede *gating* assumirão uma nova forma ao passar pela nova função *softmax* dada por:

$$g_i(\mathbf{x}) = \frac{\alpha_i P(\mathbf{x} \mid \mathbf{v}_i)}{\sum_{j=1}^{m} \alpha_j P(\mathbf{x} \mid \mathbf{v}_j)}, \tag{3.26}$$

$$P(\mathbf{x} \mid \mathbf{v}_i) = (2\pi)^{-n/2} \left|\Sigma_i\right|^{-1/2} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{r}_i)^T \Sigma_i^{-1} (\mathbf{x}-\mathbf{r}_i)}. \tag{3.27}$$

Assim, a influência do *i*-ésimo especialista está localizada numa região ao redor de $\mathbf{r}_i$ (XU *et. al.*, 1995; RAMAMURTI & GHOSH, 1996).

3.5.4 Predição de séries temporais usando mistura de especialistas heterogêneos

Ao invés de utilizar o mesmo modelo, por exemplo, RNAs do tipo MLP, para todos os especialistas, PUMA-VILLANUEVA *et al*. (2005) propôs utilizar modelos diferentes para cada especialista. Esta nova abordagem foi chamada de mistura de especialistas heterogêneos. Para apontar a viabilidade desta abordagem, foi realizado um estudo de caso envolvendo MEs com especialistas lineares e não-lineares e redes *gating* lineares ou não-lineares. Visando encontrar uma configuração mais adequada e parcimoniosa para atender o problema de predição de séries temporais de natureza distintas, foram consideradas 10 séries temporais, sendo 6 séries reais e 4 séries artificiais. Cinco das seis séries temporais reais, já mencionadas no Capítulo 2, estão incluídas. Foram simuladas várias propostas de arquitetura de MEs para cada série temporal (ver Tabela 3.2).

Tabela 3.2 – Configurações de MEs testadas.

Tipo de MEs	Modelo	
	Especialistas	**Rede *Gating***
Homogêneo	2 Lineares	Linear
	2 Não-lineares	Não-linear
	3 Lineares	Linear
	3 Não-lineares	Não-linear
	5 Lineares	Linear
	5 Não-lineares	Não-linear
Heterogêneo	1 Linear e 1 Não-linear	Linear
	1 Linear e 1 Não-linear	Não-linear
	2 Lineares e 2 Não-lineares	Linear
	2 Lineares e 2 Não-lineares	Não-linear

No treinamento das MEs, foi utilizado o algoritmo EM, sendo que no caso dos especialistas e *gating* lineares a cada passo EM foi utilizado o algoritmo LMS, e no caso de especialistas e *gating* não-lineares (MLPs) foi empregado o método do gradiente simples. Da mesma forma que nos experimentos envolvendo *ensemble*, o conjunto de dados foi dividido em 75% para treinamento e validação e 25% para teste. Foram realizadas 30 execuções, a média, desvio padrão, o maior MAE e menor MAE são apresentados na Tabela 3.3. A título de comparação, são apresentados também na Tabela 3.3 os resultados utilizando uma única MLP (com critério de parada antecipada e número máximo de épocas igual a 500) e o melhor *ensemble* produzido. Em todos os casos, as MEs obtiveram melhores resultados, tanto na média como no valor mínimo do MAE.

Tabela 3.3 – Resultados comparativos sobre o conjunto de teste: Única MLP, melhor Ensemble e MEs Heterogêneos.

SÉRIE	Estatísticas	MLP	Melhor *Ensemble*	Mistura de Especialistas Heterogêneos
Book store	**Média**	**84,82**	**77,38**	**67,72**
	STD	14,95	5,7	3,33
	Max	121,85	89,92	81,05
	Min	69,08	67,43	62,85
Clothing store	**Média**	**781,35**	**607,8**	**514,61**
	STD	437,18	201,03	134,2
	Max	1767,9	1170,6	1142,15
	Min	477,71	411,29	378,99
Furniture store	**Média**	**176,44**	**161,85**	**156,33**
	STD	32,77	7,07	16,43
	Max	310,36	181,34	181,18
	Min	155,78	143,67	131,36
Hardware store	**Média**	**42,68**	**39,66**	**38,75**
	STD	8,89	2,06	3,8
	Max	73,92	44,33	48,73
	Min	34,12	36,66	33,07
Durable Consumer Goods	**Média**	**4,11**	**3,59**	**3,54**
	STD	0,32	0,18	0,24
	Max	4,74	4,33	4,18
	Min	3,57	3,33	3,02

Na Tabela 3.4, apresentam-se as configurações de mistura de especialistas que foram finalmente consideradas para cada série temporal.

Tabela 3.4 – Configurações mais adequadas para as séries temporais consideradas.

Série Temporal	Modelo	
	Especialistas	**Rede *Gating***
Book store	1 Linear e 1 Não-linear	Linear
Clothing store	1 Linear e 1 Não-linear	Não-linear
Furniture store	1 Linear e 1 Não-linear	Não-linear
Hardware store	5 Lineares	Linear
Durable Consumer Goods	2 Lineares	Linear

A Figura 3.17 apresenta uma visualização gráfica, utilizando *boxplot*, dos resultados apresentados na Tabela 3.3. A média é representada pelo asterisco e os valores em porcentagem representam a diminuição percentual do MAE em relação a uma única MLP.

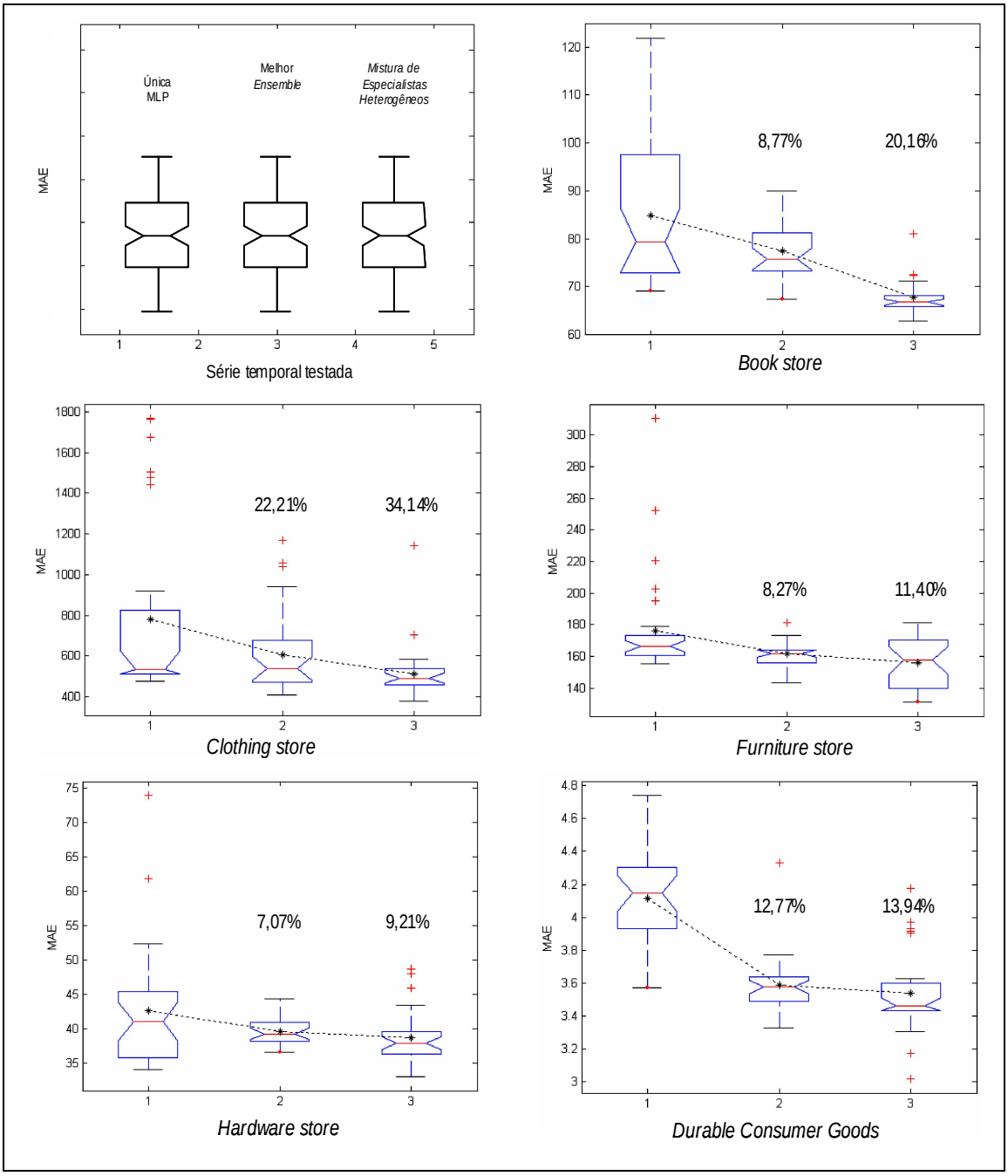


Figura 3.17 – Gráficos *boxplot* a partir da Tabela 3.3: MLP vs *Ensemble* vs MEs.

3.5.5 Outras propostas para ajuste de parâmetros em MEs

Existem na literatura outras propostas de ajuste de parâmetros para MEs, a saber: computação evolutiva e métodos bayesianos.

3.5.5.1 Computação evolutiva

KARRAS *et al*. (2004) propôs algoritmos genéticos para otimizar a estrutura de uma mistura de especialista hierárquica, a qual é uma extensão de MEs e que pode ser vista como uma árvore em que cada nível de especialistas contém uma rede *gating*. Os parâmetros a serem ajustados foram: a profundidade da árvore e o número de elementos em cada nível da árvore. Para cada conjunto desses valores, foi utilizado o método EM para treinar os especialistas e as redes *gating*. Eles também utilizaram técnicas de busca local chamadas de "*growing tree*" e "*growing chromosomes*", visando diminuir o tempo de processamento.

3.5.5.2 Métodos Bayesianos

UEDA & GHAHRAMANI (2000) apresentaram um algoritmo para inferir os parâmetros e a estrutura de um modelo de ME baseado numa abordagem variacional bayesiana, via maximização de uma proposta de função-objetivo. Esses autores apresentaram também um algoritmo determinístico para estimar o número de especialistas.

Capítulo 4

Seleção de variáveis e predição de séries temporais

Resumo: Este capítulo visa apresentar o uso de seleção de variáveis no contexto do problema de predição de séries temporais. Num primeiro momento, apresenta-se uma visão do estado da arte da área de seleção de variáveis. Em seguida, particulariza-se o uso de seleção de variáveis junto ao problema de predição de séries temporais, descrevendo duas técnicas específicas para este fim. Finalmente, são apresentados resultados comparativos empregando essas duas técnicas para as cinco séries temporais financeiras que vêm acompanhando esta tese.

4.1. Introdução

Esse capítulo se inicia com as definições de “variáveis” e “características”. Alguns autores fazem diferencia nestas duas palavras e acostumam chamar de variáveis às entradas originais, ou seja, aquelas que não sofreram nenhum tipo de transformação. E chamam-se de características as variáveis construídas com base nas variáveis de entrada originais (GUYON & ELISSEEFF, 2003). Embora esta distinção seja feita aqui, em grande parte da literatura elas são consideradas como se fossem sinônimos de dados de entrada.

4.1.1. Extração de características e seleção de variáveis

A **extração de características** requer uma transformação das variáveis de entrada originais, produzindo características que podem ser interpretadas como um conjunto de

novas variáveis habitando um novo espaço, sendo que o novo espaço tem normalmente uma dimensão menor que o espaço original.

Formalmente, dado um espaço **X** de dimensão *n*, tem-se:

$$\Omega : \mathbf{X} \rightarrow \mathbf{F}, \tag{4.1}$$

onde **F** possui dimensão *m*, geralmente com $m < n$. Dessa forma, para um padrão de entrada $\mathbf{x} \in \mathbf{X}$,

$$\Omega(\mathbf{x}) = \mathbf{x'} \tag{4.2}$$

onde $\mathbf{x'} \in \mathbf{F}$ é a nova representação do padrão **x** no espaço **F**. Para maiores informações de métodos de extração de características, veja KUSIAK (2001) e LIU & MOTOLA (1998).

A **seleção de variáveis** refere-se à identificação daquele subconjunto de variáveis úteis para obter bons resultados no reconhecimento de padrões ou em regressão de dados.

Dado um conjunto **V** que representa o espaço de variáveis { $\boldsymbol{v_i}$ }, i= 1, 2,..., L; uma busca exaustiva entre as L variáveis demandaria avaliar 2^L-1 subconjuntos de variáveis. Por exemplo, para $\mathbf{V} = \{\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}\}$, onde L=3, tem-se $2^3-1 = 7$ subconjuntos de variáveis, conforme mostra a Tabela 4.1.

Tabela 4.1 - Possíveis subconjuntos para L=3.

No.	Subconjunto de variáveis que podem ser selecionadas
1	$\boldsymbol{v_1}$
2	$\boldsymbol{v_2}$
3	$\boldsymbol{v_3}$
4	$\boldsymbol{v_1}, \boldsymbol{v_2}$
5	$\boldsymbol{v_1}, \boldsymbol{v_3}$
6	$\boldsymbol{v_2}, \boldsymbol{v_3}$
7	$\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}$

A Tabela 4.2 apresenta o número de subconjuntos possíveis para valores arbitrários de *L*. Em problemas de reconhecimento de padrões, geralmente o número de variáveis é da ordem de centenas. Um exemplo em bioinformatica típico encontra-se em ALON *et al*. (1999), que tem 2000 variáveis, correspondente a 2000 genes. Trata-se de um problema com 62 amostras de classificação de câncer de colo, sendo que 40 amostras correspondem a pacientes com câncer e 22 amostras a pacientes sem câncer. Com base na relação entre número de variáveis e número de amostras, justifica-se a aplicação de técnicas de seleção de variáveis.

Tabela 4.2 - Número de subconjuntos candidatos para valores de L arbitrários.

L	Número de Subconjuntos (2^L-1)
10	1023
20	1048575
30	1,073741823000000e+009
40	1,099511627775000e+012
50	1,125899906842623e+015
100	1,267650600228229e+030
1000	1,071508607186267e+301

As razões que levam a utilizar métodos de seleção de variáveis são as seguintes, sendo que as duas primeiras estão relacionadas ao modelo, seja este um classificador ou um regressor:

- Melhorar o desempenho;
- Diminuir o esforço computacional;
- Promover um melhor entendimento do processo gerador dos dados.

É necessário ter em conta os seguintes critérios relacionados à seleção de variáveis (YU & LIU, 2004):

- **Relevância:** A idéia de relevância está associada à importância que as variáveis podem ter para o problema, já que esta informação servirá de guia no processo de seleção. Por exemplo, no caso da abordagem baseada em filtro, a relevância seria medida por alguma forma de correlação entre as variáveis de entrada com a variável

de saída desejada. Já no caso da abordagem baseada em envoltório, a relevância seria medida por algum critério específico, por exemplo, taxa de redução do *MAE*.

- ✓ *Relevância forte*: significa que a remoção da variável ocasiona uma degradação significativa no desempenho do modelo.

- ✓ *Relevância fraca*: significa que a remoção da variável ocasiona nenhuma ou uma baixa degradação no desempenho do modelo.

JOHN *et al.* (1994) apresentaram até quatro definições de relevância, sendo que a seguir será descrita uma delas. Considere uma matriz **X** que representa os dados de entrada, e a matriz **Y** a saída desejada. Desse modo, cada linha de **X** está associada a uma linha de **Y**. Cada vetor-coluna de **X** representa uma variável de entrada. Assim, a variável X_i é dita ser relevante se, e somente se, existe algum par $\{x_j, y_j\}$ para o qual $p(X_i = x_j) > 0$ e $p(Y = y_j \,/\, X_i = x_j) \neq p(Y = y_j)$. Em outras palavras, a probabilidade de obter y_j na saída está condicionada a ter x_j na entrada.

- **Redundância:** Duas variáveis são redundantes se seus valores são completamente correlacionados. Portanto, o nível de correlação indica o grau de redundância.

- **Otimalidade:** Para um subconjunto de variáveis ser chamado de ótimo, não deveria existir um outro subconjunto que produza um melhor resultado do que o resultado alcançado por este subconjunto.

Relevância não implica otimalidade. O fato de uma variável ser relevante não implica que esta necessariamente deva fazer parte do subconjunto ótimo. Por exemplo, nos casos de redundância ou presença de ruído, podem existir variáveis que não fazem parte do conjunto ótimo.

***Otimalidade não implica relevância*.** O fato de uma variável estar no subconjunto ótimo não implica que esta seja relevante. Por exemplo, a entrada fixada em um (*bias*) numa MLP não atende a várias definições de relevância, mas sua remoção pode comprometer o desempenho.

As relações entre as definições dadas de variáveis relevantes e redundantes são mostradas de forma gráfica na Figura 4.1, onde um subconjunto de variáveis ótimas estaria composto pelas partes C+D. Mesmo que as partes B e C sejam disjuntas, isto não implica que certos elementos de B não possam fazer parte do subconjunto ótimo (YU & LIU, 2004).

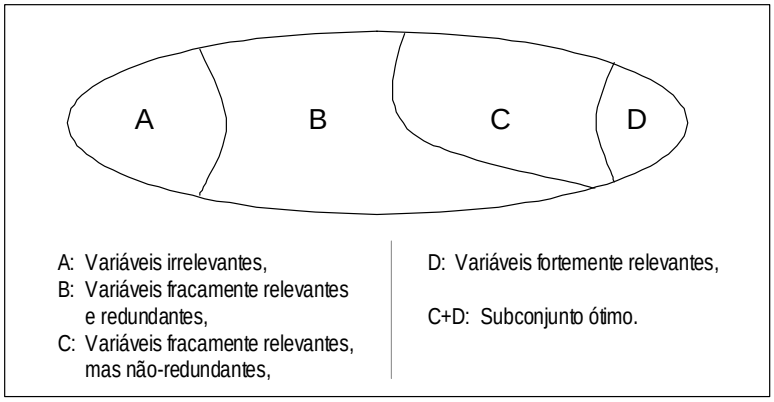


Figura 4.1 - Representação gráfica do espaço das variáveis relevantes e redundantes.

4.1.2. Extração e seleção em predição de séries temporais

Com base nas considerações da seção anterior, podemos aplicar as abordagens de extração de características e seleção de variáveis em predição de séries temporais da seguinte forma:

Extração de características:

Como já foi definido no Capítulo 2, uma série temporal é vista como uma única variável indexada no tempo. Assim ela reside no espaço $\Re^1$. As características que vamos extrair a partir dela residirão num espaço $\Re^L$ com $L \geq 1$. Na Figura 4.2, ilustra-se o resultado deste processo, que seria a criação dos conjuntos de entrada **X** e saída **Y** para um modelo regressor. Notar que a predição é de um passo à frente. As entradas para este processo de extração de características seriam duas:

- A série temporal: $S \in \Re^1$;

- O comprimento da janela de atrasos: *L*.

O cálculo deste parâmetro *L* envolve o uso da função de auto-correlação da série, caso ela seja linear. Caso a série seja não-linear, são necessárias outras formas de medir correlação não-linear, por exemplo, medida de correlação via informação mútua (FRASER & SWINNEY, 1986). No Anexo-A, apresentam-se o cálculo de valores para *L*, via essas duas técnicas.

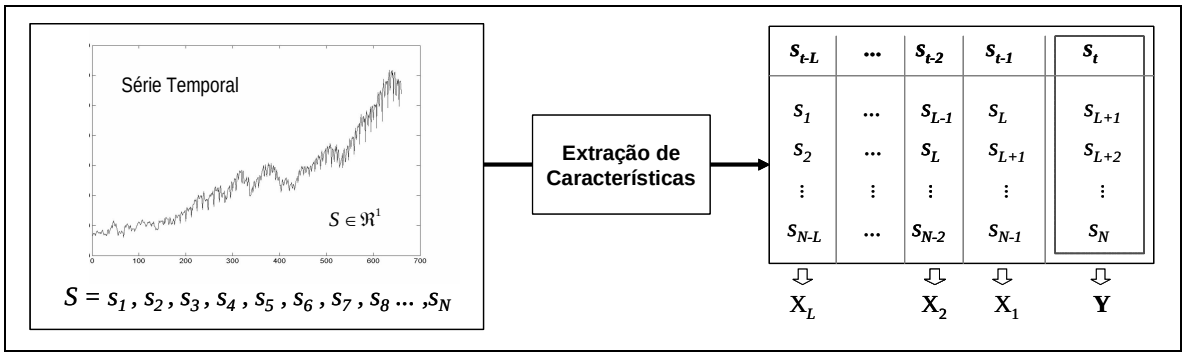


Figura 4.2 - Extração de características em predição de séries temporais.

Seleção de variáveis:

Da Figura 4.2, podemos ver que agora o espaço de entradas **X** é o $\Re^L$. Assim, temos as variáveis representadas pelos vetores-coluna X_l, l=1,...,L. O processo de seleção de variáveis irá selecionar dentre as variáveis X_l, l=1,...,L, um subconjunto que permita alcançar um melhor desempenho na predição dos valores futuros da série. Este processo é ilustrado na Figura 4.3, onde como conseqüência da seleção de variáveis partindo de $\mathbf{X} \in \Re^L$, obtém-se um novo conjunto de variáveis $\hat{\mathbf{X}}$ no espaço $\Re^K$, onde $K \leq L$.

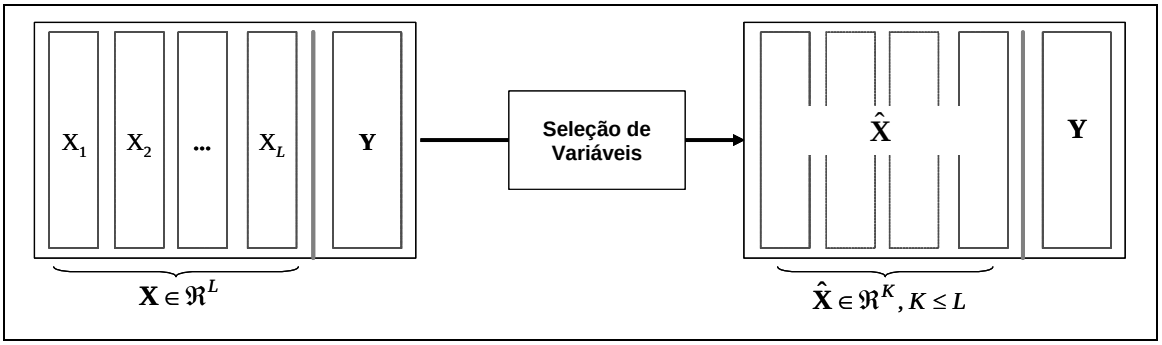


Figura 4.3 - Seleção de variáveis em predição de séries temporais.

4.2. Abordagens em seleção de variáveis

Atualmente, existe uma taxonomia ou classificação mais detalhada dos métodos de seleção de variáveis. Há alguns anos atrás, havia apenas uma distinção na literatura entre métodos livres de modelo (*model free*) e baseados em modelo (*model based*). Livre de modelo é quando o método de seleção de variáveis é realizado num passo anterior à síntese do modelo classificador ou regressor. No caso baseado em modelo, o método de seleção de variáveis opera em interdependência com a síntese do modelo.

Esta classificação mudou desde o trabalho de KOHAVI & JOHN (1997), onde apresenta-se as seguintes abordagens: Filtro, Envoltório (*Wrapper*) e Embutido (*Embedded*). Por outra parte, existem trabalhos em que combinam as abordagens Filtro e Envoltório, criando uma nova proposta híbrida. A seguir apresenta-se uma breve explicação de cada uma destas abordagens.

4.2.1. Filtro

Filtro é uma abordagem equivalente a métodos livres de modelo, conforme apresentado na Figura 4.4. Dado que não há acesso à avaliação de desempenho do modelo com o subconjunto de variáveis, a alternativa seria utilizar medidas de associação ou de correlação

entre variáveis. Para problemas que envolvam aprendizado não-supervisionado, como agrupamento, essas medidas envolvem apenas as variáveis de entrada. Já no caso de problemas de aprendizado supervisionado, classificação e regressão, as medidas de correlação envolvem as variáveis de entrada e a(s) variável(is) de saída desejada.

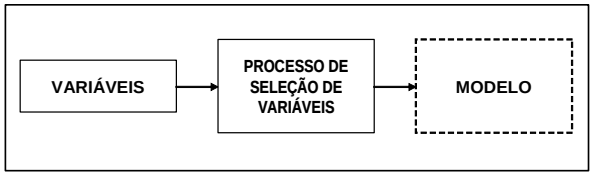


Figura 4.4 - Seleção de variáveis: Filtro.

O sucesso deste tipo de método para seleção de variáveis está em utilizar uma medida de correlação linear/não-linear robusta, assim como uma estratégia de busca capaz de fugir de mínimos locais.

Vantagem:

- É um procedimento rápido, dado que é independente da síntese do modelo.

Desvantagem:

- Dado que trata-se de um passo prévio à síntese do modelo, não só pode não alcançar o critério de otimalidade como também o desempenho final pode não ser o esperado.

Recomendação:

- Uso promissor em problemas com grande quantidade de dados disponíveis e de alta dimensão. Exemplo: alguns casos de problemas de bioinformática e mineração de dados.

É possível utilizar este método em predição de séries temporais, desde que se disponha de uma quantidade considerável de observações da série temporal. Medidas de correlação não-linear baseadas em critérios de entropia e informação mútua são amplamente utilizadas na literatura em diversas aplicações, como alternativa à função de auto-correlação (FLEURET,

2004; WANG & LOCHOVSKY, 2004). Essas abordagens inicialmente foram concebidas para medir a quantidade de informação que é transmitida por um canal de comunicação. Posteriormente, essas idéias foram utilizadas em seleção de variáveis e estão fundamentadas na Teoria da Informação (COVER & THOMAS, 1991; FRASER & SWINNEY, 1986; HAMMING, 1986).

4.2.2. Envoltório (*Wrapper*)

Na abordagem Envolutório (do inglês *wrapper*), existe uma interação do mecanismo de seleção de variáveis com um modelo, seja este para classificação ou regressão de dados. A utilidade do modelo, uma vez ajustado, será a de avaliar ou dar uma nota de desempenho a cada subconjunto de variáveis que será formado para resolver o problema. A Figura 4.5 ilustra esta idéia.

A Seção 4.3 é dedicada principalmente a descrever dois métodos específicos de seleção de variáveis na abordagem de Envoltório, a qual utiliza uma RNA do tipo MLP como modelo regressor.

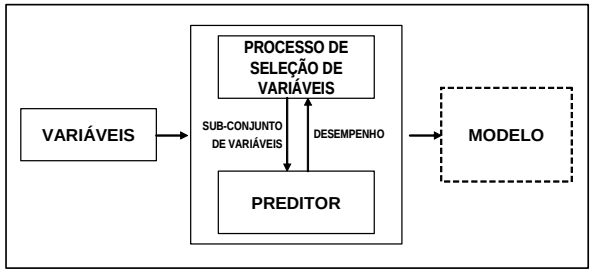


Figura 4.5 - Seleção de variáveis: Envoltório.

Vantagem:

- Por interagir com um modelo ajustado, a qualidade das variáveis selecionadas tende a superar aquelas obtidas por meio de Filtro.

Desvantagem:

- O processo pode ser demorado e computacionalmente mais custoso, pois envolve o custo de se treinar o modelo para cada subconjunto de variáveis que é tomado como candidato.

Recomendação:

- É mais recomendado para casos em que o número de amostras seja reduzido. Exemplo: séries temporais curtas, alguns casos de problemas de bioinformática e mineração de dados.

4.2.3. Embutido (*Embedding*)

A classificação de Embutido (do inglês *embedding*), considera que o processo de seleção de variáveis está implícito no processo de síntese do modelo classificador ou regressor. A Figura 4.6 ilustra esta abordagem.

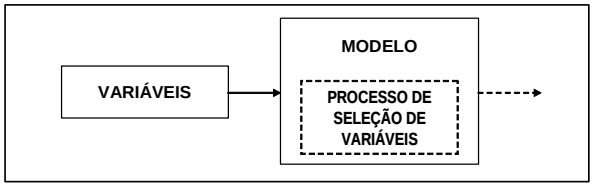


Figura 4.6 - Seleção de variáveis: Embutido.

Um exemplo de método embutido é CART (BREIMAN *et al.*, 1984), que tem um mecanismo de seleção de variáveis (GUYON & ELISSEEFF, 2003). Outra abordagem nesta vertente envolve as máquinas de vetores suporte, onde a seleção de variáveis sai como resultado da própria resolução do problema de programação quadrática associado (RAKOTMAMONJY, 2003).

4.2.4. Método híbrido

Nesta proposta, tenta-se explorar os benefícios das abordagens Filtro e Envoltório. Como anteriormente descrito, o Filtro requer menos tempo e esforço computacional, mas, por utilizar medidas de correlação, os resultados podem não ser tão satisfatórios quanto Envoltório, que é mais custoso computacionalmente. Assim, combinando ambos, Filtro para explorar entre possíveis subconjuntos candidatos e Envoltório para a seleção propriamente dita, podem ser obtidos resultados mais interessantes. Em DAS (2001) e LIU & YU (2005) são encontrados detalhes desta abordagem híbrida.

4.3. Seleção de variáveis em predição de séries temporais via o método envoltório

Como explicado no Capítulo 2, uma série temporal pode ser representada por uma variável aleatória dependente do tempo, $x(t)$, tal que o valor da variável no tempo presente pode ser expresso em função de valores da mesma em instantes passados:

$$x_t = F(x_{t-1}, x_{t-2}, x_{t-3}, \ldots, x_{t-L}), \tag{4.3}$$

onde L representa o número máximo de variáveis atrasadas que são utilizadas para aproximar o valor presente $x(t)$. O Anexo-A apresenta algumas estratégias para se estimar o valor de L.

Partindo das variáveis x_{t-1}, x_{t-2}, x_{t-3}, …, x_{t-L} da equação (4.3), esta seção apresenta dois métodos de seleção de variáveis pertencentes à abordagem Envoltório, considerando uma RNA do tipo MLP como modelo regressor.

4.3.1. Aspectos a considerar na abordagem Envoltório

Num nível mais detalhado, a abordagem Envoltório considera os seguintes aspectos para a sua implementação:

Variáveis candidatas: Variáveis candidatas a comporem o subconjunto de variáveis selecionadas, ou seja, as variáveis **X** na Figura 4.3.

Critério de busca: Refere-se à estratégia que guiará a busca no espaço de possíveis subconjuntos de variáveis $\hat{\mathbf{X}}$, onde $\hat{\mathbf{X}} \subset \mathbf{X}$. Essas estratégias envolvem idéias de vizinhança, heurísticas de inserção e poda, ou meta-heurísticas mais elaboradas, como algoritmos evolutivos e busca tabu. Se a dimensão de **X** for pequena, pode-se utilizar uma busca exaustiva (ver Tabela 4.2). Neste trabalho, foram utilizadas heurísticas de inserção (método construtivo) e de poda.

Função para avaliar as variáveis em curso: Este ponto guarda certa relação com o anterior. O critério de busca para tomar as decisões de considerar ou descartar uma variável conta com uma função que avalie o desempenho de cada subconjunto de variáveis sob análise. Na maioria dos casos, esta função coincide com a função que guia a síntese do modelo.

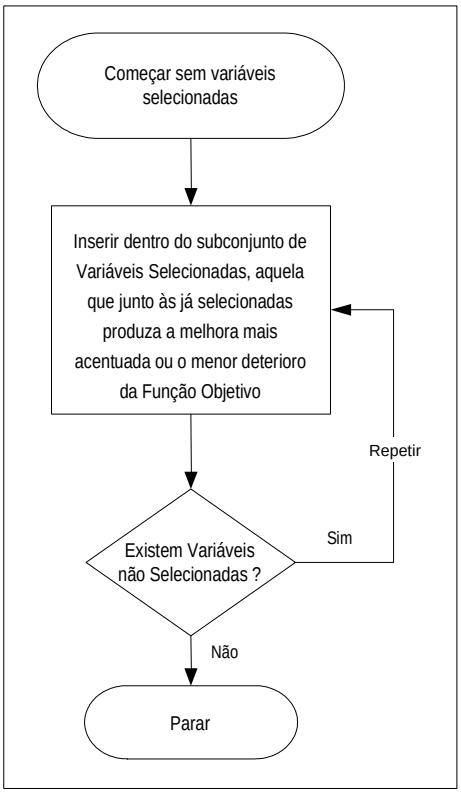
Estrutura do modelo: Modelo que será ajustado ao problema, e servirá como avaliador dos subconjuntos de variáveis. Neste trabalho, foi considerada uma RNA do tipo MLP.

Algoritmo de síntese ou treinamento do modelo: Relacionado ao treinamento do modelo, no caso foi utilizado o algoritmo do gradiente, descrito no Capítulo 2.

4.3.2. Seleção Progressiva (SP)

Como já foi definida na seção 4.1, a seleção de variáveis via busca exaustiva tem um custo computacional do tipo exponencial. Nesta seção, apresenta-se uma heurística de inserção, com um custo computacional de ordem $O(L^2)$, onde L é o número de variáveis. Dependendo da função que mede a qualidade do subconjunto de variáveis selecionadas,

esta abordagem pode até tratar de conjuntos com um número de variáveis da ordem de centenas.



Algoritmo 4.1 - Seleção de variáveis: fluxograma do método de Seleção Progressiva – SP.

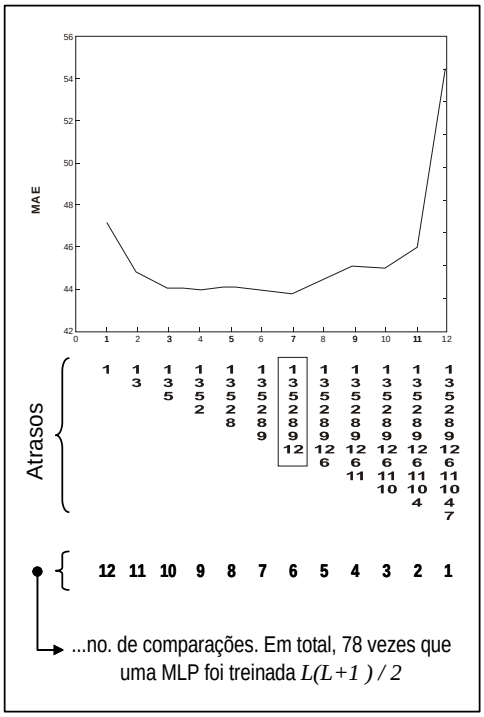


Figura 4.7 - Exemplo da execução da Seleção Progressiva, com L=12. Os dados correspondem à série *Furniture store* apresentada no Capítulo 2.

Tenta-se construir um subconjunto de variáveis partindo do conjunto vazio. Neste caso, considerando uma a uma, a estratégia é comparar todas as possíveis variáveis e selecionar a variável que produza a melhora mais acentuada ou o menor deterioro do valor da função objetivo (*best improvement*), no caso a função que calcula o MAE entre os valores desejados da série e os valores obtidos pelo modelo. Usam-se aqui as amostras do conjunto de validação. O Algoritmo 4.1 apresenta o fluxograma do método.

O conjunto de atrasos selecionado é aquele que, no decorrer do processo, apresentar menor erro. Isto é mostrado na Figura 4.7, onde o eixo horizontal indica o número iterações do

algoritmo, o qual, quando executado de forma completa coincide com o número de variáveis do conjunto de variáveis inicialmente disponíveis. Já o eixo vertical indica o valor MAE correspondente para cada um dos subconjuntos sendo construídos. Na iteração 7 ou quando o número de variáveis em construção foi de 7 (variáveis dentro do retângulo vertical) o valor do MAE experimentou o menor do processo. Os valores abaixo de cada subconjunto de variáveis representam o número de comparações realizadas para escolher esse subconjunto, que também corresponde ao número de redes MLP que devem ser treinadas. Esse número obedece à seguinte fórmula: $L(L+1)/2$, onde L é o número de variáveis candidatas, no início do processo. No exemplo da Figura 4.7, para L=12 foram 78 redes MLP treinadas.

Vale indicar outros aspectos relativos à Figura 4.7, como por exemplo:

- É possível parar o processo quando se chega numa iteração na qual a seleção de uma nova variável dentro do subconjunto de variáveis em construção deteriora ao valor da função objetivo em vez de melhorá-la (nesse caso, o critério de seleção deixaria de ser a variável que "produz a melhora mais acentuada", passando a ser "a que menos a deteriore"). No entanto, a busca via estratégia construtiva esta sujeita a cair em mínimos locais e não se descarta a possibilidade de, numa iteração posterior, poder experimentar um ganho maior ainda no valor da função-objetivo. Em conseqüência, recomenda-se executar o processo de forma completa.
- Caso a curva da figura experimente a cada iteração redução em seus valores (valores que favorecem o critério de melhora da função-objetivo), configura-se um caso em que todas as variáveis inicialmente disponíveis são de importância e nenhuma delas deve ser descartada na síntese de um preditor.

Outros métodos similares

Outras formas que seguem os requisitos da Seleção Progressiva são: (*i*) usando o critério de *poda*, onde em vez de partir com o subconjunto de variáveis selecionadas em vazio, o subconjunto conteria inicialmente todas as variáveis candidatas. Em cada passo, a variável que conduz a uma melhora mais acentuada do erro é retirada. (*ii*) É possível também usar

ambos os critérios, por exemplo, num passo *inserção* de duas variáveis e, no passo seguinte, *poda* de uma variável.

4.3.3. Seleção via Poda Baseada em Sensibilidade (PBS)

A Poda Baseada em Sensibilidade (PBS) é conhecida na literatura como *Sensitivity Based Pruning* (*SBP*). Foi proposta originalmente por MOODY & UTANS (1991) e formalizada em MOODY (1994). O objetivo era encontrar uma arquitetura de rede neural (MLP) que fosse a mais parcimoniosa possível.

PBS é menos custosa computacionalmente do que Seleção Progressiva, pois o número de vezes que uma MLP é treinada é reduzido a *L*.

O Algoritmo 4.2 apresenta os passos a serem seguidos para a implementação deste método. Vale notar que, ao se referir à retirada de uma entrada da rede, a operação efetivamente realizada é a fixação do valor da entrada em seu valor médio, conforme equações (4.4) e (4.5).

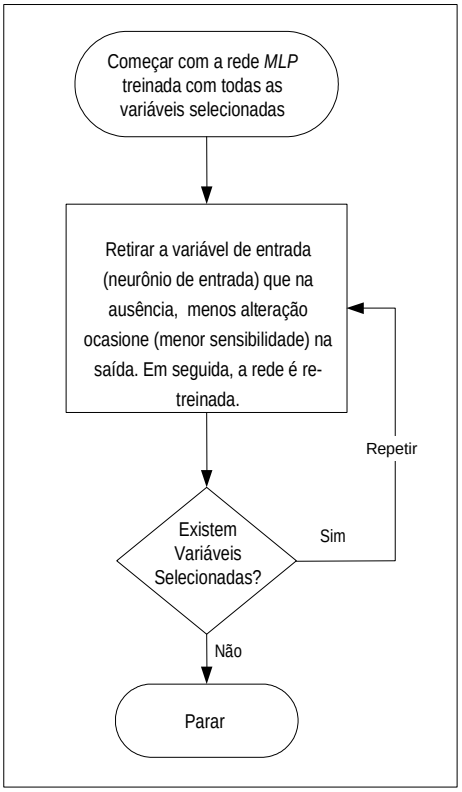
E_c é a sensibilidade ocasionada pela remoção de x_c e é obtida na forma:

$$E_c = E(\bar{x}_c) - E(x_c)\,, \tag{4.4}$$

onde $E(x_c)$ é o erro quando a variável x_c é considerada, e $E(\bar{x}_c)$ é o erro quando a variável x_c é substituída pelo seu valor médio $\bar{x}_c$, conforme equação (4.5) a seguir:

$$\bar{x}_c = \frac{1}{N}\sum_{i=1}^{N} x_{ci}\,, \tag{4.5}$$

onde *N* é o número de amostras.



Algoritmo 4.2 - Seleção de variáveis: fluxograma do método de Poda Baseada em Sensibilidade – PBS.

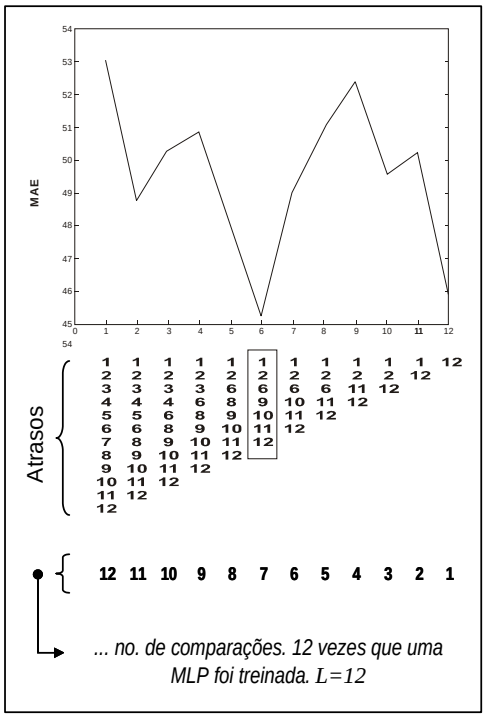


Figura 4.8 - Exemplo da execução de PBS, com L=12. Os dados correspondem à série *Furniture store* apresentada no Capítulo 2.

Na Figura 4.8, o número de comparações realizadas foi 78, mas apenas 12 redes neurais MLP foram treinadas. Auxiliados pelo Algoritmo 4.2, vemos que, na primeira iteração, a MLP é treinada com todas as variáveis de entrada. Em seguida, realizam-se comparações até encontrar a variável menos sensível. Uma vez que a variável foi encontrada, esta é retirada (retirar uma variável é equivalente a fixá-la no seu valor médio) e a rede neural é re-treinada. Repete-se, então, o processo até que todas as variáveis sejam retiradas. Assim, o número de vezes que é preciso re-treinar uma MLP é igual ao número de variáveis iniciais (L).

Outras considerações em relação a este algoritmo podem ser citadas, dentre elas:

- Comparando este algoritmo com o SP, além de terem pontos de partida e finais opostos, o critério de escolha para a poda é diferente também: em SP, busca-se pela variável que produza melhora mais acentuada ou a menor deterioração do valor da função-objetivo; já em PBS, simplesmente busca-se pela variável menos sensível, independente de ganho ou deterioração do valor da função-objetivo.
- PBS, da mesma forma que SP, está sujeita a mínimos locais, e recomenda-se a execução completa do algoritmo antes de identificar o subconjunto de variáveis finais.

4.4. Simulações e resultados aplicando SP e PBS em predição de séries temporais

Esta seção visa atender dois objetivos:

- Estudo comparativo de desempenho entre Seleção Progressiva e Poda Baseada em Sensibilidade, quando aplicadas ao problema de predição das 5 séries temporais tomadas como casos de estudo e mostradas na Figura 4.9.
- Investigação da melhor forma de separar os dados para compor os conjuntos de Treinamento e Validação, visando a síntese do modelo preditor (veja Figura 4.10).

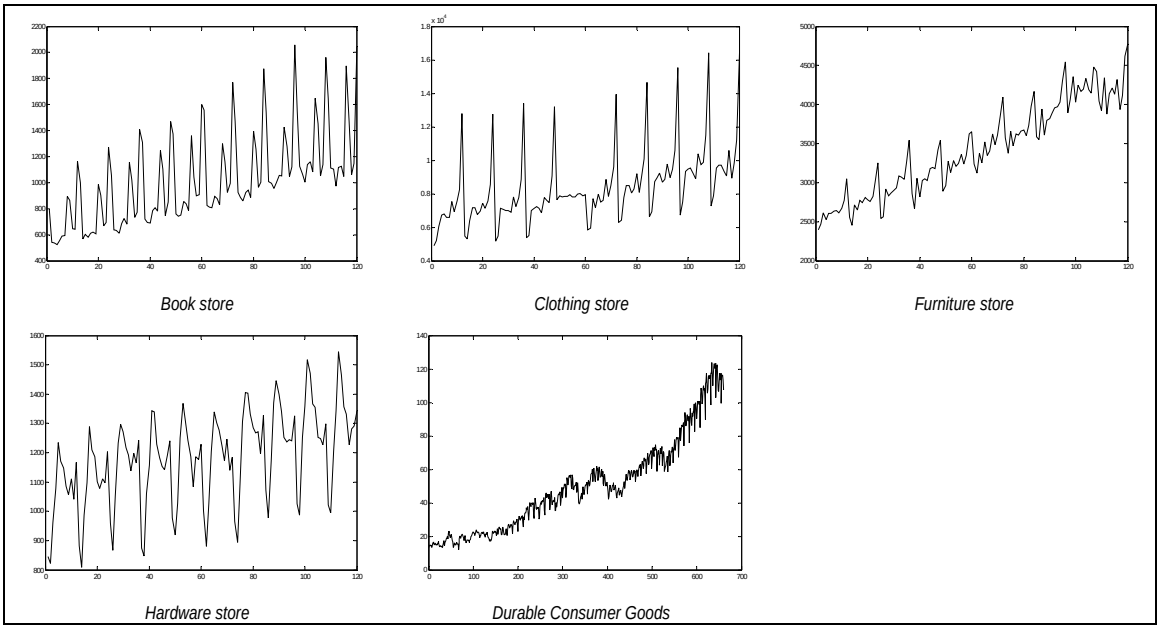


Figura 4.9 - Séries temporais consideradas.

A aplicação de métodos de seleção de variáveis ao problema de predição de séries temporais vai realizar a seleção dos atrasos, ou valores passados da série, que irão compor o vetor de entrada do preditor.

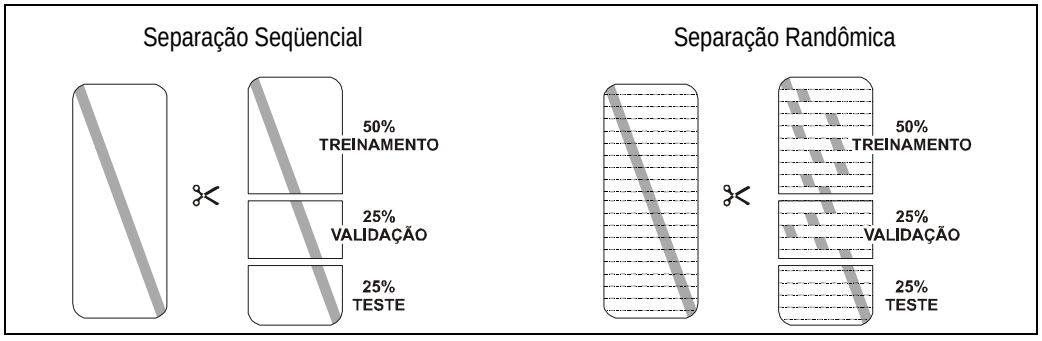


Figura 4.10 - Formas de separar os dados para compor os conjuntos de treinamento e validação

Tanto em SP como PBS, foi utilizada uma MLP como modelo preditor, e os parâmetros foram os seguintes:

Máximo número de atrasos, L:	12
Função-objetivo de seleção de variáveis:	MAE (acima do conjunto de validação)
Neurônios ocultos:	30
Máximo número de épocas no treinamento da MLP:	300
Algoritmo de treinamento da MLP:	Método do Gradiente
Validação cruzada (treinamento, validação e teste):	Seqüencial, (em porcentagens de 50, 25, 25, respectivamente), ver Figura 4.10. O subconjunto de teste não foi utilizado neste processo de seleção de variáveis.
Número de execuções de cada método:	30

4.4.1. Resultados obtidos

Na equação (4.6), é apresentada a configuração inicial, que indica a dependência de *L=12* valores atrasados da série *S*, para predizer um valor futuro.

$$\hat{s}_t = f(s_{t-1}, s_{t-2}, s_{t-3}, ..., s_{t-12}), \tag{4.6}$$

Na Tabela 4.3, é apresentado o resultado de 30 execuções de cada método de seleção de variáveis, PBS e SP, para cada uma das séries temporais testadas, com *L*=12. Constata-se que o atraso $s_{t\text{-}1}$ para a série temporal *Book store* foi selecionado nas 30 execuções, tanto pelo método PBS como por SP. Já o atraso $s_{t\text{-}7}$ foi ignorado por ambos os métodos.

Tabela 4.3 - Número de vezes em que cada atraso foi selecionado para compor o subconjunto de variáveis de entrada do preditor. Foram executados 30 vezes cada método, PBS e SP, para as 5 séries temporais testadas.

SÉRIE TEMPORAL	MÉTODO	ATRASOS DA SÉRIE (Variáveis, $L=12$)											
		t-1	*t-2*	*t-3*	*t-4*	*t-5*	*t-6*	*t-7*	*t-8*	*t-9*	*t-10*	*t-11*	*t-12*
Book store	PBS	30	21	8	6	16	10	0	9	18	17	23	20
	SP	30	5	30	0	30	12	0	9	3	0	0	29
Clothing store	PBS	30	30	30	30	30	30	30	30	30	30	30	30
	SP	30	0	0	0	0	0	14	17	30	30	0	30
Furniture store	PBS	30	0	30	30	24	30	28	2	30	30	30	30
	SP	30	0	30	30	25	25	11	0	14	30	30	30
Hardware store	PBS	27	29	0	29	27	29	23	27	27	29	16	27
	SP	30	30	0	0	30	30	30	30	25	29	30	30
Durable Consumer Goods	PBS	30	18	14	22	10	3	15	18	30	18	16	30
	SP	28	28	28	30	3	18	22	15	12	22	3	25

Uma estratégia de voto majoritário foi empregada nos resultados da Tabela 4.3 para obter a configuração de atrasos finais para cada método e para cada série temporal. Voto majoritário quer dizer que, de 30 execuções, se em 16 ou mais execuções o atraso fosse selecionado, então ele irá compor o sub-conjunto final de variáveis selecionadas.

Conseqüentemente, a configuração final dos atrasos é apresentado na Tabela 4.4. Como exemplo, a configuração final para a série *Book store* com o método SP passa então a ser descrita pela equação (4.7):

$$\hat{s}_t = f(s_{t-1}, s_{t-3}, s_{t-5}, s_{t-12}) . \tag{4.7}$$

Empregando o método PBS para a mesma série, a configuração final está descrita na equação (4.8):

$$\hat{s}_t = f(s_{t-1}, s_{t-2}, s_{t-5}, s_{t-9}, s_{t-10}, s_{t-11}, s_{t-12}) . \tag{4.8}$$

É interessante observar que, via o método SP, dos 12 atrasos inicialmente considerados, apenas 4 atrasos foram suficientes para produzir a melhora mais acentuada de desempenho do modelo de predição. E o método PBS identificou 7 atrasos para chegar à melhora mais acentuada de desempenho.

Tabela 4.4 - Configuração final dos atrasos selecionados ao aplicar voto majoritário sobre os resultados da Tabela 4.3.

SÉRIE TEMPORAL	MÉTODO	VARIÁVEIS
Book store	PBS	1 2 5 9 10 11 12
	SP	1 3 5 12
Clothing store	PBS	1 2 3 4 5 6 7 8 9 10 11 12
	SP	1 8 9 10 12
Furniture store	PBS	1 3 4 5 6 7 9 10 11 12
	SP	1 3 4 5 6 10 11 12
Hardware store	PBS	1 2 4 5 6 7 8 9 10 11 12
	SP	1 2 5 6 7 8 9 10 11 12
Durable Consumer Goods	PBS	1 2 4 8 9 10 11 12
	SP	1 2 3 4 6 7 10 12

A decisão de realizar múltiplas execuções para cada método está relacionada ao propósito de apresentar resultados comparativos de dois métodos que interagem com um modelo de aproximação de funções, no caso uma MLP. O treinamento consiste num processo de otimização em que a função a minimizar é multi-modal, depende da arquitetura da rede e dos valores das amostras. Este é um caso típico deste tipo de treinamento, onde o mínimo global não é garantido e há a possibilidade de convergir para um mínimo local associado à condição inicial da rede MLP.

Testando as variáveis:

Neste ponto, após aplicar os métodos de seleção de variáveis, os atrasos mais relevantes das séries temporais foram identificados, os quais oferecem melhor capacidade de predição. As simulações seguintes têm como objetivo comparar o ganho na predição (acima do subconjunto de teste) utilizando estes atrasos quando confrontados com o caso de não ter aplicado esses métodos de seleção, o que seria utilizar os L atrasos completos iniciais. Comparar as formas de separar os conjuntos de treinamento e validação (seqüencial vs. randômico) também faz parte das simulações (veja Figura 4.10).

Os parâmetros que guiaram as simulações foram os seguintes:

Índice de desempenho:	*MAE* (acima do conjunto teste)
Neurônios ocultos:	30
Máximo número de épocas no treinamento da MLP:	300
Algoritmo de treinamento da MLP:	Método do Gradiente
Validação cruzada (treinamento, validação e teste):	Seqüencial e randômico (em porcentagens de 50, 25, 25, respectivamente), ver Figura 4.10.
Número de execuções de cada método:	30

Vale indicar que o número de 300 épocas para o treinamento das redes MLP foi definido a partir de testes prévios e levando-se em conta que as séries temporais, na sua maioria, têm poucos pontos. Além disso, empregou-se parada antecipada no treinamento, o que permite parar o treinamento (sem precisar atingir as 300 épocas) quando o erro de validação experimentar deterioração constante. Os resultados numéricos estão apresentados na Tabela 4.5.

Tabela 4.5 - Resultados para 30 execuções comparando “sem” (não uso de seleção de variáveis) vs “PBS” vs “SP”, cada um deles com separação de dados: Seqüencial e Randômico.

			TREINAMENTO		VALIDAÇÃO		TESTE	
SÉRIES	SEPARAÇÃO DOS DADOS	MÉTODO	MAE	RMSE	MAE	RMSE	MAE	RMSE
Book store	Seqüencial	sem	33,481 ± 1,625	6,127 ± 0,216	54,091 ± 1,862	14,802 ± 0,405	74,697 ± 3,193	17,255 ± 0,668
		PBS	33,380 ± 0,268	6,484 ± 0,036	46,567 ± 0,479	13,053 ± 0,203	75,926 ± 1,749	17,663 ± 0,392
		SP	33,101 ± 0,152	6,415 ± 0,018	43,933 ± 0,862	12,646 ± 0,270	72,062 ± 1,298	16,760 ± 0,324
	Randômica	sem	35,600 ± 3,783	6,516 ± 0,727	49,208 ± 6,450	12,783 ± 1,887	72,469 ± 4,073	17,001 ± 0,846
		PBS	37,353 ± 2,857	6,964 ± 0,581	42,934 ± 6,246	11,372 ± 1,688	73,109 ± 3,766	16,976 ± 0,790
		SP	36,728 ± 3,900	6,829 ± 0,697	41,589 ± 5,957	11,170 ± 1,907	67,116 ± 3,188	15,902 ± 0,515
Clothing store	Seqüencial	sem	563,775 ± 3,948	115,910 ± 0,399	994,764 ± 36,416	277,443 ± 5,337	1149,785 ± 92,005	251,576 ± 16,781
		PBS	563,448 ± 3,112	115,975 ± 0,386	988,506 ± 34,189	276,370 ± 4,787	1137,771 ± 92,604	249,194 ± 17,385
		SP	551,540 ± 2,117	117,785 ± 0,097	837,524 ± 8,700	262,977 ± 1,300	732,521 ± 14,006	181,670 ± 3,493
	Randômica	sem	541,227 ± 112,909	126,936 ± 28,630	733,131 ± 138,395	222,186 ± 64,521	597,329 ± 100,302	145,962 ± 29,452
		PBS	538,121 ± 103,998	124,989 ± 29,488	711,641 ± 125,686	221,706 ± 68,598	580,018 ± 92,655	141,382 ± 26,947
		SP	559,053 ± 96,201	136,015 ± 25,377	595,923 ± 145,884	194,963 ± 69,571	522,458 ± 81,389	135,130 ± 23,142
Furniture store	Seqüencial	sem	63,955 ± 1,621	11,098 ± 0,195	82,875 ± 1,453	19,537 ± 0,566	176,861 ± 9,697	41,226 ± 3,042
		PBS	69,128 ± 1,313	11,626 ± 0,103	77,836 ± 2,257	19,014 ± 0,405	176,703 ± 6,055	40,450 ± 1,608
		SP	67,928 ± 0,974	11,618 ± 0,092	77,320 ± 1,102	18,764 ± 0,240	176,478 ± 5,360	40,307 ± 1,426
	Randômica	sem	66,493 ± 4,549	11,353 ± 0,730	81,017 ± 9,327	19,562 ± 2,065	168,674 ± 10,879	38,667 ± 2,345
		PBS	67,659 ± 4,377	11,570 ± 0,743	85,424 ± 11,358	20,575 ± 2,317	171,380 ± 9,895	39,124 ± 2,632
		SP	65,926 ± 4,042	11,359 ± 0,676	85,862 ± 8,158	20,611 ± 1,749	168,426 ± 9,180	37,771 ± 1,836
Hardware store	Seqüencial	sem	31,527 ± 0,286	5,437 ± 0,048	61,244 ± 1,911	13,708 ± 0,384	50,385 ± 2,126	11,526 ± 0,394
		PBS	32,444 ± 0,270	5,514 ± 0,041	58,786 ± 2,020	13,291 ± 0,375	45,478 ± 2,195	10,959 ± 0,375
		SP	32,884 ± 0,388	5,720 ± 0,062	50,151 ± 2,148	11,864 ± 0,445	37,800 ± 2,530	9,518 ± 0,493
	Randômica	sem	36,010 ± 5,702	6,075 ± 0,924	47,076 ± 6,643	11,168 ± 1,578	39,088 ± 7,116	8,815 ± 1,493
		PBS	34,772 ± 2,668	5,912 ± 0,484	43,202 ± 5,355	10,444 ± 1,329	37,489 ± 3,573	8,535 ± 0,775
		SP	34,776 ± 3,037	6,010 ± 0,495	43,846 ± 5,869	10,401 ± 1,331	37,797 ± 3,653	8,556 ± 0,735
Durable Consumer Goods	Seqüencial	sem	2,160 ± 0,096	0,150 ± 0,006	2,584 ± 0,173	0,263 ± 0,016	4,004 ± 0,296	0,451 ± 0,037
		PBS	2,033 ± 0,085	0,142 ± 0,006	2,415 ± 0,145	0,246 ± 0,014	3,769 ± 0,233	0,418 ± 0,029
		SP	2,094 ± 0,115	0,144 ± 0,008	2,504 ± 0,174	0,253 ± 0,016	3,843 ± 0,356	0,425 ± 0,041
	Randômica	sem	2,073 ± 0,103	0,146 ± 0,007	2,112 ± 0,166	0,210 ± 0,014	3,685 ± 0,177	0,398 ± 0,025
		PBS	2,057 ± 0,095	0,145 ± 0,006	2,081 ± 0,149	0,208 ± 0,016	3,591 ± 0,201	0,397 ± 0,027
		SP	2,051 ± 0,098	0,144 ± 0,006	2,093 ± 0,179	0,207 ± 0,017	3,639 ± 0,206	0,388 ± 0,022

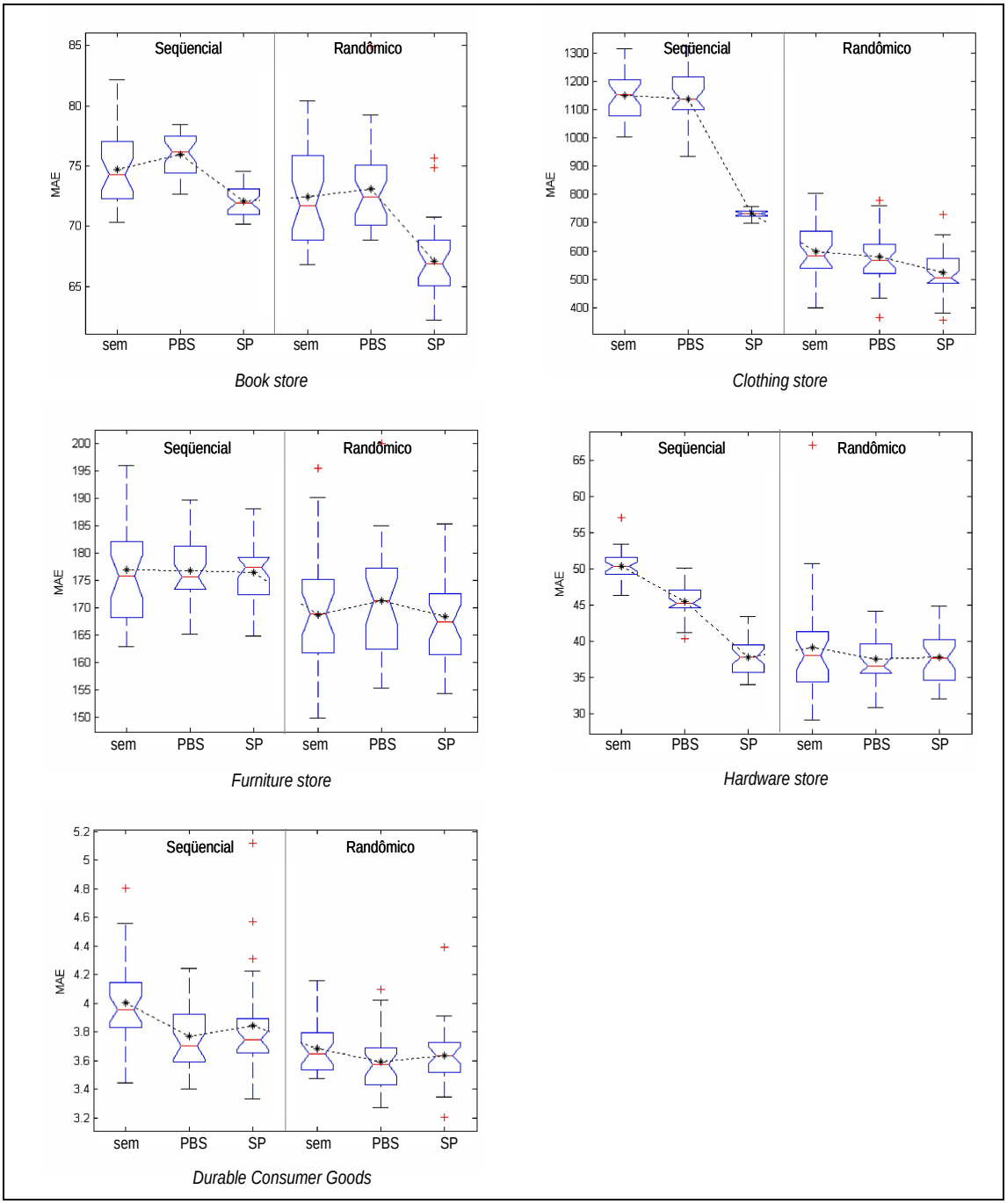


Figura 4.11 - Visualização gráfica dos resultados da Tabela 4.5 referente ao conjunto de teste.

Na Figura 4.11, foram utilizados *boxplots* para mostrar as estatísticas sobre as 30 repetições. Os símbolos "*" unidos por linhas tracejadas representam a média.

A Figura 4.12 apresenta um exemplo comparativo para a série temporal *Clothing store*, dados de teste, onde para a parte esquerda da figura foram utilizados todos os atrasos disponíveis, L=12, com separação de dados seqüencial, e na direita foram usadas as variáveis obtidas via o método SP com separação de dados randômica. O *MAE* diminuiu de 1.213,967 para 532,549. Nos gráficos, as curvas sólidas representam os valores reais da série e as tracejadas a predição.

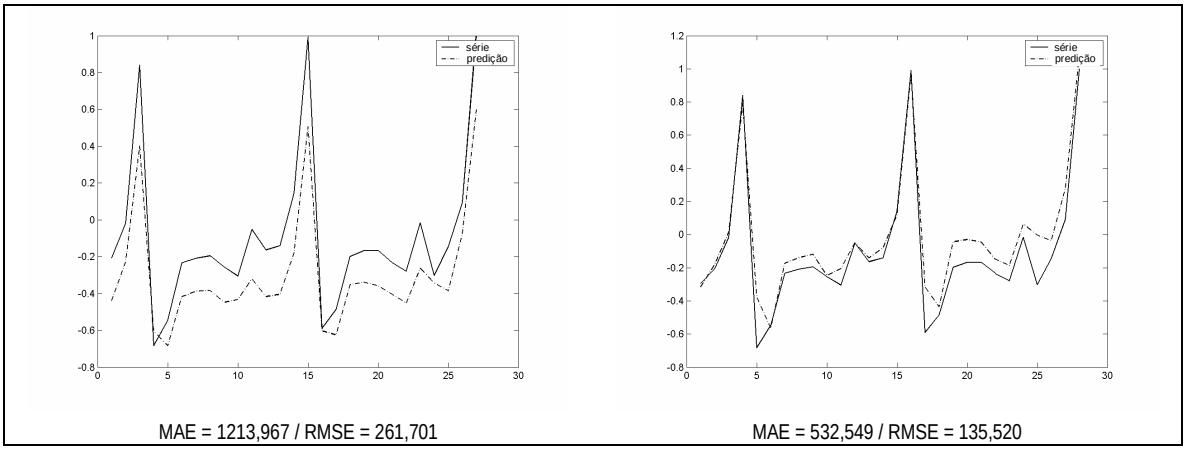


Figura 4.12 - Um exemplo gráfico na predição da série temporal *Clothing store*. Na esquerda, preditor obtido via separação seqüencial e usando todos os atrasos, L=12 iniciais. Na direita, preditor obtido via separação randômica e com os 5 atrasos obtidos via SP.

T- Teste:

A motivação para usar o T-teste está relacionada a ter um indicativo que permita inferir se as duas propostas são estatisticamente diferentes a partir dos resultados obtidos e do número de experimentos realizados.

As Tabelas 4.6–4.9 apresentam os resultados do T-teste, onde os valores P e t foram obtidos a partir dos resultados da Tabela 4.5, especificamente da coluna dos valores *MAE* do conjunto de teste (penúltima coluna) e com um grau de confiança de 95%.

A Tabela 4.6 apresenta o T-teste para “PBS vs SP”, tanto para as separações Seqüencial como Randômica. Pode-se concluir que, em 5 das 10 predições, os resultados são

estatisticamente diferentes (ver células com fundo cinza), e SP teve melhor desempenho que PBS. Nas outras 5 predições, ambos os métodos tiveram desempenhos equivalentes.

Tabela 4.6 - T-teste para análise de significância da diferença de desempenho entre os métodos PBS e SP.

SÉRIE	P/t (PBS-SP) - Seqüencial		P/t (PBS-SP) - Randômica	
	P	t	P	t
Book store	**0**	9,732	**0**	6,653
Clothing store	**0**	23,7	**0,0133**	2,556
Furniture store	0,8796	0,152	0,2342	1,199
Hardware store	**0**	12,555	0,7424	-0,33
Durable Consumer Goods	0,3453	-0,953	0,0597	1,922

O T-teste para "Sem seleção vs SP" é apresentado na Tabela 4.7. De forma similar ao anterior, separações seqüenciais e randômicas também foram consideradas. Em 5 casos, não usando critério de seleção e usando SP produziram resultados estatisticamente diferentes, sendo SP superior. Nos outros casos, foram estatisticamente equivalentes.

Tabela 4.7 - T-teste para análise de significância da diferença de desempenho entre ausência de seleção e SP.

SÉRIE	P/t (sem-SP) - Seqüencial		P/t (sem-SP) - Randômica	
	P	t	P	t
Book store	**0,0002**	4,187	**0**	5,669
Clothing store	**0**	24,558	**0,0024**	3,175
Furniture store	0,8494	0,189	0,9246	0,095
Hardware store	**0**	20,859	0,3777	0,884
Durable Consumer Goods	0,0623	1,905	0,3606	0,928

A Tabela 4.8 considera "Sem seleção vs PBS", e em apenas 2 casos o PBS foi estatisticamente superior. Nos outros casos, foram equivalentes.

Tabela 4.8 - T-teste para análise de significância da diferença de desempenho entre ausência de seleção e PBS.

SÉRIE	P/t (sem-PBS) - Seqüencial		P/t (sem-PBS) - Randômica	
	P	t	P	t
Book store	0,0709	-1,849	0,5287	-0,632
Clothing store	0,6161	0,504	0,4889	0,694
Furniture store	0,9404	0,076	0,3156	-1,008
Hardware store	**0**	8,795	0,2818	1,100
Durable Consumer Goods	**0,0012**	3,417	0,0597	1,922

Finalmente, a Tabela 4.9 apresenta o T-teste para as separações "Seqüencial vs Randômica", e foram consideradas para os três casos, sem seleção, seleção via SP e seleção via PBS. Nas simulações, somente em um dos 15 casos analisados é que a separação seqüencial foi estatisticamente equivalente à randômica. Nos 14 restantes, a separação randômica mostrou-se superior.

Tabela 4.9 - T-teste para análise de significância da diferença de desempenho entre separação seqüencial e randômica.

SÉRIE	P/t (Seqüencial-Randômica) - Sem seleção		P/t (Seqüencial-Randômica) - Com seleção via SP		P/t (Seqüencial-Randômica) - Com seleção via PBS	
	P	t	P	t	P	t
Book store	**0,0218**	2,358	**0**	7,870	**0,0006**	3,716
Clothing store	**0**	22,232	**0**	13,932	**0**	23,321
Furniture store	**0,003**	3,077	**0,0002**	4,149	**0,0154**	2,513
Hardware store	**0**	8,331	0,9971	0,004	**0**	10,435
Durable Consumer Goods	**0**	5,066	**0,0017**	3,356	**0,0026**	3,168

4.5. Considerações finais

Com base na análise estatística realizada, pode-se apontar que:

- baseado nas 5 séries temporais analisadas, o método de seleção de variáveis SP mostrou-se consideravelmente superior ao método PBS, além de apresentar menor número de variáveis selecionadas;

- a principal desvantagem do método SP é o custo computacional. Empregando a abordagem Envoltório e tendo uma MLP como modelo regressor, o fato do número de variáveis candidatas ser L implica que para SP foram necessários $L(L+1)/2$ processos de treinamento completos de redes MLP, e para PBS apenas L.

- Dado que SP e PBS obedecem a uma estratégia de busca baseada em heurística, no caso construtiva e de poda, respectivamente, não há garantias de se alcançar o subconjunto ótimo de variáveis.

- Quanto à avaliação da melhor forma de separar os dados de treinamento e validação, foi mostrado estatisticamente que a separação randômica é a melhor opção. Uma possível justificativa é o ganho que se tem com a inclusão de amostras mais recentes, a qual só é possível na separação randômica. A diminuição do erro sobre o conjunto de teste reflete esse ganho de desempenho. Para montar o conjunto de teste em todas as simulações, sempre foram consideradas as amostras mais recentes da série e numa porcentagem de 25% , como indicado na Figura 4.10. A escolha do conjunto de teste independe da separação ser seqüencial ou randômica.

- Foram verificados que, via métodos de seleção de variáveis, não só é possível reduzir o número de parâmetros do modelo, como também é possível produzir um incremento de desempenho em termos de qualidade da predição.

Capítulo 5

Conclusões e perspectivas futuras

5.1 Conclusões

As conclusões são apresentadas de forma separada, segundo o tema tratado. Num primeiro momento, as relacionadas a comitês de máquinas e, posteriormente, seleção de variáveis, ambos no contexto de predição de séries temporais.

5.1.1 Conclusões relacionadas a comitês de máquinas

Os resultados das simulações realizadas na predição das 5 séries temporais com as abordagens "um único modelo preditor (MLP)" e "comitês de máquinas (*ensemble* e mistura de especialistas)" foram apresentadas no Capítulo 3.

Nos resultados, observa-se a ocorrência de ganho significativo na predição ao combinar modelos preditores, seja em forma de *ensemble* ou de mistura de especialistas, quando comparado com o ganho obtido via um único modelo preditor.

É importante aclarar que não seria apropriado concluir que mistura de especialistas é uma abordagem superior a *ensemble*, pelas seguintes razões:

- Tanto *ensemble* quanto mistura de especialistas são abordagens gerais, que dependem da qualidade dos componentes ou especialistas e as formas como eles interagem. Por exemplo, para o caso de *ensemble* existem outras formas de impor diversidade na etapa de *geração de componentes*, como *Boosting*, *AdaBoost*, computação evolutiva, que ainda podem produzir ganho de desempenho;

- Da mesma forma para a etapa de *combinação*: além da média simples, poderia optar-se por estratégias de média ponderada, por exemplo;
- E para a etapa de *seleção*, também há perspectivas de melhora, com o emprego de técnicas de construção ou poda mais sofisticadas;
- O mesmo se aplica no caso de mistura de especialistas, que também admite variações, como usar uma função *kernel* gaussiana normalizada em lugar da função *softmax*;
- Com base na metodologia de projeto adotada, as misturas de especialistas apresentaram melhores resultados, mas restrito apenas à predição de 5 séries temporais;

Em ambos os casos, a variância foi menor do que quando se considera uma única *MLP*. Isso corrobora resultados já apresentados na literatura (HAYKIN, 1999; BICHOP, 1995)

5.1.2 Conclusões relacionadas à Seleção de Variáveis

Dois métodos de seleção de variáveis dentro da abordagem de Envoltório foram implementados: Seleção Progressiva (SP) e Poda Baseada em Sensibilidade (PBS). Os desempenhos foram contrastados com a ausência de seleção de variáveis, permitindo concluir:

- Houve ganho ao usar técnicas de seleção de variáveis e que foram estatisticamente significantes na predição da maioria das séries consideradas nesta tese;
- SP trouxe melhores ganhos quando comparado a PBS;
- O número de variáveis selecionadas via SP foi menor em todos os casos quando comparado ao número de variáveis selecionadas via PBS;
- No entanto, o custo computacional de SP é superior ao de PBS.

5.2 Perspectivas futuras

Antes de discorrer sobre os possíveis próximos passos da pesquisa, é preciso ressaltar alguns aspectos importantes que marcaram a história das técnicas de predição de séries temporais, particularmente no contexto de aprendizado de máquinas. Para tanto, as Figuras 5.1 e 5.2 servirão de referência.

Na Figura 5.1, mostra-se na parte A a forma em que tradicionalmente (nos anos 1970 e começo dos anos 1980) as séries temporais foram abordadas. A idéia, nesses anos, era tentar pré-processar a série temporal (S.T.) visando ampliar a aplicabilidade dos modelos propostos. Sendo assim, na maioria dos casos a série temporal sofria transformações para fugir da não-estacionariedade, por exemplo. Modelos *ARMA* e *ARIMA* (BOX, *et al*., 1994) representam exemplos clássicos desse período. Havia uma tendência entre os estudiosos da época em buscar o "melhor ajuste", daí a preferência por ajustar modelos lineares. Essa concepção criou obstáculos para que a pesquisa junto a modelos não-lineares, onde o "melhor ajuste" não tinha solução global, sofresse progressos significativos.

Posteriormente, viu-se que os resultados de predição podiam ser melhorados usando modelos não-lineares (parte B da Figura 5.1), e assim foram utilizados modelos como os bi-lineares, TAR (lineal por partes) e redes neurais artificiais, dentre outros. Sob esta nova perspectiva, o caminho era o oposto: já não se buscava "encaixar" a série temporal num modelo com fortes restrições de flexibilidade, mas sim usar a flexibilidade dos modelos não-lineares para sintetizar o preditor, sem necessariamente passar pela etapa de pré-processamento da série temporal.

Quando escolhemos um modelo não-linear, no caso uma rede neural *MLP*, considerada como aproximador universal, o desafio está na calibração do grau de flexibilidade a ser explorado:

- ✓ se a arquitetura escolhida for reduzida, por exemplo, baixo número de neurônios na camada oculta, há uma redução na flexibilidade do modelo e problemas altamente complexos podem não ser tratados a contento. Com isso, comete-se um *erro de aproximação*.

- ✓ por outro lado, se a arquitetura escolhida for adequada, apresentando um grau de flexibilidade dentro das exigências do problema, ainda falta o ajuste dos parâmetros, o qual pode não ser uma tarefa trivial, dado que a função-objetivo que guia esta etapa é multimodal. A ocorrência de mínimos locais no processo de otimização produz então o *erro de estimação*.

O erro de generalização, ou em nosso caso, *erro de predição*, é altamente dependente de ambos os erros citados anteriormente.

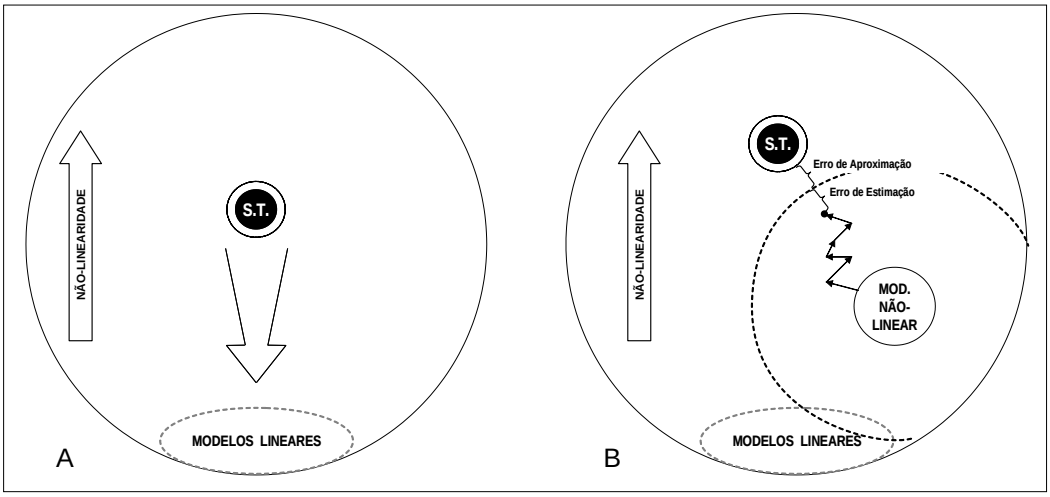


Figura 5.1 - A: visão tradicional e B: visão alternativa para predição de séries temporais.

Spyros Makridakis é um estatístico conhecido por fugir da idéia do "melhor ajuste" e criar, junto a outros colaboradores, a área de predição (*forecasting*). Ele mostrou que o "melhor ajuste" em predição nem sempre é o mais indicado, particularmente por se supor que o passado repete-se da mesma forma no futuro. Ele é defensor da idéia de que o futuro nunca é igual ao passado (MAKRIDAKIS *et al.*, 1982). Mas isto tem muita relação com modelos de RNAs, em que o treinamento deverá ser detido antes de cair no que se conhece como sobre-ajuste ou sobre-treinamento (*overfitting*) (BISHOP, 1995; HAYKIN 1999). Este pode ser visto como um ponto de consenso em que modelos de RNAs são favorecidos.

Como se apresentou no Capítulo 2, existem na literatura de RNAs até 3 formas empíricas de evitar o sobre-ajuste num único modelo. Aqui foi empregada a *validação cruzada*. Mas,

em séries temporais isto nem sempre garante atingir uma boa predição, e não é porque tratam-se de técnicas empíricas, nem por causa de mínimos locais. O fato é que, em predição, o ponto em que se deve ajustar um modelo pode não ser o mesmo quando o objetivo da predição é de longo, médio ou curto prazo. Um exemplo claro ocorre quando a série tem componentes complexos, como "grandes ciclos" (do inglês *huge cycles*) (MAKRIDAKIS, 1995).

Ante este panorama, surgem as propostas de comitês de máquinas. Referindo-se à Figura 5.2, ela ilustra como a combinação de vários modelos, alguns deles possivelmente até de baixo desempenho, pode conduzir a resultados que superam o desempenho individual de qualquer dos componentes, reduzindo também a variância. Aqui é possível citar: (*i*) NEWBOLD & GRANGER (1974) como os primeiros a aplicar estratégias de combinação de modelos de predição, MCLEOD (1993) e FILDES & MAKRIDAKIS (1995) como as principais contribuições na área de predição; (*ii*) toda a bibliografia citada ao longo do texto e relacionada a comitê de máquinas, representando a comunidade de RNAs e aprendizado de máquinas.

Ao combinar modelos de predição, busca-se obter um consenso a partir deles, que está relacionado com a média do erro. Com isto, espera-se estar o mais próximo de acompanhar o padrão fundamental (tendência) que guia a série temporal.

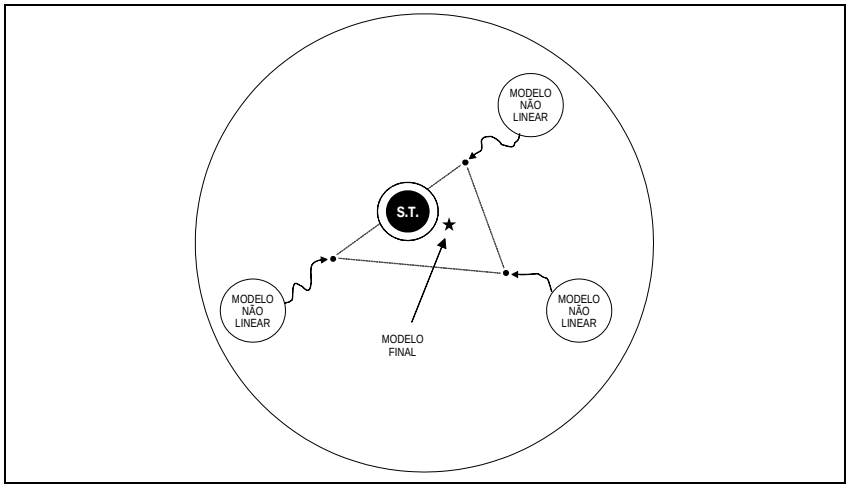


Figura 5.2 - Visão atual de comitê de máquinas em predição de séries temporais.

Seguindo esses delineamentos, as perspectiva futuras são:

- Encontrar técnicas de seleção de modelos (ponto de ajuste do modelo) mais favoráveis para o problema de predição de séries temporais, considerando o horizonte de predição (longo, médio e curto prazo).
- Estudo de modelos de regressão lineares/não-lineares baseados em máquinas de vetores-suporte (*Support Vector Machines - SVM*). Basicamente o estudo visa descobrir até que ponto essas técnicas que têm garantia de mínimo global no ajuste dos parâmetros podem ser úteis no problema de predição de séries temporais.
- Estudo de outros métodos de Seleção de Variáveis, principalmente na abordagem Filtro, utilizando técnicas que medem correlação não-linear entre variáveis, como por exemplo Informação Mútua Condicional (FLEURET, 2004; WANG & LOCHOVSKY, 2004) e aplicá-las ao problema de predição de séries temporais.
- Estudo de problemas de séries temporais multi-variáveis, com predição a longo, médio e curto prazo.
- Estudo de técnicas para a definição automática do número de especialistas em MEs (WATERHOUSE & ROBINSON, 1995).

Anexo A

Auto-correlação e informação mútua para cálculo da janela de predição

De uma forma simplificada, pode-se representar um valor de uma série temporal em função de L valores passados consecutivos e de um componente aleatório e_t, comumente representado como um ruído branco $N(0,1)$, conforme indicado na equação (A.1). Sendo assim, o parâmetro L indica o número de atrasos (*lags*) que é preciso considerar para obter um valor futuro da série temporal:

$$s_t = f(s_{t-1}, s_{t-2}, ..., s_{t-L}) + e_t \,. \tag{A.1}$$

Particularmente no caso em que a série temporal apresenta sazonalidade, o valor da variável L é tomado como o produto de outros dois parâmetros , n e T, na forma:

$$L = nT \,. \tag{A.2}$$

Assim, reescrevendo a equação (A.1), resulta:

$$s_t = f(s_{t-1}, s_{t-2}, ..., s_{t-nT}) + e_t \,. \tag{A.3}$$

Por exemplo, para o caso n=2 e T=6, a representação da série temporal seria dada na seguinte forma:

$$s_t = f(s_{t-1}, s_{t-2}, ..., s_{t-5}, s_{t-T}, s_{t-7}, ..., s_{t-11}, s_{t-2T}) + e_t \,. \tag{A.4}$$

O significado da variável T pode ser associado ao número de atrasos que se encontram embutidos na série temporal (FRASER & SWINNEY, 1986), onde o termo embutido pode ser interpretado como “sujeito a correlação ou associação”.

Definindo agora a função genérica $fcor(V_1, V_2)$ como a função que mede a correlação existente entre as variáveis V_1 e V_2, tem-se que esta função excursiona no intervalo [–1,+1], sendo nula quando não há correlação entre as variáveis (diz-se então que as

variáveis são ortogonais) e batendo nos limites do intervalo quando o conhecimento de uma delas é suficiente para se estimar completamente a outra.

Cálculo de *T*:

Para o cálculo do valor de *T* em uma série temporal que apresenta uma dependência do tipo da equação (A.3), é necessário estimar os valores de:

$$fcor(S_t, S_{t-1}),\quad fcor(S_t, S_{t-2}),\quad fcor(S_t, S_{t-3}), \ldots,\quad fcor(S_t, S_{t-Tmax}), \tag{A.5}$$

onde *Tmax* é um valor definido a priori e espera-se que valha $Tmax \geq T$. A definição do valor de *T* se dá ao plotar os valores obtidos e vai depender da função *fcor* utilizada:

- Se *fcor* for uma função que mede correlação linear, por exemplo, funções com base na covariância entre as variáveis, o valor de *T* depende da adoção de um limiar abaixo do qual as variáveis são descartadas, havendo portanto uma faixa de valores de *fcor* que indica baixa correlação. O primeiro valor a entrar nessa faixa representará o valor de *T* (BOX, *et al.*, 1994). Em caso da série apresentar sazonalidade, é comum que os valores entrem e saiam da faixa com a mesma periodicidade da sazonalidade.

- Se *fcor* for uma função que mede correlação não-linear entre variáveis, como por exemplo, a função de informação mútua conjunta para duas variáveis, o valor de *T* corresponde ao primeiro mínimo de *fcor* (FRASER & SWINNEY, 1986), onde o *k*-ésimo valor da função é dado por $fcor(S_t, S_{t-k})$.

Cálculo de *n*:

O cálculo de *n* na equação (A.2) é um tema junto ao qual não há um estudo teórico para sua definição. Em princípio, o cálculo de *T* serviu para identificar o número de atrasos que seriam úteis para a predição de valores futuros da série, por medidas de correlação linear ou não-linear.

Correlação via Informação Mútua Conjunta

Informação Mútua é uma forma de medir a informação que uma variável detém de outra variável, foi inicialmente utilizada na área de comunicações e está embasada na Teoria de Informação (COVER & THOMAS, 1991; SHANNON & WEAVER, 1949). Posteriormente, essas idéias foram utilizadas para medir correlações não-lineares entre duas variáveis (CELLUCCI, 2005; FRASER & SWINNEY, 1986).

O cálculo da IM conjunta $I(\mathrm{X},\mathrm{Y})$ está baseado na seguinte equação:

$$I(\mathrm{X},\mathrm{Y}) = \sum_{i=1}^{N}\sum_{j=1}^{N} P_{\mathrm{XY}}(x_i, y_j)\log_2\left\{\frac{P_{\mathrm{XY}}(x_i, y_j)}{P_{\mathrm{X}}(x_i)P_{\mathrm{Y}}(y_j)}\right\} \qquad \text{(A.6)}$$

onde X e Y são variáveis que podem estar em qualquer escala. $P_{XY}(x_i,y_j)$ é a probabilidade conjunta de $X=x_i$ e $Y=y_j$, $P_X(x_i)$ e $P_Y(y_j)$ são as probabilidades marginais de X e Y, respectivamente, e N é o número de amostras consideradas no cálculo.

Repare que, para o cálculo de $I(\mathrm{X},\mathrm{Y})$, é preciso aproximar uma função densidade de probabilidade envolvendo duas variáveis. Nesta dissertação, foi empregado o cálculo de $I(\mathrm{X},\mathrm{Y})$ baseado no trabalho de CELLUCCI (2005), que utiliza um histograma uniforme para o cálculo das probabilidades conjunta e marginais.

Desvantagem da Informação Mútua

No mesmo trabalho de CELLUCCI (2005), mostra-se que a informação mútua é sensível ao número de amostras da série, com o valor de T para o primeiro mínimo variando mais quando o número de amostras é pequeno. A estimativa fica mais confiável para um número de amostras mais elevado.

Cálculo de *L* para as séries temporais desta dissertação

As Figuras A.1 a A.5 a seguir mostram os resultados de aplicação das funções de auto-correlação e de informação mútua conjunta para cada uma das séries temporais consideradas nos experimentos realizados, utilizando $Tmax$=50.

Série *Bookstore*:

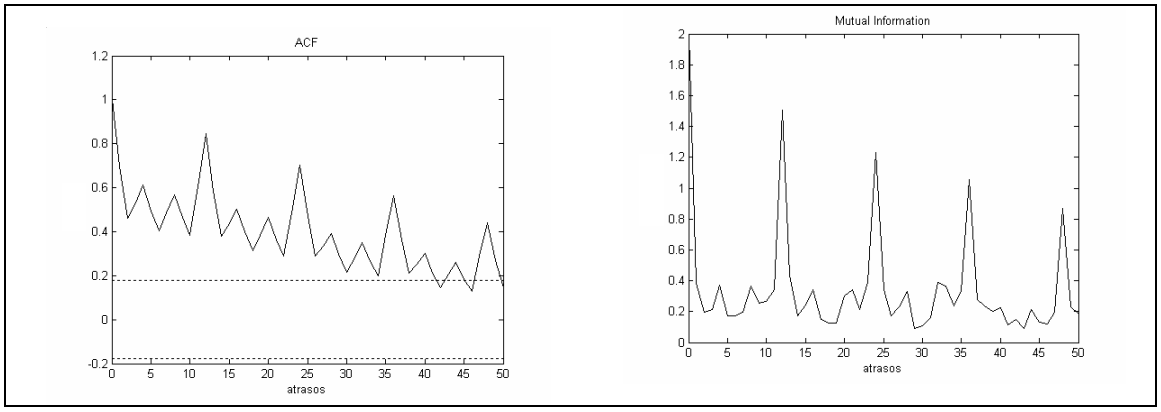


Figura A.1: Esquerda: função de auto-correlação. Direita: Informação Mútua. Série *Bookstore*

Série *Clothing store*:

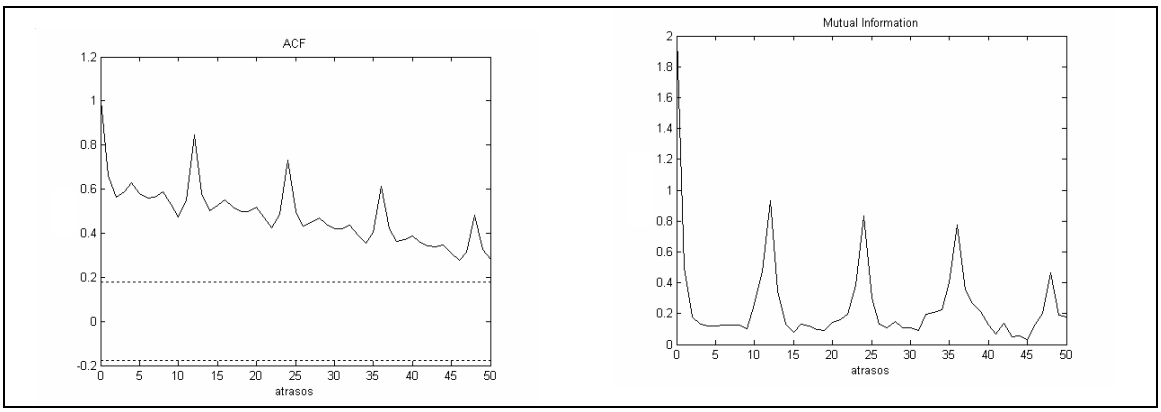


Figura A.2 Esquerda: função de auto-correlação. Direita: Informação Mútua. Série *Clothing store*

Série *Furniture store*:

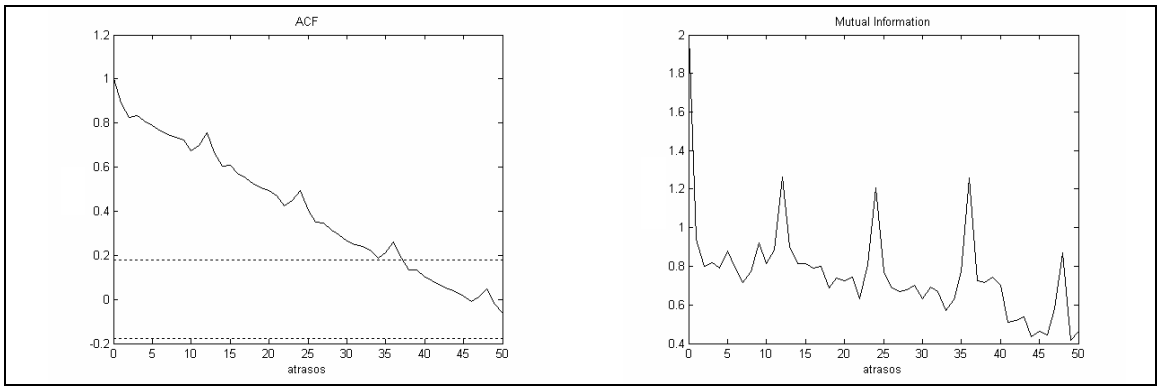


Figura A.3 Esquerda: função de auto-correlação. Direita: Informação Mútua. Série *Furniture store*

Série *Hardware store*:

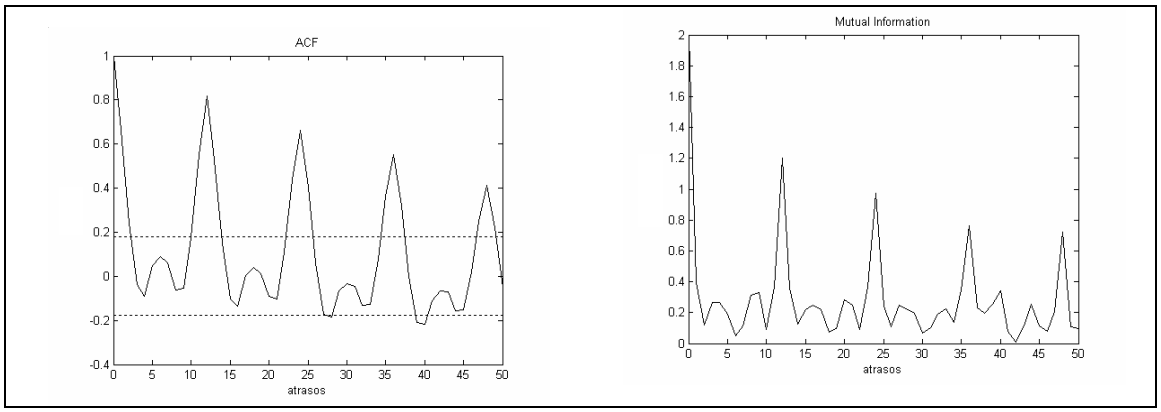


Figura A.4 Esquerda: função de auto-correlação. Direita: Informação Mútua. Série *Hardware store*

Série *Durable Consumer Goods*:

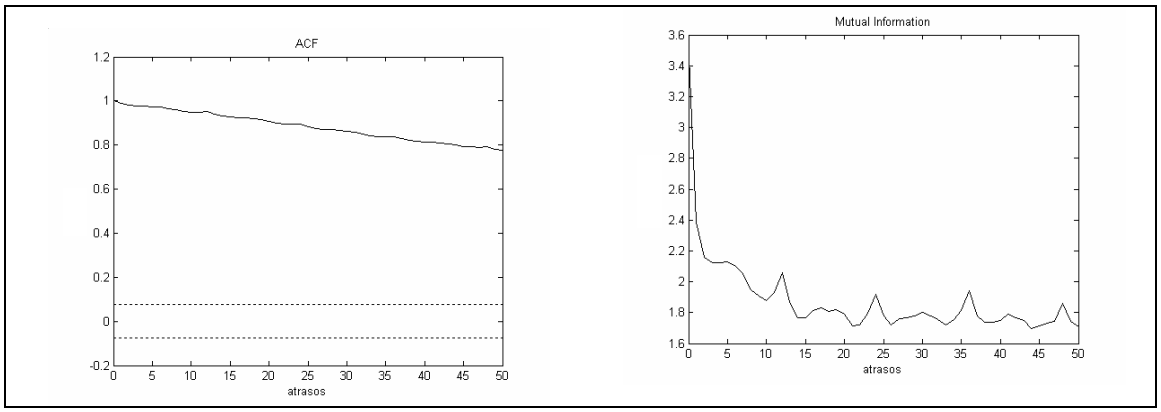


Figura A.5 Esquerda: função de auto-correlação. Direita: Informação Mútua. Série *Durable Consumer Goods*

Analisando as cinco série consideradas nas Figuras A.1 a A.5, identificam-se os seguintes valores para as variáveis T e n:

SÉRIE TEMPORAL	T	n	$L=nT$
Book store	2	6	12
Clothing store	4	3	12
Furniture store	2	6	12
Hardware store	2	6	12
Durable Consumer Goods	3	4	12

Os valores de *n* foram escolhidos de forma a alcançar os 12 atrasos. Isto porque todas as séries apresentam uma forte sazonalidade anual. O cálculo da variável *T* foi realizado a partir dos gráficos da correlação via informação mútua, escolhendo o valor de *T* (eixo horizontal, que representa os atrasos da série) que corresponde ao primeiro mínimo que experimenta a curva de informação mútua.

A função de auto-correlação é comumente utilizada quando a série temporal é estacionária e mede correlações lineares. Modelos como ARIMA utilizam esta função para a definição dos atrasos. Para o caso de nossas séries temporais, visto que os preditores serão RNAs, obtou-se por empregar os resultados obtidos via informação mútua.

Uma outra característica das séries temporais, além das descritas no Capítulo 2, é de *memória de longo prazo*. Nas séries que possuem esta característica, a correlação envolvendo os valores passados cai lentamente. Na literatura, pode-se encontrar os modelos ARFIMA (*autoregressive fractionally integrated moving average*) propostos por GRANGER & JOYEUX (1980) e HOSKING (1981) para modelar essas séries.

Na Figura A.6, apresenta-se um teste envolvendo uma série temporal artificial com memória de longo-prazo, gerada no Matlab versão 7.0:

```
N=25000;
serie = wfbm(0.7,N);
```

Foram gerados 25000 pontos, dos quais apenas os últimos 1000 foram considerados (fbm vem do inglês *fractional brownian motion*). O movimento fracionário browniano foi proposto por MANDELBROT & VAN NESS (1968) como um meio de modelar processos estocásticos não-estacionários que exibem dependência de longo prazo. O movimento depende de um único parâmetro *h*, onde: $0<h<1$. No caso considerado, $h=0,7$. O valor de *Tmax* foi ampliado para 200.

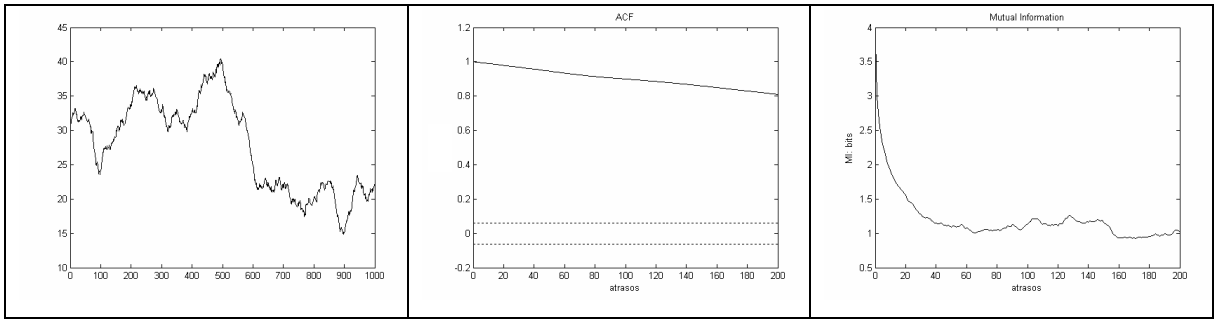


Figura A.6 Esquerda: série temporal *fbm* com 1000 pontos. Centro: função de auto-correlação. Direita: Informação Mútua.

O objetivo deste teste é mostrar que, via informação mútua, uma série temporal com memória de longo prazo é claramente representada, onde a curva da informação mútua não apresenta um mínimo definido, o que diferencia das séries temporais consideradas nesta tese, justamente por estas não apresentarem este tipo de memória. Já o resultado via função de auto-correlação para a série *Durable Consumer Goods* daria indício de presença de memória de longo prazo na série (se comparássemos este gráfico com o obtido para a série *fbm* da Figura A.6). No entanto, para o mesmo caso, a informação mútua indica o contrário (veja Figura A.5).

Anexo B

Publicações vinculadas a esta dissertação

Dentre as publicações realizadas no período do mestrado, as diretamente associadas a esta dissertação são:

Puma-Villanueva, W. J.; dos Santos, E. P.; Von Zuben, F. J. Data partition and variable selection for time series prediction using wrappers. *Proceedings of the IEEE International Joint Conference on Neural Networks - IJCNN*, vol. 1, pp. 9490-9497, Vancouver, 2006.

Puma-Villanueva W. J.; Lima C. A. M.; dos Santos E. P.; Von Zuben F. J. Mixture of Heterogeneous Experts Applied to Time Series: A Comparative Study. *Proceedings of the IEEE International Joint Conference on Neural Networks - IJCNN*, vol. 1, pp. 1160-1165, Montreal, 2005.

Puma-Villanueva W. J.; Bezerra G. B. P.; Lima C. A. M.; Von Zuben F. J. Improving Support Vector Clustering with Ensembles. *IJCNN 2005 Workshop on Achieving Functional Integration of Diverse Neural Models*, Montreal, 2005.

Lima C.A.M.; **Puma-Villanueva W. J.**; dos Santos E. P.; Von Zuben F. J. A Multistage Ensemble of Support Vector Machine Variants. *Proceedings of 5th International Conference on Recent Advances in Soft Computing*, vol. 1, pp. 670-675, Nottingham, 2004.

Lima C.A.M.; Coelho A.; **Puma-Villanueva W. J.**; Von Zuben F. J. Gated Mixtures of Least Squares Support Vector Machine Experts Applied to Classification Problems. *Proceedings of the 5th International Conference on Recent Advances in Soft Computing*, vol. 1, pp. 494-499, Nottingham, 2004.

Lima C.A.M.; **Puma-Villanueva W.J.**; dos Santos E.P.; Von Zuben F. J. Mistura de especialistas aplicada à predição de séries temporais financeiras. *VIII Simpósio Brasileiro de Redes Neurais - SBRN*, vol. 1, no. 3708, São Luís, MA, 2004.

Referências Bibliográficas

Albertson, K.; Aylen, J. Modelling the Great Lake freeze: Forecasting and seasonality in the market for ferrous scrap. *International Journal of Forecasting*, vol. 12, pp. 345-359, 1996.

Allwein, E.; Schapire, R.; Singer, Y. Reducing multiclass to binary: A unifying approach for margin classifiers. In *Machine Learning: Proceedings of the Seventeenth International Conference*, 2000.

Almeida, L. B. A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. In *Proceeding of the IEEE 1st International Conference on Neural Networks*, June 21-24, pp. 609-618, San Diego, CA, 1987.

Alon, U; Barkai, N.; Notterman, D. A.; Gish, K.; Ybarra, S.; Mack, D.; Levine, A. J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In *Proceedings of the National Academy of Science USA*, vol. 96, pp. 6745-6750, 1999.

Barto, A. G.; Anandan, P. Pattern recognizing stochastic learning automata. *IEEE Transactions on Systems, Man and Cybernetics*, 1985.

Bates, J. M.; Granger, C. W. J. Combination of forecasts. *Operations Research Quarterly*, vol. 20, pp. 451-468, 1969.

Battiti, R. First and second-order methods for learning: between steepest descent and newton's method. *Neural Computation*, vol. 4, no. 2, pp. 141-166, 1992.

Battiti, R.; Colla, A. M. Democracy in neural nets: Voting schemes for classification. *Neural Networks*, vol. 7, no. 4, pp. 691-707, 1994.

Benediktsson, J. A.; Swain, P. H. Consensus theoretic classification methods. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 4, pp. 688-704, 1992.

Bishop C. M. **Neural Networks for Pattern Recognition**. Clarendon Press, Oxford, 1995.

Bollerslev, T.; Engle, R. F.; Nelson, D. B. ARCH models. In R. F. Engle, & D. L. McFadden (Eds.), Handbook of econometrics, vol. IV. Amsterdam North-Holland, pp. 2959-3038, 1994.

Box, G. E. P.; Jenkins, G. M.; Reinsel, G. C. **Time Series Analysis: Forecasting and Control**. 3nd ed. Holden-Day, 1994.

Brace, M.C.; Schmit, J.; Hadlin, M. Comparison of forecasting accuracy of neural networks with other established techniques, In *Proceedings of the First International Forum on ANNPS*, Seattle, 1991.

Breiman, L. Bagging predictors. *Machine Learning*, vol. 24, pp. 123-140, 1996.

Breiman, L. Combining predictors. In **Combining artificial neural nets**, Sharkey, A. J. C., Ed., Springer, 31, 1999.

Breiman, L. Randomizing outputs to increase prediction accuracy. *Machine Learning*, vol. 40, pp. 229-242, 2000.

Breiman, L. Random Forests. *In Machine Learning*, vol. 45, pp. 5-32, 2001.

Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. Classification and Regression Trees. **Wadsworth International Group**, Belmont, CA, 1984.

Brockwell P.; Davis, R. **Time series: theory and methods**. Springer, N.Y., 1991.

Broomhead D. S.; Lowe D. Multivariate functional interpolation and adaptive networks. *Complex Systems* 2, pp. 321-355, 1988.

Caire, P.; Hatabian, G.; Muller, C. Progress in forecasting by neural networks. In *Proceedings of the International Joint Conference on Neural Networks*, vol. 2, pp. 540-545, 1992.

Carpenter G. A.; Grossberg S. The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network. *IEEE Computer*, vol. 21, no. 3, pp.77-88, 1988.

Cellucci, C.J.; Albano, A. M.; Rapp, P. E. Statistical validation of mutual information calculations: Comparison of alternative numerical algorithms. *Physical Review E*, vol. 71, issue 6, id. 066208, pp. 1-14, 2005.

Chakraborty, K.; Mehrotra, K.; Mohan, C. K.; Ranka S. Forecasting the behavior of multivariate time series using neural networks. *Neural Networks*, vol. 5, pp. 961-970, 1992.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

Cherkauer, K. J. Human expert level performance on a scientific image analysis task by a system using combined artificial neural networks. In P. Chan, S. Stolfo, D. Wolpert (Eds), *Proceeding AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, Portland, OR, AAAI Pres, Menlo Park, CA, pp. 15-21, 1996.

Cho, S. B.; Kim, J. H. Combining multiple neural networks by fuzzy integral for robust classification. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 2, pp. 380-384, 1995.

Coelho G. Geração Seleção e Combinação de Componentes para Ensembles de Redes Neurais Aplicadas a Problemas de Classificação. Tese de Mestrado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, 2006.

Connor J. T.; Martin, R. D. Recurrent neural networks and robust time series prediction, *IEEE Transactions on Neural Networks*, vol. 5, pp. 240-254, 1994.

Cover, T.; Thomas, J. **Elements of Information Theory**. John Wiley & Sons, New York, NY, 1991.

Cryer, J. D. **The time series analysis**. Duxbury Press, 1986.

Cybenko G. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control Signals and Systems*, vol. 2, pp. 303-314, 1989.

Das S. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 74-81, 2001.

Dasarathy, B. V.; Sheela, B.V. Composite classifier system design: Concepts and methodology. *Proceedings of the IEEE*, vol. 67, no. 5, pp. 708-713, 1979.

De Groot, C; Wurtz, D. Analysis of univariate time series with connectionist nets: a case study of two classical examples, *Neuro Computing*, vol. 3, pp. 177- 92, 1991.

Denton, J. W. How good are neural networks for causal forecasting?. *Journal of Business Forecasting Methods and Systems*, vol. 14, no. 2, pp. 17-21, 1995.

Dempster, A. P.; Laird, N. M.; Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B, vol. 39, no 1, pp. 1-38, 1977.

Dietterich T. G. Ensemble methods in machine learning. In *Proceedings of the International Workshop on Multiple Classifier Systems (MCS), LNCS 1857*, pp. 1-15, Italy, Springer, 2000.

Duliba, K. A. Contrasting neural nets with regression in predicting performance in the transportation industry. In *Proceedings of the 24th Annual Hawaii International Conference on System Sciences*, vol. 4, pp. 163-170, 1991.

Drucker, H.; Cortes, C.; Jackel, L. D.; LeCun, Y.; Vapnik, V. Boosting and other ensemble methods. *Neural Computation*, vol. 6, no. 6, pp. 1289-1301, 1994.

Engle, R. F. Autoregressive conditional heteroscedasticity with estimates of the variance of the United Kingdom inflation. *Econometrica*, vol. 50, pp. 987-1008, 1982.

Fildes, R.; Makridakis, S. The impact of empirical accuracy studies on time series analysis and forecasting. *International Statistical Review*, vol. 63, pp. 289-308, 1995.

Findley, D. F.; Monsell, B. C.; Bell, W. R.; Otto, M. C.; Chen., B. C. New capabilities and methods of the X-12-ARIMA seasonal adjustment program. *Journal of Business and Economic Statistics*, vol. 16, pp. 127-152, 1998.

Fishwick, P.A. Neural network models in simulation: A comparison with traditional modeling approaches. In *Proceedings of Winter Simulation Conference*, pp. 702-710, 1989.

Fleuret F. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, vol. 5, pp. 1531-1555, 2004.

Foster, B.; Collopy, F.; Ungar, L. H. Neural Network Forecasting of Short Noisy Time Series. *Computers and Chemical Engineering*, vol. 16, no. 4, pp. 293-298, 1992.

Franses, P. H.; van Dijk, D. The forecasting performance of various models for seasonality and nonlinearity for quarterly industrial production. *International Journal of Forecasting*, vol. 21, pp. 87-102, 2005.

Fraser A. M.; Swinney, H. L. Independent coordinates for strange attractors from mutual information. *Physical Review, A*, vol. 33, pp. 1134-1140, Feb. 1986.

Fred, A. L. N.; Jain, A. K. Data clustering using evidence accumulation. In *Proceedings of the International Conference on Pattern Recognition*, pp. 276-280, 2002.

Freund, Y.; Schapire, R. E. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 148-156, 1996.

Freund, Y.; Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, 1997.

Friedman, J. H. Multivariate adaptive regression splines. *Annals Statistics*, vol. 19, pp. 1-41, 1991.

Friedman J.; Hastie T.; Tibshirani, R. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, vol. 28, no. 2, 2000.

Gomes, L. C. T.; Von Zuben, F. J. Vehicle routing based on self-organization with and without fuzzy inference. Proceedings *of the IEEE International Conference on Fuzzy Systems* (FUZZ-IEEE'2002), in the 2002 IEEE World Congress on Computational Intelligence (WCCI'2002), Honolulu, Hawaii, vol. 2, pp. 1310-1315, May 12-17, 2002.

Gorr, W. L.; Daniel, N.; Szczypula, J. Comparative study of artificial neural network and statistical models for predicting student grade point averages. *International Journal of Forecasting*, vol. 10, pp. 17-34, 1994.

Granger, C. W. J.; Joyeux, R. An introduction to long-memory time series models and fractional diferencing. *Journal of Time Series Analysis*, vol. 1, no. 1, pp 15-29, 1980.

Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.

Hall, D. L.; Llinas, J. (eds.) **Handbook of Multisensor Data Fusion**. CRC Press, 2001.

Hamming R. W. **Coding and Information Theory**, 2nd Ed., Prentice-Hall Inc., 1986.

Hampshire, J.; Waibel, A. A novel objective function for improved phoneme recognition using time-delay neural networks. *IEEE Transactions on Neural Networks*, vol. 1, no. 2, pp. 216-228, 1990.

Hans, P.; Dijk, D. V. **Nonlinear Time Series Models in Empirical Finance**. Cambridge University Press. 2000.

Hansen, L. K.; Salamon, P. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993-1001, 1990.

Hashem S. Optimal linear combinations of neural networks. *Neural Networks*, vol. 10, no 4, pp. 599-614, 1997.

Haykin S. **Neural Networks: A Comprehensive Foundation**, 2nd Edition. Prentice-Hall, 1999.

Hebb, D. O. **Organization of Behaviour: A Neuropsychological Theory**. Wiley, New York, 1949.

Hinton, G. E.; Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science Magazine*, vol. 313. no. 5786, pp. 504 - 507, 28 July 2006.

Ho, T. K.; Hull, J. J.; Srihari, S. N. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis Machine Intelligent*, vol. 16, no. 1, pp. 66-75, 1994.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Sciences*, vol. 79. pp. 2554-2558, April, 1982.

Hornik, K.; Stinchombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, no 2, pp.359-366, 1989.

Hosking J. R. M. Fractional diferencing. *Biometrika*, vol. 68, no. 1, pp. 165-176, 1981.

Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; Hinton, G. E. Adaptive mixtures of local experts. *Neural Computation*, vol. 3, no 1, pp. 79-87, 1991.

Jain A. K.; Mao J. Artificial Neural Networks: A Tutorial. *IEEE Computer*, pp. 31-44, 1996.

John, G. H.; Kohavi, R.; Pfleger, K. Irrelevant feature and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 121-129, 1994.

Jordan, M. I.; Jacobs, R. A. Hierarchical Mixtures of experts and the EM algorithm. *Neural Computation*, MIT Press , vol. 6, pp. 181-214, 1994.

Kang, S. An investigation of the use of feedforward neural networks for forecasting. Ph. D. Thesis, Kent State University, 1991.

Karras, D. A.; Vlitakis, C. E.; Boutalis, Y. S.; Mertzios, B.G. Optimizing the structure of hierarchical mixture of experts using genetic algorithms. In *Proceedings of the 2nd International IEEE Conference of Intelligent Systems*, vol. 1, pp. 144-149, 2004.

Kittler J.; Hatef M.; Duin R.; Matas J. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226-239, 1998.

Kleinberg E. M. On the algorithmic implementation of stochastic discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 5, pp. 473-490, 2000.

Kohavi, R.; John, G. Wrappers for feature subset selection. *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273-324, 1997.

Kohonen, T. **Self-Organization and Associative Memory**, 3nd edition, Springer-Verlag New York, New York, 1989.

Kohzadi, N.; Boyd, M. S.; Kermanshahi, B.; Kaastra, I. A comparison of artificial neural network and time series models for forecasting commodity prices. *Neurocomputing*, vol. 10, pp. 169-181, 1996.

Kulendran, N.; King, M. L. Forecasting international quarterly tourist flows using error-correction and time-series models. *International Journal of Forecasting*, vol. 13, pp. 319-327, 1997.

Kuncheva, L. I. Classifier ensembles for changing environments. *5th Interantional Workshop on Multiple Classifier Systems*, Lecture Notes in Computer Science, F. Roli, J. Kittler, and T. Windeatt (eds.), vol. 3077, pp. 1-15, 2004.

Kuncheva, L. I.; Bezdek, J. C.; Duin, R. Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition*, vol. 34, no. 2, pp. 299-314, 2001.

Kusiak, A. Feature Transformation Methods in Data Mining. *IEEE Transactions on Electronics Packaging Manufacturing*, vol. 24, no. 3, pp. 214-221, 2001.

Lachtermacher, G.; Fuller, J. D. Backpropagation in time-series forecasting. *Journal of Forecasting*, vol. 14, pp. 381-93, 1995.

Li Ch.; Shujun Li; Dan Zhang; Guanrong Chen. Cryptanalysis of a Chaotic Neural Network Based Multimedia Encryption Scheme. **Book Series Lecture Notes in Computer Science Publisher Springer Berlin / Heidelberg**, vol. 3, pp. 418-425, 2004.

Lima, C. A. M. Comitê de máquinas: Uma abordagem unificada empregando máquinas de vetores-suporte. Tese de doutorado, DCA-FEEC/UNICAMP Campinas/SP, Brasil, 378 p, 2004.

Lima, C. A. M.; Puma-Villanueva W. J.; dos Santos, E. P.; Von Zuben, F. J. A multistage ensemble of support vector machine variants. In *Proceedings of the 5th International Conference on Recent Advances in Soft Computing*, vol. 1. pp. 670-675, Nottingham, 2004a.

Liu H.; Motoda H. Feature transformation and subset selection. *IEEE Intelligent Systems*, vol. 13, no. 2, pp. 26-28, 1998.

Liu H.; Yu L. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering - TKDE*, vol. 17, no. 4, pp. 491-502, 2005.

Maclin, R.; Shavlik, J. W. Combining the predictions of multiple classifiers: using competitive learning to initialize neural networks. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence IJCAI-95*, Montreal, Canada, Morgan Kaufmann, San Mateo, CA, pp. 524-530, 1995.

Makridakis, S. Forecasting Accuracy and System Complexity. *RAIRO*, vol. 29, no. 3, pp. 259-283, 1995.

Makridakis, S.; Andersen, A.; Carbone, R.; Fildes, R.; Hibon, M., Lewandowski, R., et al. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, vol. 1, pp. 111–153, 1982.

Mandelbrot, B. B.; Van Ness, J. W. Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Review*, vol. 10, no. 4, pp. 422-437, 1968.

Marquez, L.; Hill, T.; O'Connor, M.; Remus, W. Forecasting with neural networks: A review. In *Proceedings of the Hawaii International Conference on System Sciences*, vol. 4, pp. 494-498, 1992.

McCleary, R.; Hay, R. A. **Applied Time Series Analysis for the Social Sciences**. Beverly Hills, CA.: Sage, 1980.

McLachlan, G. J.; Basford, K. E. **Mixture Models: Inference and Applications to Clustering**. Marcel Dekker, Inc., New York, 1988.

McLeod, A. I. Parsimony, model adequacy and periodic correlation in forecasting time series, *International Statistical Review*, vol. 61, pp. 387-393, 1993.

Minsky, M. L.; Papert, S. A. **Perceptrons**. Cambridge, MA: MIT Press, 1969.

Mitchell, T. **Machine Learning**. New York: McGraw-Hill, 1997.

Moody J. Prediction Risk and Neural Network Architecture Selection. In **From Statistics to Neural Networks: Theory and Pattern Recognition Applications**, V. Cherkassky, J. H. Friedman, and H. Wechsler (Eds.), Springer-Verlag, 1994.

Moody J.; Utans, J. Principled architecture selection for neural networks: Application to corporate bond rating prediction. In J. Moody, S. J. Hanson, and R. P. Lippmann, editors, **Advances in Neural Information Processing Systems**, volume 4. Morgan Kaufmann, 1991.

Naftaly U.; Intrator, N.; Horn, D. Optimal ensemble averaging of neural networks. *Network*, vol. 8, pp. 283-296, 1997.

Nam, K.; Schaefer, T. Forecasting international airline passenger traffic using neural networks. *Logistic and Transportation*, vol. 31, no. 3, pp. 239-251, 1995.

Narendra, K. S.; Thathachar, M. A. L. Learning automata: A survey. *IEEE Transactions in Systems, Man and Cybernetics*, vol. SMC-4, no. 4, 1974.

Narendra, K. S.; Parthasarathy, K. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, vol. 1, pp. 4-27. 25, 1990.

Narendra, K.S.; K. Parthasarathy, Gradient Methods for The optimisation of dynamical systems Containing Neural Networks. *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 252-262, 1991.

Nelson, M.; Hill, T.; Remus, T.; O'Connor, M. Time series forecasting using NNs: Should the data be deseasonalized first. *Journal of Forecasting*, vol. 18, pp. 359–367, 1999.

Newbold, P.; Granger, C. W. J. Experience with forecasting univariate time series and the combination of forecasts. *Journal of the Royal Statistical Society*. Series A, 137, pp. 131-146, 1974.

Noakes, D. J.; McLeod, A. I.; Hipel, K. W. Forecasting monthly riverflow time series. *International Journal of Forecasting*, vol. 1, pp. 179-190, 1985.

Nørgaard M.; Ravn O.; Poulsen, N. K.; Hansen, L. K. **Neural Networks for Modelling and Control of Dynamic Systems**, Springer-Verlag, London, 2000.

Opitz, D. W.; Schavlik, J. W. Actively searching for an effective neural network ensemble. *Connection Science*, vol. 8, no. 3 & 4, pp. 337-353, 1996.

Perrone, M. P.; Cooper, L. N. When networks disagree: Ensemble methods for hybrid neural networks. In R. J. Mammone, editor, **Neural Networks for Speech and Image Processing**, chapter 10. Chapman-Hall, 1993.

Pineda, F. J. Generalization of back-propagation to recurrent neural networks. *Physical Review Letters*, vol. 59, no. 19, pp. 2229-2232, 1987.

Polikar, R. Ensemble based systems in decision making. IEEE *Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21-45, 2006.

Poskitt, D. S.; Tremayne, A. R. The selection and use of linear and bilinear time series models. *International Journal of Forecasting*, vol. 2, pp. 101-114, 1986.

Prechelt L., Early Stopping - but when?, Technical Report, 1997 URL: http://wwwipd.ira.uka.de/~prechelt/Biblio/stop_tricks1997.ps.gz

Prechelt L. Automatic early stopping using cross validation: Quantifying the criteria. *Neural Networks*, vol. 11, no 4, pp. 761-767, 1998.

Puma-Villanueva W. J.; Lima C. A. M.; dos Santos E. P.; Von Zuben F. J. Mixture of Heterogeneous Experts Applied to Time Series: A Comparative Study. *Proceedings of the IEEE International Joint Conference on Neural Networks - IJCNN*, vol. 1, pp. 1160-1165, Montreal, 2005.

Puma-Villanueva, W. J.; dos Santos, E. P.; Von Zuben, F. J. Data partition and variable selection for time series prediction using wrappers. *Proceedings of the IEEE International Joint Conference on Neural Networks - IJCNN*, vol. 1, pp. 9490-9497, Vancouver, 2006.

Quinlan, J. R. Induction of decision trees. *Machine Learning*, vol. 1, pp. 81-106, 1986.

Rakotmamonjy, A. Variable selection using SVM-based criteria. *Journal of Machine Learning Research*, vol. 3, pp. 1357–1370, 2003.

Ramamurti, V.; Ghosh, J. Structural adaptation in mixture of experts. In *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 4, pp. 704-708, 1996.

Raviv, Y.; Intrator, N. Variance reduction via noise and bias constraints. In *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems* (ed. A. J. C. Sharkey). London, Springer-Verlag, pp. 163-175, 1999.

Refenes, A. P. Constructive learning and its application to foreign exchange forecasting. In Trippi, R. R. y Turban, E. (eds.): *Neural networks in finance and investing*. Probus Publishing Company, Chicago, pp. 465-493, 1993.

Rodríguez J. J.; Kuncheva, L. I.; Alonso C. J. Rotation Forest: A new classifier ensemble method. *IEEE Transaction on Pattern Analysis and Machine Intelligence*. vol. 28, no. 10, pp. 1619-1630, 2006.

Rogova, G. Combining the results of several neural network classifiers. *Neural Networks*, vol. 7, no. 5, pp. 777-781, 1994.

Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, vol. 65, pp. 386-408, 1958.

Rumelhart D. E.; McClelland J.L. Parallel distributed processing: Exploration in the microstructure of cognition, *MIT Press*, Cambridge, Massachussetts, vol. 1, 1986.

Schapire, R. E..The strength of weak learnability. *Machine Learning*, vol. 5, no. 2, pp. 197-227, 1990.

Shannon, C. E.; Weaver, W. **The mathematical theory of communication**. Urbana IL: University of Illinois Press, 1949.

Sharda, R.; Patil, R. B. Neural networks as forecasting experts: An empirical test. *International Joint Conference on Neural Networks*, vol. 1, pp. 491-494, Washington D.C. IEEE, 1990.

Sharda, R.; Patil, R. B. A connectionism approach to time series prediction: An empirical test", *Journal of Intelligent Manufacturing*, Chapman Hall, *Neural Networks in Finance and Investing*. Trippi, R. R. y Turban, E. (eds.). Probus Publishing Company, Chicago, 1992.

Sharkey A. (Ed.). **Combining artificial neural nets: Ensemble and modular multi-net systems**. Springer-Verlag, London, 1999.

Smieja, F. Pandemonium system of reflective agents. *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 97-106, 1996.

Srinivasan, D.; Liew, A. C.; Chang, C. S. Neural network short term load forecaster. *Electric Power Systems Research*, vol. 27, no. 3, pp. 227-234, 1994.

Strehl A.; Ghosh, J. Cluster ensembles - a knowledge reuse framework for combining partitionings. In *Proceedings of the Conference on Artificial Intelligence* (AAAI 2002), Edmonton, AAAI/MIT Press, pp 93-98, July 2002a.

Strehl, A.; Ghosh, J. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal on Machine Learning Research (JMLR), vol. 3, pp 583-617, December 2002b.

Tang, Z.; de Almeida, C.; Fishwick, P. A. Time series forecasting using neural networks vs. Box-Jenkins methodology. *Simulation*, vol. 5, no. 7, pp. 303-310, 1991.

Tang, Z.; Fishwick, P. A. Feed-forward neural nets as models for time series forecasting. *ORSA Journal of Computing*, vol. 5, no. 4, pp. 374-386, 1993.

Taniguchi, M.; Tresp, V. Averaging regularized estimators. *Neural Computation*, vol. 9, pp. 1163-1178, 1997.

Taylor, S. J. Forecasting the volatility of currency exchange rates. *International Journal of Forecasting*, vol. 3, pp. 159-170, 1987.

Tong, H. **Threshold models in non-linear time series analysis**. New York, Springer-Verlag, 1983.

Tong, H. **Non-linear time series: A dynamical system approach**. Oxford, Clarendon Press, 1990.

Ueda, N.; Ghahramani, Z. Optimal model inference for Bayesian mixture of experts. *Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop*, vol. 1, no. 11-13, pp. 145-154, 2000.

Vapnik, V. **Statistical Learning Theory**. John Wiley and Sons, New York, 1998.

Vapnik V. N.; Chervonenkis A. Y. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, vol. 16, pp. 264-280, 1971.

Volterra, V. **Theory of functionals and of integro-differential equations**. New York-Dover, 1930.

Von Zuben F. J. Modelos Paramétricos e Não-Paramétricos de Redes neurais Artificiais e Aplicações, Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, 1996.

Wang, G., Lochovsky, F. H. Feature selection with conditional mutual information maximin in text categorization", In *Proceedings of the Conference on Information and Knowledge Management*, pp. 342-349, 2004.

Waterhouse, S. R.; Robinson, A. J.; Pruning and growing hierachical mixtures of experts. In *Proceedings of the Fourth International Conference on Artificial Neural Networks*, pp 341-346, 1995.

Weigend, A. S.; Huberman, B. A.; Rumelhart, D. E. Predicting sunspots and exchange rates with connectionist networks. In *Nonlinear Modeling and Forecasting*, edited by Casdagli, M.; Eubank, S. pp. 395-432. Redwood City, CA: Addison-Wesley, 1992.

Weigend, A. S.; Mangeas, M.; Sristava, A. N. Nonlinear gated experts for time series: Discovering regimes and avoiding over fitting. *International Journal of Neural Systems*, vol. 6, pp. 373-399, 1995.

Werbos, P. Beyond regression: New tools for prediction and analysis in the behavioral sciences, PhD thesis, Harvard University, 1974.

Widrow B.; Sterns, S. D. **Adaptive Signal Processing**, New York: Prentice-Hall, 1985.

Wolpert, D. H. Stacked generalization. *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

Woods, K.; Kegelmeyer, W. P. J.; Bowyer, K. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 405-410, 1997.

Xu, L.; Krzyzak, A.; Suen, C. Y. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 3, pp. 418-435, 1992.

Xu, L., Jordan, M. I.; Hinton, G. E. An Alternative model for mixtures of experts. *Advances in Neural Information Processing Systems 7*, eds., Cowan; J. D., Tesauro, G.; Alspector, J., MIT Press, Cambridge MA, pp. 633-640, 1995.

Yao, X.; Liu, Y. Making use of population information in evolutionary artificial neural networks. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, vol. 28, no 3, pp. 417-424, 1998.

Yu, L.; Liu, H. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, vol. 5, pp. 1205-1224, 2004.

Zhang, G., B. E. Patuwo, M. Y. Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, vol. 14, no. 1, pp. 35-62, 1998.

Zhang, G. P., Qi, M. Neural network forecasting for seasonal and trend time series. *European Journal of Operation Research*, vol. 160, pp. 501-514, 2005.