



Pós-Graduação em Ciência da Computação

**“Seleção de Modelos de Previsão de Séries Temporais
baseada em Informações de Desempenho”**

Por

Patrícia Maforte dos Santos

Dissertação de Mestrado



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, 08/2006



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

PATRÍCIA MAFORTE DOS SANTOS

“Seleção de Modelos de Previsão baseada em
Informações de Desempenho”

*ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA
UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO
REQUISITO PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE
EM CIÊNCIA DA COMPUTAÇÃO.*

ORIENTADORA: PROF^ª. DR^ª. TERESA BERNARDA LUDERMIR (CIN-UFPE)
CO-ORIENTADOR: PROF^º. DR^º. RICARDO BASTOS CAVALCANTE PRUDÊNCIO (CAC-UFPE)

RECIFE, AGOSTO/2006

RESUMO

Uma série temporal é definida como um conjunto de observações de um fenômeno ordenadas no tempo. Existem vários problemas reais que podem ser representados por séries temporais, como o consumo mensal de água de uma casa, registrado ao longo de um mês; ou os valores de uma determinada aplicação financeira, medidos no decorrer de uma semana.

A utilização da previsão de séries temporais pode ocorrer em diversas áreas, como mercado financeiro, detecção de fraude, indústria farmacêutica, medicina, entre outras. Existem vários modelos que podem ser utilizados para prever uma série temporal. Com isso, selecionar o modelo mais adequado pode ser uma tarefa difícil, que depende de fatores como o ajuste dos parâmetros dos modelos candidatos e as características da série.

Podemos encontrar na literatura diversas abordagens que são utilizadas na seleção de modelos de previsão. Em nosso trabalho foi utilizada uma abordagem de Meta-Aprendizado, desenvolvida inicialmente para a seleção de algoritmos para problemas de aprendizado e adaptada ao problema de seleção de modelos. Diferentemente das abordagens mais comuns, a abordagem utilizada indica não apenas o melhor modelo aplicável ao problema de entrada, mas um ranking dos modelos candidatos baseado em critérios de desempenho fornecidos pelo usuário. Os resultados de desempenho obtidos pelos modelos candidatos em problemas processados no passado são utilizados na sugestão de modelos para novos problemas. Desta forma, a solução aqui proposta é mais informativa, no sentido de possibilitar ao usuário uma melhor percepção da relação entre os modelos candidatos. A abordagem foi investigada em 4 estudos de caso e apresentou resultados satisfatórios.

Palavras-chave: Séries Temporais, Previsão, Meta-Aprendizado.

ABSTRACT

A time series is defined as a sequence of observations, which are ordered in time. There are several real problems that can be represented as time series, such as the monthly water consumption, registered during a month; or the values of a financial application, measured during a week.

The use of time series forecasting can occur in several areas, as the financial market, fraud detection, chemist industry, medicine, among others. There are several models that can be used to forecast a time series. Thus, the selection of the most appropriate model to use can be a difficult task, which depends on different factors as the adjustment of the candidate models parameters and the time series characteristics.

We can find in the literature many approaches that are used for the selection of forecasting models. In this work, we used an Meta-Learning approach, initially developed to the selection of learning algorithms and adapted to the model selection problem. Differently from the most common solutions to model selection, the approach used indicates not only the best model which is applicable to the input problem, but it also provides a ranking of the candidate models based on performance criteria provided by user. The performance results obtained by the candidate models on past problems are used for the suggestion of models to new problems. Thus, the proposed solution is more informative, giving the user a better perception of the relation between the candidate models.

Key-words: Time Series, Forecasting, Meta-Learning.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus por ter me concedido a oportunidade de ter ingressado e concluído a graduação em Ciência da Computação na UFPE, algo que alcancei com muito esforço e perseverança.

Aos meus pais, por acreditarem em mim sempre, reconhecendo meu esforço e dedicação e compreendendo meus momentos de instabilidade emocional causados pelo acúmulo de atividades.

Agradeço ao meu namorado Hugo, pelo amor e compreensão em todos os momentos.

Agradeço à minha orientadora Teresa, pela disponibilidade e apoio em todos os momentos e por ter me dado a oportunidade de ser sua aluna desde a época de Iniciação Científica até o Mestrado.

Ao meu co-orientador Ricardo, pela ajuda de extrema importância em pontos fundamentais deste trabalho.

ÍNDICE ANALÍTICO

1.	Introdução	1
1.1.	Contexto da Dissertação	1
1.2.	Objetivos.....	2
1.3.	Trabalho realizado	3
1.4.	Conteúdo da dissertação	4
2.	Séries Temporais.....	5
2.1.	Previsão de Séries Temporais.....	5
2.2.	Avaliação de um Modelo de Séries Temporais	6
2.3.	Seleção de Modelos de Séries Temporais	8
2.3.1.	Abordagem por torneio	8
2.3.2.	Aprendizado de máquina.....	11
2.3.2.1.	Meta-Aprendizado.....	12
2.4.	Considerações finais	14
3.	Abordagens de Meta-Aprendizado	15
3.1.	Abordagem simples	16
3.2.	Algoritmos baseados em instâncias.....	17
3.3.	Abordagem NOEMON.....	18
3.4.	Abordagem Zoomed Ranking	19
3.4.1.	Seleção dos exemplos mais similares.....	20
3.4.2.	Geração do ranking baseada na precisão e no tempo	20
3.5.	Considerações Finais	22
4.	Metodologia.....	24
4.1.	Arquitetura geral.....	24
4.2.	Aspectos de Implementação	26
4.2.1.	Séries temporais	26
4.2.2.	Modelos candidatos.....	28
4.2.3.	Abordagem de Meta-Aprendizado	29
4.2.4.	Ranking recomendado	29
4.3.	Considerações finais	31
5.	Análise Experimental.....	32
5.1.	Estudo de caso I.....	32

5.1.1.	Módulo extrator de características.....	32
5.1.2.	Base de dados.....	33
5.1.2.1.	Meta-exemplos.....	34
5.1.2.2.	Desempenho dos modelos.....	35
5.1.3.	Módulo Meta-Aprendiz.....	37
5.1.4.	Experimentos e Resultados.....	39
5.2.	Estudo de caso II.....	41
5.2.1.	Módulo extrator de características.....	42
5.2.2.	Base de dados.....	42
5.2.3.	Módulo Meta-Aprendiz.....	42
5.2.4.	Experimentos e Resultados.....	44
5.3.	Estudo de caso III.....	46
5.3.1.	Módulo extrator de características.....	46
5.3.2.	Base de dados.....	46
5.3.3.	Módulo Meta-Aprendiz.....	47
5.3.4.	Experimentos e Resultados.....	47
5.4.	Estudo de caso IV.....	50
5.4.1.	Módulo extrator de características.....	50
5.4.2.	Base de dados.....	50
5.4.3.	Módulo Meta-Aprendiz.....	50
5.4.4.	Experimentos e Resultados.....	51
5.5.	Considerações finais.....	52
6.	Conclusão.....	53
6.1.	Contribuições.....	55
6.2.	Trabalhos Futuros.....	55
Apêndice A.	Rede SOM.....	57
A.1.	Arquitetura.....	57
A.2.	Treinamento.....	58
Apêndice B.	Modelos de Séries Temporais.....	61
B.1.	Rede Neural SOMTAD.....	61
B.1.1.	Estruturas de Memória para Seqüências Temporais.....	61
B.1.2.	Auto-organização com Reações de Difusão.....	62
B.1.3.	O MODELO SOMTAD.....	62
B.1.4.	Descrição do Algoritmo.....	64
B.2.	Rede Neural RBF.....	66
B.2.1.	Arquitetura.....	66

B.2.2. Treinamento	68
B.3. Rede Neural TDNN	68
Referências.....	70

LISTA DE FIGURAS

Figura 4.1: Arquitetura geral da solução proposta	25
Figura 5.1.1: Exemplo de uma série da EUNITE.....	34
Figura 5.1.2: Algoritmo do Zoomed Ranking	38
Figura 5.2: Algoritmo do Zoomed Ranking com uso da Rede SOM.....	44
Figura A.1: Arquitetura de uma rede SOM bidimensional	57
Figura A.2: Formatos possíveis da região de vizinhança para uma rede SOM. (a) Forma retangular. (b) Forma hexagonal	59
Figura B.1: Arquitetura da rede SOMTAD.....	63
Figura B.1.2: Enhancement com $\mu = 0.5$ (esquerda) e $\mu = 0.75$ (direita) da SOMTAD	65
Figura B.2: Rede de função de base radial	67

LISTA DE TABELAS

Tabela 5.1: Médias das medidas MAE (Média dos Erros Absolutos) no conjunto de treinamento dos meta-exemplos das séries da EUNITE	37
Tabela 5.2: Média dos tempos de execução dos modelos candidatos no conjunto de treinamento dos meta-exemplos das séries da EUNITE	37
Tabela 5.3: Ranking gerado para $Z = 10$ para as séries da EUNITE.....	40
Tabela 5.4: Ranking gerado para $Z = 50$ para as séries da EUNITE.....	40
Tabela 5.5: Ranking gerado para $Z = 90$ para as séries da EUNITE.....	40
Tabela 5.6: Média dos coeficientes de Spearman para as séries da EUNITE	41
Tabela 5.7: Ranking recomendado com o uso da Rede SOM para as séries da EUNITE	45
Tabela 5.8: Média dos coeficientes de Spearman para as séries da EUNITE com o uso da Rede SOM	45
Tabela 5.9: Médias das medidas MAE (Média dos Erros Absolutos) no conjunto de treinamento dos meta-exemplos das séries da TSDL.....	47
Tabela 5.10: Média dos tempos de execução dos modelos candidatos no conjunto de treinamento dos meta-exemplos das séries da TSDL.....	47
Tabela 5.11: Ranking gerado para $Z = 10$ para as séries da TSDL	48
Tabela 5.12: Ranking gerado para $Z = 13$ para as séries da TSDL	48
Tabela 5.13: Ranking gerado para $Z = 16$ para as séries da TSDL	48
Tabela 5.14: Médias das medidas MAE (Média dos Erros Absolutos) no conjunto de teste dos meta-exemplos das séries da TSDL.....	49
Tabela 5.15: Média dos tempos de execução dos modelos candidatos no conjunto de teste dos meta-exemplos das séries da TSDL.....	49
Tabela 5.16: Média dos coeficientes de Spearman para as séries da TSDL	49
Tabela 5.17: Ranking recomendado com o uso da Rede SOM para as séries da TSDL	51
Tabela 5.18: Média dos coeficientes de Spearman com o uso da Rede SOM para as séries da TSDL	52

LISTA DE EQUAÇÕES

Equação 2.1: Valor previsto de uma série retornado por um modelo de previsão	6
Equação 2.2: Soma dos erros quadráticos	6
Equação 2.3: Média dos erros quadráticos	7
Equação 2.4: Média dos erros absolutos	7
Equação 2.5: Mediana dos erros quadráticos	7
Equação 3.1: Métrica da <i>Norma Sem Peso</i> L_1	20
Equação 3.2: Métrica ARR (Adjusted Ratio of Ratios)	21
Equação 3.3: Métrica ARR para o algoritmo p	22
Equação 4.1: Descrição formal de um meta-exemplo	25
Equação 4.2: Operador de diferenças simples	27
Equação 4.3: Ranking ideal gerado pelo Zoomed Ranking	30
Equação 4.4: Coeficiente de correlação de Spearman	30
Equação 5.1: Taxa de aprendizado para treinamento da rede SOM	42
Equação A.1: Distância mínima euclidiana	58
Equação A.2: Ajuste do vetor de peso da rede SOM	58
Equação A.3: Taxa de aprendizado da rede SOM	59
Equação B.1.1: Função de ativação no modelo SOMTAD	64
Equação B.1.2: Definição da unidade vencedora da SOMTAD	64
Equação B.1.3: Regra de aprendizado de Kohonen	64
Equação B.1.4: Função Gaussiana	65
Equação B.1.5: <i>Enhancement</i> temporal da SOMTAD	65
Equação B.1.6: Equação para a taxa de aprendizado da SOMTAD	66
Equação B.1.7: Equação do parâmetro de acoplamento β da SOMTAD	66
Equação B.2.1: Função multiquadrática inversa	67
Equação B.2.2: Função de saída da rede RBF	68

1. INTRODUÇÃO

Uma série temporal é definida então como um conjunto de observações de um fenômeno ordenadas no tempo [Box, Jenkins 1970]. Existem vários problemas reais que podem ser representados por séries temporais, como o consumo mensal de água de uma casa, registrado ao longo de um mês; os valores de uma determinada aplicação financeira, medidos no decorrer de uma semana; ou ainda os valores da tensão elétrica de um dispositivo, medidos durante um dia.

A previsão de séries temporais é um processo de análise, que envolve a identificação das características e propriedades importantes da série, utilizadas para descrever o seu fenômeno gerador. A utilização da previsão de séries temporais em problemas reais pode ocorrer em diversas áreas, como mercado financeiro, segurança em redes de computadores, detecção de fraude, indústria farmacêutica, medicina, entre outras.

1.1. CONTEXTO DA DISSERTAÇÃO

Existem vários modelos que podem ser utilizados para prever uma série temporal, como os modelos de *Box-Jenkins* [Box, Jenkins 1970], o modelo *Random Walk* [Harvey 1993] e vários modelos de Redes Neurais [Haykin 2001]. O desempenho de cada modelo varia de acordo com o problema abordado. Com isso, selecionar o modelo mais adequado para prever uma dada série pode ser uma tarefa difícil, que depende de fatores como o ajuste dos parâmetros dos modelos candidatos e as características da série.

Podemos encontrar na literatura diversas abordagens que podem ser utilizadas na seleção de modelos de previsão. A abordagem mais direta é a realização de um torneio entre os modelos candidatos, utilizando os dados disponíveis da série e selecionando o modelo que obteve o melhor desempenho para uma determinada amostra de observações. Esta abordagem, que parece simples inicialmente, pode ser bastante custosa caso seja considerado um grande número de modelos ou de séries a serem previstas.

Na tentativa de minimizar esta dificuldade, têm sido utilizados algoritmos de aprendizado de máquina, que adquirem conhecimento de forma automática, utilizando-o na seleção dos modelos [Arinze 1994]. Nesta abordagem, um algoritmo de aprendizado relaciona as características das séries temporais (tamanho, tendência, entre outras) ao desempenho dos modelos de previsão. Existem várias abordagens de aprendizado de máquina conhecidas na literatura como as Redes Neurais [Haykin 2001] e os algoritmos baseados em instâncias [Aha et. al 1991]. Contudo, o desempenho de uma determinada técnica pode variar de acordo com os dados que são utilizados como entrada ou com os valores que são ajustados para seus parâmetros. Neste contexto, o uso de abordagens de meta-aprendizado é visto como uma solução motivadora para o problema de seleção de modelos, visto que estas abordagens utilizam o conhecimento sobre o desempenho dos modelos em problemas passados na melhoria da seleção dos modelos para o problema atual, fornecendo informações como rankings de modelos baseados em múltiplos critérios fornecidos pelo usuário.

1.2. OBJETIVOS

Neste trabalho, iremos utilizar uma técnica específica de meta-aprendizado no problema de seleção de modelos. A abordagem escolhida foi o Zoomed Ranking [Soares, Brazdil 2000] [Brazdil et al. 2003], originalmente proposto para a seleção de algoritmos para problemas de aprendizado, mais especificamente problemas de classificação, e adaptada ao problema de seleção de modelos de previsão. Esta abordagem utiliza um conjunto de exemplos de treinamento, onde cada exemplo armazena: (1) as características de uma dada série temporal e (2) as informações de desempenho (especificamente erro de precisão e tempo de execução) dos modelos de previsão. Por ser um algoritmo baseado em instâncias, esta abordagem possibilita ao usuário a seleção dos modelos com uma importância relativa sendo indicada para cada um dos critérios de desempenho. O Zoomed Ranking utiliza então as informações de desempenho em problemas anteriores na geração de rankings dos modelos para o problema atual

Em um trabalho anterior [Santos et. al 2004a] [Santos et. al 2004b], utilizamos com sucesso a abordagem Zoomed Ranking no problema de seleção e ordenação de modelos. No trabalho aqui proposto, iremos investigar com mais profundidade o uso da

técnica em problemas de outros domínios, mais especificamente nos domínios elétrico, financeiro e econômico.

Uma outra característica verificada na abordagem aqui investigada, típica de um algoritmo baseado em instâncias, é o uso de exemplos processados anteriormente pelos modelos na geração de rankings para o problema atual. Originalmente, o Zoomed Ranking fez uso da função de distância definida no algoritmo dos k-vizinhos mais próximos (K-NN) [Cover, Hart 1967] na seleção dos exemplos mais similares aos exemplos de entrada, utilizados na geração dos rankings. Neste contexto, um outro ponto a ser investigado no trabalho aqui proposto é a modificação da técnica original para utilizar uma Rede SOM em lugar da função de distância original. Esta adaptação visa diminuir a sensibilidade da técnica em relação a características irrelevantes dos exemplos de entrada.

1.3. TRABALHO REALIZADO

A metodologia utilizada neste trabalho para solucionar o problema proposto baseia-se em conceitos de meta-aprendizado, como a utilização de *meta-exemplos* que armazenam resultados de problemas solucionados no passado e a utilização de um *meta-aprendiz*, que utiliza as características dos dados de entrada para selecionar e ordenar modelos.

Os modelos candidatos à previsão ordenados pelo Zoomed Ranking em todos os estudos de caso, são modelos baseados em técnicas de Redes Neurais [Haykin 2001]. Como a previsão de uma série temporal é um processo de descoberta de comportamentos futuros a partir de informações passadas, uma rede neural apresenta-se como uma boa alternativa para utilização como modelo de previsão, dado que são capazes de inferir corretamente novas informações após o aprendizado dos dados apresentados.

Foram realizados 4 estudos de caso para a investigação da solução aqui proposta. A abordagem Zoomed Ranking foi utilizada em duas bases de séries, extraídas da competição organizada pela EUNITE [Rojas 2004], com séries do domínio elétrico, e do repositório *benchmark* da *Time Series data Library (TSDL)* [Hyndman 2006], contendo séries dos domínios financeiro e econômico. Verificamos que a abordagem Zoomed Ranking como proposta no trabalho anterior apresentou bons resultados nos dois domínios investigados. Os rankings sugeridos pelo Zoomed Ranking são

fortemente correlacionados com os rankings ideais gerados para os modelos. A adaptação realizada na técnica, proposta neste trabalho, apresentou resultados encorajadores para as séries de todos os domínios aplicados, motivando a uma maior investigação em trabalhos futuros.

1.4. CONTEÚDO DA DISSERTAÇÃO

Os demais capítulos desta dissertação estão estruturados da seguinte forma:

- **Capítulo 2 – Séries Temporais:** Este capítulo apresenta conceitos que servirão de base para a compreensão do que é uma série temporal e em quais aspectos as séries temporais serão exploradas neste trabalho. Serão abordados pontos como a previsão de séries temporais e a seleção de modelos, com um estudo comparativo entre algumas abordagens que podem ser utilizadas na seleção de modelos de previsão.
- **Capítulo 3 – Abordagens de Meta-Aprendizado:** Neste capítulo, apresentaremos diferentes abordagens de Meta-Aprendizado, que podem ser utilizadas na seleção de algoritmos de classificação para problemas de aprendizado e, conseqüentemente, adaptadas ao problema de seleção de modelos.
- **Capítulo 4 – Metodologia:** Apresentaremos neste capítulo a metodologia aplicada neste trabalho para a investigação da solução aqui proposta, incluindo a arquitetura utilizada e alguns aspectos de implementação adotados.
- **Capítulo 5 – Análise Experimental:** Neste capítulo serão apresentados detalhes experimentais e os resultados obtidos nos quatro estudos de caso realizados.
- **Capítulo 6 – Conclusão:** Este capítulo apresentará um resumo dos pontos mais importantes abordados neste trabalho, contribuições do presente trabalho e possíveis trabalhos futuros.

2. SÉRIES TEMPORAIS

Para uma melhor contextualização do problema de seleção de modelos de séries temporais, existem alguns pontos importantes a serem vistos, que serão apresentados nas próximas seções deste capítulo.

2.1. PREVISÃO DE SÉRIES TEMPORAIS

Existem muitos fenômenos no mundo real que variam com o decorrer do tempo. Muitos destes fenômenos podem ser registrados como, por exemplo, a variação da tensão elétrica medida em um dispositivo eletrônico durante um dia ou a temperatura de uma reação química durante um certo intervalo de tempo.

Uma série temporal é um conjunto de observações de um fenômeno ordenadas no tempo [Box, Jenkins 1970]. As séries temporais podem ser contínuas, quando são medidas sem interrupções no tempo, ou discretas quando são medidas em intervalos sucessivos de tempo, em geral, em instantes de tempo igualmente espaçados.

Existem vários objetivos envolvidos na análise de uma série temporal. Porém, os mais frequentes são [Tukey 1980]:

- a) Modelagem do fenômeno sob consideração;
- b) Obtenção de conclusões em termos estatísticos;
- c) Avaliação da adequação do modelo em termos de previsão.

Esta análise leva a um processo de indução, uma vez que, a partir de um conjunto de observações, características gerais são inferidas acerca do fenômeno observado.

Segundo [Dorffner 1996], dentre os diversos objetivos da análise de séries temporais, o maior deles é a previsão. Porém, a previsão normalmente não se constitui como o objetivo final desta análise, mas um meio de fornecer informações para uma conseqüente tomada de decisão, visando determinados objetivos [Morettin, Toloí 1987].

Como descrito na Equação 2.1, dada uma série definida por Z_1, Z_2, \dots, Z_t , a previsão desta série, representada por $\hat{Z}_t(h)$, é uma estimativa do valor da série no tempo $t+h$, ou seja, é a previsão h passos à frente a partir do instante de tempo t . O valor de h é chamado de horizonte de previsão.

$$\hat{Z}_t(h) = f(Z_{t-1}, Z_{t-2}, \dots, \theta) \quad (2.1)$$

Equação 2.1: Valor previsto de uma série retornado por um modelo de previsão

Os modelos de previsão de séries temporais são capazes de definir, com um certo grau de confiabilidade, os valores futuros de uma série a partir de informações passadas e de outras variáveis significantes para o problema. O valor previsto de uma série retornado por um modelo de previsão normalmente faz uso de uma função matemática f que se baseia nos valores passados da série e em parâmetros livres θ (Ver Equação 2.1). Os modelos utilizam parâmetros que refletem os relacionamentos entre os valores passados da série e o valor a ser estimado.

O uso de modelos de previsão é fundamental para diminuir os riscos na tomada de decisões, uma vez que a eficácia de uma decisão depende obviamente dos eventos que a sucedem [Montgomery et. al 1990]. Entre as diversas aplicações de previsão de séries temporais, podemos citar como exemplos, aplicações na área financeira [Oliveira 2000], na área de recursos hídricos [Valença 2001] [Valença 2005] e na área elétrica [Chen et. al 2001].

2.2. AVALIAÇÃO DE UM MODELO DE SÉRIES TEMPORAIS

Existem diversos fatores que podem ser considerados na avaliação de um modelo de previsão, como a precisão, a facilidade de interpretação e uso, a complexidade do modelo estimado, entre outros. A precisão tem sido usada como o critério principal para comparar modelos de séries temporais [Collopy, Armstrong 1992]. Esta métrica pode ser estimada com o uso de várias medidas de erro de previsão, cada uma apresentando suas vantagens e desvantagens.

$$SSE = \sum_{t=1}^N (Z_t - \hat{Z}_t)^2 \quad (2.2)$$

Equação 2.2: Soma dos erros quadráticos

De acordo com [Chatfield 2001], as medidas mais usadas são aquelas que avaliam os erros quadrados de previsão, como a soma dos erros quadráticos (SSE, *Sum of Squared Errors*) (Ver Equação 2.2) ou a média dos erros quadráticos (MSE, *Mean of*

Squared Errors) (Ver Equação 2.3), nas quais N é o número de pontos previstos da série, Z_t é o valor da série no instante de tempo t e Z_t^\wedge é a previsão da série no instante de tempo t .

$$MSE = \frac{1}{N} \sum_{t=1}^N (Z_t - Z_t^\wedge)^2 \quad (2.3)$$

Equação 2.3: Média dos erros quadráticos

$$MAE = \frac{1}{N} \sum_{t=1}^N |Z_t - Z_t^\wedge| \quad (2.4)$$

Equação 2.4: Média dos erros absolutos

As medidas de erro quadrado, apesar de serem muito utilizadas pela sua adequação à análise teórica, apresentam uma série de desvantagens, como sensibilidade à presença de valores aberrantes¹ e à escala da série; e dificuldades de interpretação e comparação entre séries diferentes. Desta forma, outras medidas de erro podem ser utilizadas na tentativa de corrigir estas deficiências, como, por exemplo, a média dos erros absolutos (MAE, *Mean Absolute Error*) (Ver Equação 2.4) e a mediana² dos erros absolutos (MdAE, *Median Absolute Error*) (Ver Equação 2.5), que têm a vantagem de ter a mesma escala dos dados da série.

$$MdAE = \text{median} |Z_t - Z_t^\wedge| \quad (2.5)$$

Equação 2.5: Mediana dos erros quadráticos

Uma outra maneira de controlar a escala dos dados da série é fazendo uso de erros percentuais, ou seja, calcular o erro como um percentual do valor atual [Armstrong, Collopy 1992]. Exemplos deste tipo de medida são a Média dos Erros Percentuais Absolutos (MAPE, *Mean Absolute Percentage Error*) e a Mediana dos Erros Percentuais Absolutos (MdAPE, *Median Absolute Percentage Error*). Existe

¹ Os valores aberrantes são aqueles que se apresentam fora da escala de valores observados na série.

² Se as observações de uma variável estão ordenadas por valor, a mediana representa o valor médio observado. Ou seja, a mediana corresponde ao $\frac{n+1}{2}$ ésimo valor, onde n é o número de valores observados.

ainda a medida *Percentage Better* (PB), que é dada pelo percentual de previsões para o qual um dado modelo sendo analisado é mais preciso que um modelo de referência.

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{Z_t - Z_t^{\wedge}}{Z_t} \right|$$

$$MdAPE = \text{median} \left| \frac{Z_t - Z_t^{\wedge}}{Z_t} \right|$$

2.3. SELEÇÃO DE MODELOS DE SÉRIES TEMPORAIS

Diversos modelos podem ser usados para prever uma dada série temporal. A seleção do melhor modelo deve ser feita com base nas necessidades do usuário e nas características das séries e dos modelos candidatos à previsão. De acordo com [Fildes 1989], um ganho substancial no desempenho de previsão pode ser alcançado através da escolha de um modelo adequado de previsão. Nesta seção, apresentaremos duas das principais abordagens de seleção de modelos de previsão de séries temporais, com suas vantagens e desvantagens.

2.3.1. ABORDAGEM POR TORNEIO

Uma das abordagens mais diretas utilizadas na seleção de modelos é a realização de um torneio entre os modelos candidatos. Esta abordagem consiste em estimar os parâmetros de todos os modelos candidatos com os dados da série, em seguida usar os modelos estimados para prever um determinado conjunto de pontos da série e, então, selecionar o modelo que obteve o melhor desempenho de acordo com alguma medida de avaliação pré-definida. Um torneio entre modelos é realizado em duas etapas: a definição da medida de avaliação usada para definir o melhor modelo e a definição da amostra de dados da série sob a qual esta medida será calculada.

Vimos na Seção 2.2 deste capítulo quais medidas podem ser utilizadas na realização da primeira etapa desta abordagem. Na segunda etapa, o critério de avaliação pode ser calculado com os mesmos dados usados na estimação dos parâmetros dos modelos candidatos. Esse tipo de avaliação é denominado comumente como avaliação

in-sample [Tashman 2000]. Segundo [Tashman 2000], a avaliação *in-sample* geralmente superestima o desempenho dos modelos de previsão. Outra desvantagem, segundo [Collopy et. al 1994] seria que a adequação de um modelo aplicado a dados históricos de uma série não é um indicador confiável de sua habilidade para previsão de valores futuros.

Uma outra forma de avaliar um modelo é calcular o critério de desempenho sob uma amostra separada de dados da série, usada como conjunto de validação. Desta forma, os dados são divididos em dois conjuntos: um conjunto utilizado para ajustar os parâmetros de cada modelo e um outro conjunto de validação utilizado na comparação do desempenho dos modelos anteriormente ajustados. O modelo selecionado será aquele que obtiver o melhor desempenho para o conjunto de validação. Este tipo de avaliação é denominado comumente de avaliação *ex-ante*, *out-of-sample* ou *holdout* [Collopy et. al 1994] [Tashman 2000]. A avaliação *out-of-sample*, anteriormente utilizada em [Fildes 1989], apresentou resultados indicando que este critério pode não ser bem sucedido para séries com comportamento muito instável. Já em [Ashley 2003], resultados experimentais mostraram que a avaliação *out-of-sample* não é confiável a menos que a amostra de dados tenha um tamanho razoável.

Apesar de ser uma abordagem simples e direta, a abordagem por torneio apresenta a limitação de avaliar os modelos de forma dependente de uma amostra de dados e de uma medida de desempenho, que são fatores normalmente sensíveis a instabilidades nas séries. Existe ainda o fato de que resultados obtidos para uma determinada série nem sempre resultam em informações relevantes para selecionar modelos para outras séries.

Uma alternativa a estas limitações é fazer uso de *competições entre modelos* para grandes conjuntos de séries temporais. A idéia é tentar encontrar regularidades nos resultados que possam ser generalizadas para outras séries.

Uma competição de previsão de séries temporais extremamente conhecida é a M-Competition [Makridakis et. al 1982] e as suas versões subseqüentes, a M2-competition [Makridakis et. al 1993] e a M3-Competition [Makridakis, Hinbon 2000]. A idéia principal dos autores foi a realização de uma competição com o maior número de séries e modelos possível.

Na M-Competition, foi utilizado um conjunto de 1001 séries temporais. Os participantes utilizavam modelos de seu interesse na previsão destas séries para

posterior avaliação pelos autores da competição em um conjunto de validação não disponível inicialmente. O conjunto de séries disponibilizado foi submetido a 24 modelos diferentes. As principais conclusões obtidas foram que o desempenho dos modelos está diretamente associado ao horizonte de previsão utilizado; pode haver uma variação significativa no desempenho dos modelos de acordo com a medida de erro escolhida; o uso de modelos complexos não implica na obtenção de previsões mais precisas que as geradas por modelos mais simples; combinar diferentes modelos pode levar à obtenção de um melhor desempenho comparando com a aplicação dos modelos individuais. As versões posteriores da M-Competition incluíram novas séries e novos modelos na competição, entretanto as conclusões obtidas foram as mesmas.

Uma outra referência nesta área é a competição organizada pela EUNITE [Rojas 2004] para a previsão de cargas elétricas. Ao todo, participaram 56 competidores de 21 países diferentes. Nesta competição, dadas as temperaturas e as cargas elétricas nos anos de 1997 e 1998, os competidores forneciam as previsões das cargas elétricas para o mês de Janeiro de 1999. Os dados da competição foram fornecidos pela Corporação Elétrica Oriental Eslovaca (*Eastern Slovakian Electricity Corporation*) que assume grande interesse em aplicações com tecnologia inteligente e adaptativa em previsões de cargas elétricas.

Os resultados obtidos comprovaram a habilidade das técnicas utilizadas (entre elas Máquina de Vetores de Suporte e Redes Neurais) nas previsões com precisão razoável.

Uma limitação de várias competições de séries temporais é que as suas conclusões, na maioria dos casos, não são expressas de forma precisa. Na M-Competition, por exemplo, os autores chegaram à conclusão de que modelos simples trabalham tão bem quanto modelos complexos. Porém, essa conclusão não deixa claro quais modelos são mais adequados a quais tipos de séries e em que condições. Em [Armstrong 2001], o autor ressalta que mesmo para um conjunto pequeno de características, o número de condições que podem ser testadas pode ser muito grande. A dificuldade nesse caso é que um especialista humano não seria eficiente em analisar inúmeras hipóteses que podem ser testadas para explicar os resultados de uma competição. Nesse caso, seria necessário o uso de técnicas automatizadas para realizar a aquisição de conhecimento.

2.3.2. APRENDIZADO DE MÁQUINA

A abordagem por torneio é uma das abordagens mais diretas para seleção de modelos, porém torna-se custosa quando um grande número de modelos ou de séries a prever é considerado. Essa abordagem ainda apresenta a desvantagem de não fornecer, de forma clara, conclusões acerca dos seus resultados.

Desta forma, as técnicas de aprendizado de máquina podem ser utilizadas na tentativa de superar essas deficiências. Esta abordagem proporciona o aprendizado automático das características de um conjunto de dados, levando ao aumento do desempenho. Existem diversas vantagens associadas ao uso desta abordagem, entre elas destacamos a descoberta de peculiaridades em grandes massas de dados, aumentando os níveis de automação, economizando recursos e aumentando o desempenho.

Entre os diversos algoritmos de aprendizado de máquina existentes estão as Redes Neurais [Haykin 2001] e os algoritmos baseados em instâncias [Quinlan 1993]. Como não existe um único algoritmo que seja o melhor para todos os problemas possíveis [Aha 1992], um importante ponto a ser analisado é como escolher o algoritmo mais adequado para cada problema.

Desta forma, a utilização de algoritmos de aprendizado de máquina na seleção de modelos de previsão de séries temporais, trata-se de uma solução mais ampla comparada às abordagens tradicionais. Esta solução além de permitir a seleção do melhor modelo para prever uma dada série pode fornecer uma resposta ainda mais informativa: um ranking dos modelos candidatos para prever a série em questão.

O uso de algoritmos de aprendizado de máquina para seleção de modelos foi proposto originalmente por [Arlinze 1994]. Neste trabalho, foi utilizado o algoritmo *ID3* na indução de árvores de decisão para seleção entre seis modelos candidatos. Foram utilizadas para previsão 67 séries de domínios econômicos.

Neste contexto, cada exemplo de treinamento consiste de uma série temporal descrita por um conjunto de características, associadas a um atributo classe, que é o melhor modelo de previsão para a série. O valor do atributo classe para uma dada série normalmente é definido de maneira experimental, realizando um torneio entre os modelos candidatos.

A partir de um conjunto de exemplos de treinamento, um algoritmo de aprendizado de máquina é capaz de associar características de novas séries temporais

fornecidas como entrada aos melhores modelos de previsão. O treinamento de um algoritmo de aprendizado de máquina para seleção de modelos é realizado em cinco etapas:

- (1) definir os modelos de previsão candidatos;
- (2) definir as características relevantes para descrever as séries temporais;
- (3) para cada série realizar um torneio entre os modelos candidatos para definir o atributo classe;
- (4) aplicar o algoritmo de aprendizado usando o conjunto de exemplos de treinamento como entrada;
- (5) testar o modelo de aprendizado gerado com novas séries.

O uso de algoritmos de aprendizado de máquina é bastante relacionado com as competições da abordagem por torneio, apresentada na seção 2.3.1.1. A geração do conjunto de exemplos de treinamento é feita através de uma competição entre os modelos candidatos para um conjunto significativo de séries. A diferença entre este processo e uma competição tradicional é que neste último o processo de aquisição de conhecimento é realizado de forma manual, enquanto que o uso de algoritmos de aprendizado de máquina se apresenta como uma ferramenta para a aquisição automática.

Por outro lado, esta abordagem apresenta também algumas desvantagens, como a dificuldade no uso de características subjetivas e informações contextuais dos problemas. A vantagem do uso de características objetivas é que a seleção de modelos pode ser feita de forma inteiramente automática. As características subjetivas demandam maior tempo para serem identificadas. Desta forma, a construção de um conjunto de exemplos de treinamento contendo essas características poderia se tornar inviável, uma vez que cada série teria que ser avaliada manualmente.

2.3.2.1. Meta-Aprendizado

Existem várias abordagens de aprendizado de máquina conhecidas na literatura como as Redes Neurais [Haykin 2001] e os algoritmos baseados em instâncias [Aha et. al 1991]. Porém, não existe uma única abordagem que seja a melhor para qualquer tipo de problema sendo investigado. O desempenho de uma determinada técnica pode variar

de acordo com os dados que são utilizados como entrada, ou ainda de acordo com os valores que são ajustados para seus parâmetros.

Neste contexto, introduzimos o conceito de meta-aprendizado que seria o estudo de como sistemas de aprendizado podem aumentar sua eficiência através de experiência [Vilalta, Drissi 2002]. Em [Giraud-Carrier 2004], os autores definem meta-aprendizado como o processo de exploração de conhecimento sobre o aprendizado que possibilita a compreensão e a melhoria do desempenho de algoritmos de aprendizado. O termo meta-aprendizado pode ser associado ainda a métodos de aprendizado capazes de (1) avaliar e comparar seus resultados, (2) medir o benefício de aprendizados anteriores em aprendizados subseqüentes, (3) comparar técnicas de aprendizado para selecionar a melhor entre elas [Schmidhuber et. al 1996].

Alguns trabalhos como [Prudêncio, Ludermir 2003], [Prudêncio, Ludermir 2004], [Santos et. al 2004a] e [Santos et. al 2004b] utilizaram uma abordagem de meta-aprendizado na solução do problema de seleção de modelos.

Em [Prudêncio, Ludermir 2003], foi utilizada a abordagem NOEMON no problema de seleção. Neste trabalho, cada nova série a ser prevista tem suas características extraídas e posteriormente utilizadas na geração do ranking dos modelos candidatos. Esta extração é realizada de acordo com o tipo da série temporal sendo analisada, uma vez que cada tipo é representado por um conjunto diferente de características. Cada exemplo de treinamento associa uma série temporal (representada por um conjunto de características) ao desempenho dos modelos candidatos durante a previsão da série. Desta forma, o ranking é gerado de acordo com as características da série de entrada e com uma função de erro que se deseja minimizar.

Em [Prudêncio, Ludermir 2004], os autores investigaram o uso de duas abordagens de meta-aprendizado na seleção de modelos. Inicialmente foi utilizada uma técnica de aprendizado de máquina para selecionar o melhor entre dois modelos na previsão de séries estacionárias. Em seguida, foi utilizada novamente a abordagem NOEMON [Kalousis, Theoharis 1999] na geração do ranking para três modelos utilizados na previsão das séries da M3-Competition [Makridakis, Hibon 2000].

No trabalho desenvolvido por [Santos et. al 2004a] [Santos et. al 2004b], foi investigado o uso de uma abordagem de meta-aprendizado na seleção de três modelos utilizados na previsão das séries da M3-Competition. Os autores utilizaram a abordagem Zoomed Ranking [Brazdil et al. 2003] [Soares, Brazdil 2000], que utiliza as

informações de desempenho na previsão das séries para a geração de rankings dos modelos, segundo critérios definidos pelo usuário.

2.4. CONSIDERAÇÕES FINAIS

Neste capítulo, apresentamos inicialmente os principais conceitos envolvidos no problema de seleção de modelos. Vimos que vários modelos podem ser utilizados na previsão de uma dada série temporal. Com isso, a escolha do melhor modelo deve levar em conta alguns critérios, como as necessidades do usuário, as características associadas às séries (tipo, domínio, entre outras) e aos modelos candidatos à previsão (informações a serem previstas, quantidade de parâmetros, valores estimados para os parâmetros, entre outras).

Vimos também algumas abordagens que podem ser utilizadas na seleção de algoritmos de aprendizado, que conseqüentemente podem ser utilizadas na seleção de modelos de previsão, entre elas a abordagem de meta-aprendizado.

Sabemos que a eficiência no processo de seleção de modelos de previsão pode ser melhorada eficientemente com o uso abordagens de meta-aprendizado [Giraud-Carrier 2004]. Com isso, iremos focar em abordagens de meta-aprendizado, mais especificamente em uma abordagem que gera um ranking dos modelos candidatos à previsão de um conjunto de séries. A abordagem utilizada neste trabalho, que será explicada em detalhes na Seção 3.4, gera rankings dos modelos candidatos a partir dos resultados de desempenho dos modelos em problemas anteriores do mesmo domínio. Esta abordagem utiliza ainda mais de um critério de desempenho, permitindo ao usuário a obtenção de um ranking dos modelos de acordo com as suas necessidades.

No Capítulo 3 serão apresentadas algumas técnicas meta-aprendizado que podem ser utilizadas na seleção de modelos de previsão de séries temporais.

3. ABORDAGENS DE META-APRENDIZADO

As abordagens que serão apresentadas neste capítulo têm três pontos em comum: um conjunto de *algoritmos-base*, que são os algoritmos candidatos a resolver um dado problema de aprendizado; um *meta-algoritmo* (ou meta-aprendiz), que é o responsável por gerar conhecimento de forma automática para auxiliar a seleção dos algoritmos-base; e um conjunto de *meta-exemplos*, que contém informações sobre o uso dos algoritmos-base no problema abordado ou em problemas de outro domínio.

Um aspecto importante a ser considerado ao se trabalhar com abordagens de meta-aprendizado é a caracterização dos dados, ou seja, a definição dos meta-atributos. Os meta-atributos são as características usadas para descrever as séries temporais sendo previstas. Normalmente, essas características são medidas descritoras dos dados de treinamento, que podem ser medidas simples, como número de exemplos, número de atributos e número de classes; medidas estatísticas, como média absoluta das correlações entre atributos; e medidas da teoria de informação, como entropia da classe e média da entropia dos atributos. Em [Engels, Theusinger 1998], os autores utilizaram 51 características de problemas de classificação, calculadas de forma automática com o *Data Characterization Tool*.

O conjunto de meta-exemplos serve como fonte de aquisição de conhecimento para o meta-algoritmo, ou seja, o meta-algoritmo utiliza o conjunto de meta-exemplos para gerar meta-conhecimento relacionando os valores dos meta-atributos com o desempenho dos algoritmos-base.

Ao utilizarmos uma abordagem de meta-aprendizado na seleção de modelos de previsão de séries temporais, podemos fazer uso das mesmas técnicas que têm sido utilizadas amplamente na seleção de algoritmos para problemas de classificação. Neste contexto, os meta-algoritmos passam a ser os modelos de previsão e os meta-atributos são as características que descrevem as séries temporais.

Neste capítulo iremos falar apenas sobre abordagens de meta-aprendizado utilizadas na seleção de algoritmos de aprendizado: a abordagem simples de meta-

aprendizado [Aha 1992], a abordagem baseada em instâncias [Aha et. al 1991], a abordagem NOEMON [Kalousis, Theoharis 1999] e a abordagem Zoomed Ranking [Soares, Brazdil 2000] [Brazdil et al. 2003].

3.1. ABORDAGEM SIMPLES

Uma das abordagens mais simples de meta-aprendizado é usar um algoritmo de aprendizado para selecionar apenas um algoritmo-base dentre o conjunto de candidatos. O meta-algoritmo recebe um conjunto de meta-exemplos como entrada, no qual cada meta-exemplo consiste de um conjunto de características (meta-atributos) associado a um atributo de classe representando o melhor algoritmo-base entre os candidatos. O atributo de classe normalmente é definido de forma experimental, representando o algoritmo que obteve os melhores resultados para o problema em questão.

Desta forma, o meta-algoritmo é capaz de associar os meta-atributos das séries recebidos como entrada aos atributos de classe representando o algoritmo mais adequado ao problema.

Esse tipo de abordagem para seleção de algoritmos foi utilizado inicialmente em [Aha 1992]. Neste trabalho, o algoritmo CN2 [Clark, Niblett 1989] foi usado para induzir regras utilizadas na seleção do melhor algoritmo entre três algoritmos-base. Os meta-exemplos de treinamento foram gerados a partir de dados artificiais.

No trabalho realizado em [Merz 1996], o autor propõe uma abordagem que prevê a classe de um exemplo de entrada através de sua localização no espaço de exemplos e, em seguida, escolhe o melhor algoritmo de aprendizado de acordo com a região selecionada. As regiões do espaço de exemplos são determinadas de acordo com os padrões de previsão dos algoritmos. Os algoritmos são escolhidos de acordo com suas informações de desempenho passadas naquela região. Os dados utilizados neste trabalho foram extraídos do *UCI Repository (Glass, Iris, Breast Cancer Wisconsin and Liver Disorders)* [Newman et. al 1998]. Esta abordagem foi comparada com duas outras abordagens de seleção de algoritmos de aprendizado: a SAM (*Select All Majority*), que considera todas as previsões de cada algoritmo com igual importância e retorna a mais frequente; e a CVM (*Cross-Validation Majority*), que retorna a previsão do algoritmo que obteve o maior valor para a precisão em um processo de validação-cruzada.

3.2. ALGORITMOS BASEADOS EM INSTÂNCIAS

A abordagem simples define o melhor algoritmo em termos da precisão obtida para cada problema, geralmente avaliada por um processo de validação cruzada. Embora a precisão seja um critério importante para definir quão bom é um algoritmo de aprendizado, existem outros critérios como o tempo de execução, a facilidade de interpretação, dentre outros [Lindner, Studer 1999]. A relação entre cada um desses critérios varia de acordo com as necessidades do usuário e com o problema avaliado. Desta forma, pode ser dada uma maior importância a um determinado critério em detrimento de um outro.

O uso de algoritmos de aprendizado baseados em instâncias [Aha et al. 1991] como meta-aprendizes possibilita a consideração de critérios flexíveis durante a seleção. Ao fazer uso desta abordagem, a solução aplicada a cada problema de aprendizado pode ser armazenada em uma instância. Com isso, dado um novo problema, o meta-aprendiz recupera de uma base de instâncias os problemas mais similares àquele fornecido como entrada.

Desta forma, o meta-aprendiz define o algoritmo que obteve os melhores resultados nos problemas similares já processados, conforme as necessidades do usuário. Por exemplo, se o usuário desejar um algoritmo com maior precisão, o meta-aprendiz retorna o algoritmo candidato que obteve o menor erro na solução dos problemas similares.

No trabalho desenvolvido em [Punronen, Terziyan 2000], os autores desenvolveram um *framework* para avaliação de similaridade utilizado na mineração dos dados de um conjunto de algoritmos de classificação. Nesta avaliação, foi utilizado um conjunto de treinamento com predicados, que definem relacionamentos entre três conjuntos: o conjunto de instâncias, o conjunto de classes e o conjunto de classificadores. A similaridade entre as instâncias foi utilizada na identificação dos seus vizinhos mais próximos. A similaridade entre as classes foi necessária no aprendizado dos classificadores. A similaridade entre classificadores foi utilizada para sua avaliação no momento em que são integrados como exemplos de treinamento da abordagem de voto ponderado (*Weighted Voting Approach*) que avalia os pesos dos classificadores. Os resultados dos classificadores são armazenados em uma instância da matriz de

predicados. Os resultados obtidos neste trabalho foram obtidos em um nível teórico, sendo necessárias pesquisas com dados reais para a validação da abordagem proposta.

3.3. ABORDAGEM NOEMON

Uma das principais limitações que observamos na abordagem simples, é que as soluções fornecidas são normalmente pouco informativas. Na verdade, a maioria das abordagens de meta-aprendizado utilizadas na seleção de algoritmos sugere apenas um algoritmo ou um grupo de algoritmos que realizam bem um determinado problema [Brazdil et. al 1994] [Todorovski, Dzeroski 1999]. Na maioria dos problemas, a quantidade de algoritmos a ser utilizada pelo usuário não é conhecida a priori.

Neste contexto, a geração de um ranking de algoritmos é vista como uma alternativa à solução destes problemas, na medida que mostra a relação entre os algoritmos candidatos por ordem de melhor indicação. Isto possibilita a utilização de um algoritmo no qual se tenha uma maior experiência caso o desempenho obtido pelo mesmo seja ligeiramente menor que aquele obtido pelo algoritmo no topo do ranking. Além disso, é possível fazer uso de mais de um algoritmo caso recursos suficientes (tempo, poder de processamento, entre outros) estejam disponíveis.

Em [Kalousis, Theoharis 1999], os autores fazem uso de uma abordagem de geração de ranking de algoritmos. O sistema NOEMON, apresentado neste trabalho, é capaz de retornar um ranking de algoritmos usando múltiplos algoritmos de aprendizado. Este trabalho foi originalmente desenvolvido para a seleção de algoritmos para problemas de classificação. A geração do ranking é feita através da criação de um espaço de meta-aprendizado composto por n^2 meta-problemas. Cada meta-problema é então associado a um par específico de algoritmos-base. Os meta-exemplos desta abordagem também são descritos pelas características de um problema associadas a um atributo de classe.

Para cada par (X, Y) de algoritmos-base, um problema é classificado em uma das classes 'X', 'Y' ou 'igual' de acordo com o desempenho obtido por estes algoritmos para o problema. Após a construção dos conjuntos de meta-exemplos, NOEMON aplica um algoritmo de aprendizado para cada meta-problema, gerando assim n^2 meta-aprendizes. Com isso, a cada novo problema fornecido como entrada, o sistema

NOEMON gera um ranking de algoritmos combinando as respostas fornecidas pelos meta-aprendizes para o problema.

O sistema NOEMON apresenta a vantagem de ser flexível por possibilitar a associação de diferentes algoritmos de aprendizado a cada meta-problema. Além disso, o sistema fornece um mecanismo que possibilita a definição de diferentes meta-atributos para selecionar cada par de algoritmos-base. Por outro lado, a principal desvantagem desta abordagem é a necessidade de se construir um conjunto de meta-aprendizes na $O(n^2)$. Isso faz com que o tempo de construção dos meta-aprendizes seja decisivo, dependendo do número de algoritmos-base considerados.

3.4. ABORDAGEM ZOOMED RANKING

Uma das desvantagens das abordagens que geram rankings de algoritmos é que as informações de desempenho utilizadas levam em conta um único critério. Em [Brazdil, Soares 2000], os autores utilizaram uma abordagem de meta-aprendizado na geração do ranking de algoritmos a partir unicamente da precisão dos algoritmos. Uma outra desvantagem observada é que os rankings são gerados por estas abordagens levam em conta todas as informações de desempenho disponíveis. Algumas dessas informações podem ser irrelevantes para o problema de entrada, levando à geração de rankings imprecisos.

Em [Soares, Brazdil 2000], os autores propõem a abordagem Zoomed Ranking, que fornece um ranking dos algoritmos candidatos à solução de um determinado problema, baseada em informações de desempenho dos algoritmos. Em uma versão posterior [Brazdil et. al 2003], os autores dão continuidade a este trabalho com algumas modificações no algoritmo e na avaliação dos resultados. Por ser um algoritmo baseado em instâncias, o Zoomed Ranking possibilita ao usuário a seleção dos algoritmos baseada em mais de um critério de desempenho, com uma importância relativa dada a cada um dos critérios. Outra característica de um algoritmo baseado em instâncias utilizada por esta abordagem é o uso de exemplos processados anteriormente pelos algoritmos na geração de rankings para o problema atual.

O Zoomed Ranking opera em duas fases distintas para a geração dos rankings, que serão apresentadas em detalhes nas próximas seções deste capítulo.

3.4.1. SELEÇÃO DOS EXEMPLOS MAIS SIMILARES

Na primeira fase desta abordagem, chamada de *zooming*, são selecionados os problemas mais similares ao problema dado como entrada na base de instâncias. Esta fase é assim chamada, pois dado um espaço de problemas processados anteriormente, foca-se apenas na vizinhança do problema de entrada.

A similaridade entre um problema processado anteriormente e o problema de entrada é medida através do conjunto de seus meta-atributos descritores. A distância entre os problemas é medida através da métrica *Norma Sem Peso L_1* [Atkeson et. al 1997], descrita segundo a Equação 3.1 abaixo:

$$dist(v_{x,p_i}, v_{x,p_j}) = \sum_x \frac{|v_{x,p_i} - v_{x,p_j}|}{\max_{k \neq i} (v_{x,p_k}) - \min_{k \neq i} (v_{x,p_k})} \quad (3.1)$$

Equação 3.1: Métrica da *Norma Sem Peso L_1*

Nesta equação, p_i e p_j representam os problemas e v_{x,p_i} representa o valor do meta-atributo x no problema p_i .

A medida *dist* é dividida pelo domínio de valores possíveis para que o resultado seja normalizado. Esta função de distância é utilizada como parte do algoritmo dos k -vizinhos mais próximos (K-NN) [Cover, Hart 1967], cuja idéia básica é selecionar os k casos mais próximos a um caso de entrada, dada alguma função de distância [Mitchell 1997].

3.4.2. GERAÇÃO DO RANKING BASEADA NA PRECISÃO E NO TEMPO

Após a seleção dos problemas mais similares na fase anterior, o ranking dos algoritmos candidatos é gerado a partir das informações de desempenho obtidas pelos algoritmos nos problemas selecionados. Esta abordagem assume que o desempenho de um algoritmo quando aplicado a um problema específico será semelhante quando aplicado a problemas similares.

O ranking dos algoritmos candidatos é gerado segundo a métrica *ARR (Adjusted Ratio of Ratios)*, que agrega as informações de precisão e de tempo total de execução dos algoritmos. O *ARR* é definido segundo a Equação 3.2 abaixo:

$$ARR_{a_p, a_q}^{p_i} = \frac{SR_{a_p}^{p_i}}{SR_{a_q}^{p_i}} \cdot \frac{1}{1 + AccD * \log\left(\frac{T_{a_p}^{p_i}}{T_{a_q}^{p_i}}\right)} \quad (3.2)$$

Equação 3.2: Métrica ARR (Ajusted Ratio of Ratios)

Nesta equação, $SR_{a_p}^{p_i}$ e $T_{a_p}^{p_i}$ representam a precisão (taxa de sucesso) e o tempo de execução do algoritmo a_p no problema p_i , respectivamente, e $AccD$ é um parâmetro definido pelo usuário que representa a importância relativa entre a precisão e o tempo de execução.

O valor do parâmetro $AccD$ indica quanto da precisão o usuário está disposto a trocar por 10 vezes mais velocidade do algoritmo. O valor 1 é somado ao denominador da métrica ARR para obtermos valores em torno de 1, como acontece normalmente com a precisão.

A métrica ARR pode ser vista como a razão entre o benefício e o custo, utilizada para calcular a qualidade do algoritmo candidato como um todo. Desta forma, a razão entre as taxas de sucesso $SR_{a_p}^{p_i} / SR_{a_q}^{p_i}$ pode ser vista como uma medida de vantagem do algoritmo a_p sobre o algoritmo a_q , ou seja, como um benefício. Já a razão entre os tempos $T_{a_p}^{p_i} / T_{a_q}^{p_i}$ pode ser vista como uma medida de desvantagem do algoritmo a_p sobre o algoritmo a_q , ou seja, como um custo.

Um outro aspecto que deve ser observado em relação à razão entre os tempos é que ela está inserida em um domínio muito mais amplo de valores possíveis do que a razão entre as taxas de precisão. Com isso, se a razão simples entre os tempos fosse utilizada, esta dominaria a medida ARR . Desta forma, o efeito deste domínio pode ser minimizado com o uso do $\log(T_{a_p}^{p_i} / T_{a_q}^{p_i})$, que fornece a ordem de magnitude desta razão.

O ranking dos algoritmos candidatos é gerado a partir das informações da métrica ARR de forma agregada, ou seja, o valor do ARR é calculado para cada algoritmo candidato e o ranking é gerado diretamente destes valores. Isto pode ser feito da seguinte forma:

- Primeiro, calcula-se a média geométrica percorrendo todas os problemas selecionados;

- Em seguida, calcula-se a média aritmética percorrendo todos os algoritmos a_q diferentes de a_p , onde a_q é o algoritmo cuja posição no ranking está sendo calculada. Observe este processo na Equação 3.3 abaixo:

$$ARR_{a_p} = \frac{\sum_{a_q} \sqrt[n]{\prod_{p_i} ARR_{a_p, a_q}^{p_i}}}{m} \quad (3.3)$$

Equação 3.3: Métrica ARR para o algoritmo p

Nesta equação, n representa o número de problemas e m representa o número de algoritmos.

3.5. CONSIDERAÇÕES FINAIS

Neste capítulo, vimos algumas abordagens de meta-aprendizado propostas para a seleção de algoritmos de aprendizado e, que podem ser adaptadas para a seleção de modelos de previsão de séries temporais. Estas abordagens utilizam o conhecimento sobre o desempenho dos algoritmos em problemas passados na melhoria da seleção dos algoritmos no problema atual. Apresentamos desde técnicas mais diretas, que selecionam apenas um algoritmo dentre o conjunto de candidatos, passando pela seleção com múltiplos critérios, até técnicas mais elaboradas, que geram rankings dos algoritmos candidatos.

Vimos entre as abordagens apresentadas o algoritmo baseado em instâncias, que seleciona o melhor algoritmo para uma dada série levando em conta múltiplos critérios de desempenho. Esta abordagem seleciona na base de instâncias os exemplos anteriormente processados pelos algoritmos, que são mais similares ao exemplo de entrada, utilizando seus resultados de desempenho na indicação do melhor algoritmo.

Uma outra abordagem apresentada foi o Zoomed Ranking, que gera rankings dos algoritmos candidatos utilizando mais de um critério de desempenho. Os rankings gerados por esta abordagem levam em conta a precisão e o tempo de execução dos algoritmos, de acordo com a importância relativa definida pelo usuário. Por ser um algoritmo baseado em instâncias, esta abordagem leva em conta ainda as características da série de entrada na seleção dos exemplos mais similares. Os resultados de

desempenho dos algoritmos armazenados nestes exemplos são utilizados então na geração do ranking dos algoritmos candidatos.

A abordagem Zoomed Ranking foi utilizada com sucesso anteriormente na seleção de modelos de previsão de séries temporais [Santos et. al 2004a] [Santos et. al 2004b]. Neste trabalho, iremos utilizar esta abordagem na seleção de modelos para novos problemas de previsão, como veremos nos próximos capítulos. Outro aspecto que será investigado é a modificação da técnica original na tentativa de reduzir sua sensibilidade a características irrelevantes durante a fase de seleção dos exemplos mais similares.

No Capítulo 4, iremos apresentar a arquitetura utilizada neste trabalho de acordo com o problema a ser investigado, incluindo aspectos de implementação que foram considerados.

4. METODOLOGIA

No capítulo 3, apresentamos algumas técnicas de meta-aprendizado, originalmente propostas para a seleção de algoritmos de aprendizado para problemas de classificação. No entanto, técnicas de meta-aprendizado podem ser extrapoladas para a resolução de problemas de seleção em outros domínios onde diferentes algoritmos competem entre si. Dentre esses domínios podemos citar a seleção de algoritmos para problemas de planejamento [Tsoumakas et. al 2004] e a seleção de algoritmos de agrupamento [Mascaat 2006].

Neste contexto, em [Prudêncio, Ludermir 2002, 2003, 2004] os autores propõem o uso de técnicas de meta-aprendizado para a seleção de modelos de previsão de séries temporais. Dentre as abordagens de meta-aprendizado já aplicadas no problema de previsão citamos a abordagem simples [Prudêncio, Ludermir 2002] e a abordagem NOEMON [Prudêncio, Ludermir 2003]. No entanto, estas técnicas não foram investigadas com mais profundidade pelos autores.

Neste trabalho, investigamos o uso específico da técnica de Zoomed-Ranking [Soares, Brazdil 2000] [Brazdil et. al 2003] para seleção de modelos de previsão. Como vimos no capítulo anterior, o Zoomed-Ranking é uma abordagem de meta-aprendizado capaz de gerar rankings de algoritmos com base em mais de um critério de desempenho. Essa característica é também desejável para o problema de previsão de séries temporais.

Além da adaptação da técnica Zoomed-Ranking para a seleção de modelos, proposta em um trabalho anterior [Santos et. al 2004a] [Santos et. al 2004b], realizamos modificações na técnica original visando diminuir sua sensibilidade em relação a meta-atributos irrelevantes.

Comentário: Explicar com mais detalhes, incluindo os resultados obtidos.

4.1. ARQUITETURA GERAL

A solução investigada neste trabalho segue a arquitetura apresentada na Figura 4.1, que foi adaptada a partir da arquitetura proposta em [Prudêncio 2004] para sua adequação ao problema aqui proposto. Esta arquitetura faz uso de alguns conceitos de Meta-Aprendizado, citados no capítulo anterior, como a utilização de *meta-exemplos* que armazenam resultados de problemas processados anteriormente e a utilização de um

meta-aprendiz, que utiliza as características dos dados de entrada na geração dos resultados.

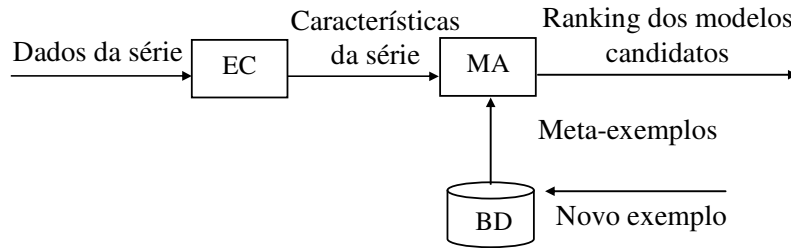


Figura 4.1: Arquitetura geral da solução proposta

Nesta arquitetura, o módulo BD (Base de Dados) contém os meta-exemplos processados pelos modelos candidatos. Cada meta-exemplo associa uma série temporal (representada por um conjunto de meta-atributos) ao desempenho dos modelos candidatos durante a previsão da série. O conjunto dos meta-atributos é obtido pelo módulo EC (Extrator de Características), que extrai as características de cada série recebida como entrada, fornecendo-as como entrada ao módulo MA (Meta-Aprendiz). Formalmente, cada meta-exemplo e_i é descrito por:

$$e_i = (c_1, \dots, c_j, d_i^1, \dots, d_i^m) \quad (4.1)$$

Equação 4.1: Descrição formal de um meta-exemplo

Cada termo c_j representa o valor de uma característica da série, que pode ser uma característica descritiva (como o tamanho da série) ou uma característica contextual (como o domínio da série). Essas características, às quais chamamos de meta-atributos das séries, são extraídas a partir dos dados originais da série.

O vetor representado pelo termo d_i^m corresponde aos resultados de desempenho do modelo m na previsão da i -ésima série. Este vetor pode conter informações como o erro de previsão obtido pelo modelo, o tempo de execução do modelo, a posição no ranking obtida pelo modelo, entre outras.

O módulo MA representa o meta-aprendiz, cujo conhecimento adquirido de forma automática é utilizado na seleção dos modelos candidatos. Este conhecimento pode ser refinado à medida que novos exemplos são inseridos na base. O módulo MA

utiliza as características da série recebidas como entrada para recuperar os meta-exemplos mais similares à série, contidos no módulo BD. As informações de desempenho contidas nos meta-exemplos selecionados são utilizadas então pelo módulo MA na geração do ranking dos modelos candidatos à previsão da série recebida como entrada.

Comentário: de entrada

4.2. ASPECTOS DE IMPLEMENTAÇÃO

Durante a implementação da abordagem proposta, surgiu a necessidade de avaliarmos alguns fatores relacionados às séries consideradas para previsão, aos modelos candidatos a serem utilizados, à abordagem de Meta-Aprendizado a ser utilizada bem como aos rankings gerados por esta abordagem. Estes fatores serão descritos nas próximas sessões deste capítulo.

4.2.1. SÉRIES TEMPORAIS

O conjunto de exemplos considerados para o treinamento da abordagem proposta foi construído levando-se em conta quatro fatores: a coleta das séries temporais, o vetor de desempenho utilizado, o cálculo das características das séries e o tipo de cada série.

A coleta das séries temporais utilizadas para previsão foi feita em dois repositórios disponíveis publicamente. O primeiro foi o *Time Series Data Library* (TSDL) [Hyndman 2006], que contém séries de vários domínios diferentes, desde o domínio financeiro até o domínio de saúde, usadas como *benchmark* na literatura de previsão de séries temporais. Outra fonte de coleta utilizada foi uma base disponível na competição de previsão de séries temporais organizada pela EUNITE [Rojas 2004], que disponibiliza séries do domínio elétrico para seus participantes.

Um outro fator considerado é o tipo da série a ser prevista. Após a coleta das séries temporais para a formação de cada base de dados, algumas destas séries podem ser originalmente não-estacionárias³.

³ Séries estacionárias são aquelas que se desenvolvem aleatoriamente no tempo ao redor de uma média constante, refletindo alguma forma de equilíbrio estável [Morettin, Toloí 1987].

A maioria das séries encontradas na prática apresenta alguma forma de não-estacionariedade. Como a maioria dos procedimentos de análise de séries temporais supõe que estas sejam estacionárias, é necessário transformar os dados originais, caso esta suposição não seja satisfeita [Morettin, Toloí 1987]. Desta forma, utilizamos neste trabalho operadores de diferenças, como sugerido em [Box, Jenkins 1970], para que as séries passassem a ser estacionárias. No caso de séries com tendência, aplicamos o operador de diferenças simples (Ver Equação 4.2):

$$Z_t^d = Z_t - Z_{t-1} \quad (4.2)$$

Equação 4.2: Operador de diferenças simples

Nesta equação, Z é a série original e Z^d é a série estacionária transformada.

Um outro fator que foi considerado foi o vetor de desempenho associado à previsão das séries temporais. Utilizamos um procedimento *out-of-sample* [Tashman 2000] no qual, dada uma série temporal, seus dados são divididos em: dados de ajuste e dados de teste. Os dados de ajuste consistem nas observações iniciais da série e os dados de teste nas observações finais. Os dados de ajuste são utilizados na estimação (ajuste ou treinamento) dos parâmetros dos modelos candidatos. Os modelos ajustados são utilizados então na previsão dos dados de teste. Cada vetor de desempenho na nossa implementação contém: (1) tempo de execução (resultado do processo de ajuste) e (2) erro de previsão (resultado da previsão dos modelos nos dados de teste).

O último aspecto considerado é o cálculo das características associadas às séries, às quais chamamos de meta-atributos. Essas características são utilizados pelo meta-aprendiz na recuperação dos meta-exemplos mais similares à série dada como entrada.

A escolha dos meta-atributos mais adequados para descreverem uma série temporal depende do tipo da série sendo analisada. Enquanto que a característica *tendência básica* é utilizada para séries que apresentam tendência ela não faria sentido se fosse aplicada em séries estacionárias, por exemplo.

Existem conjuntos padrão de meta-atributos que podem ser utilizados em problemas de classificação na área de meta-aprendizado, como por exemplo, o *Data Characterization Tool* [Engels, Theusinger 1998], desenvolvido dentro do projeto METAL [Metal 2003]. Na área de previsão de séries temporais não existem meta-

atributos que possam ser utilizados para qualquer tipo de problema. No entanto, podemos seguir alguns critérios para definir os melhores meta-atributos descritores de um conjunto de séries temporais. Um desses critérios seria a escolha de características que possam ser identificadas de forma confiável, evitando análise subjetiva, como inspeção visual no gráfico da série, por exemplo. A análise subjetiva é geralmente ineficiente no tempo, requer conhecimento especialista e tem baixo grau de confiabilidade [Adya et. al 2001]. Um outro critério seria a avaliação da utilização dos meta-atributos por outros autores na literatura de séries temporais, que deve ser ampla preferencialmente. Por fim, é aconselhável o uso de um número manipulável de características, evitando um gasto de tempo acima do esperado durante a seleção de modelos.

As implementações dos meta-atributos foram desenvolvidas no Matlab⁴ [Mathworks 1996]. Como veremos no Capítulo 5, cada conjunto de séries é descrito por um conjunto apropriado de meta-atributos.

4.2.2. MODELOS CANDIDATOS

Nessa seção, discutimos os critérios usados para a escolha dos modelos candidatos para a previsão das séries temporais.

O tempo constitui um ingrediente essencial no processo de aprendizagem, sendo uma entidade ordenada básica para muitas tarefas cognitivas encontradas na prática, como o processamento de sinais e o controle motor. Uma informação temporal pode ser inserida em um método de rede neural através de uma representação implícita do tempo, na qual o sinal de entrada é amostrado uniformemente. Fazendo assim, a estrutura temporal do sinal de entrada é inserida na estrutura espacial da rede [Haykin 2001].

Em oposição aos métodos tradicionais, as redes neurais são métodos auto-adaptativos dirigidos a dados, nos quais existem poucas suposições a fazer a priori sobre os modelos para o problema em estudo. As redes neurais aprendem com os exemplos de entrada e capturam relacionamentos funcionais entre os dados mesmo que estes sejam desconhecidos ou difíceis de descrever [Zhang et. al 1998].

⁴ Foi utilizada a versão 6.1 desta ferramenta.

Considerando que as redes neurais são capazes de generalizar, após o aprendizado dos dados apresentados, novas informações podem ser corretamente inferidas mesmo que as amostras apresentadas contenham ruídos. Como a previsão de uma série temporal é um processo de descoberta de comportamentos futuros a partir de informações passadas, uma rede neural apresenta-se como uma boa alternativa para utilização como modelo de previsão.

Desta forma, utilizamos apenas métodos de redes neurais como modelos de previsão para todas as séries temporais previstas neste trabalho. Os modelos utilizados foram: a Rede Neural SOMTAD (*SOM with Temporal Activity Diffusion*) [Principe et. al 2002], a Rede Neural RBF (*Radial Basis Function*) [Haykin 2001] e a Rede Neural TDNN (*Time Delay Neural Network*) [Lang, Hinton 1988].

As descrições detalhadas do funcionamento de cada um dos modelos utilizados podem ser encontradas no Apêndice A deste trabalho.

4.2.3. ABORDAGEM DE META-APRENDIZADO

A abordagem Zoomed Ranking, utilizada na geração dos rankings dos modelos candidatos, passou por uma nova adaptação além daquelas necessárias à adequação da abordagem ao problema de seleção de modelos. Na etapa de seleção das séries mais similares à série de entrada, utilizamos inicialmente uma função de distância que é parte do algoritmo dos k-vizinhos mais próximos (K-NN), como proposto originalmente. Porém, esta função apresenta como primeira dificuldade uma instabilidade nos resultados caso atributos irrelevantes ou redundantes sejam utilizados, por exemplo. Desta forma, substituímos esta função de distância por uma abordagem de Redes Neurais Artificiais: a Rede SOM (*Self-organizing Maps*) [Kohonen 1990] [Kohonen 1997]. Com esta melhoria, podemos assegurar que caso um atributo seja considerado desnecessário ao aprendizado dos dados, uma importância menor é dada a este atributo e a rede rapidamente se adapta a esta deficiência dos dados de entrada.

4.2.4. RANKING RECOMENDADO

Para a avaliação da qualidade do ranking recomendado pelo Zoomed Ranking utilizamos o coeficiente de correlação de Spearman [Neave, Worthington 1992], proposto no trabalho original. Este coeficiente mede a distância entre o ranking

recomendado e o ranking ideal, que corresponde à correta ordenação dos modelos candidatos. A cada série de teste, um ranking ideal é gerado a partir das informações de desempenho dos modelos candidatos nesta série. Contudo, o ranking ideal obtido simplesmente pela ordenação destas informações pode não capturar bem a noção da real situação dos modelos. Para solucionar este problema, o Zoomed Ranking explora o fato de que os modelos podem trocar suas posições nos rankings caso sejam comparados resultados de várias execuções diferentes do algoritmo.

Desta forma, utilizamos resultados de 10 execuções do Zoomed Ranking na representação do ranking ideal. O ranking ideal da série s_i é construído então a partir da média aritmética das j ordenações do modelo m_p em relação a cada modelo m_q , segundo a Equação 4.3:

$$ir_{s_i} = (\sum_j ARR_{m_p, m_q}^{s_i, j}) / m \quad (4.3)$$

Equação 4.3: Ranking ideal gerado pelo Zoomed Ranking

onde m é o número de modelos e $ARR_{m_p, m_q}^{s_i}$ é calculado com base nas informações de desempenho dos modelos na série s_i de teste (Ver Equação 3.2). O coeficiente de correlação de Spearman é definido de acordo com a Equação 4.4:

$$r_s = 1 - \frac{6 \sum_{i=1}^m (rr_i - ir_i)^2}{m^3 - m} \quad (4.4)$$

Equação 4.4: Coeficiente de correlação de Spearman

onde rr_i e ir_i são a ordenação do modelo m_i no ranking recomendado e ideal, respectivamente, e m é o número de modelos. Esta medida é normalizada pelo valor $m^3 - m$ para gerar valores mais significativos.

O valor 1 para este coeficiente representa o emparelhamento perfeito entre os rankings e o valor -1 representa o total desemparelhamento. O valor 0 significa apenas que os ranking não estão relacionados. Desta forma, valores próximos de 1 indicam que os rankings possuem a maioria das posições emparelhadas e valores próximos de -1 indicam que os rankings possuem quase todas as posições desemparelhadas.

4.3. CONSIDERAÇÕES FINAIS

Vimos neste capítulo todos os aspectos envolvidos na investigação do uso da abordagem Zoomed Ranking para o problema de seleção de modelos de séries temporais. Apresentamos a arquitetura utilizada na implementação da solução proposta, que faz uso de vários conceitos associados ao uso de uma técnica de meta-aprendizado para a seleção de modelos, apresentados no Capítulo 3.

Vimos ainda alguns aspectos que foram considerados durante a implementação da arquitetura proposta. Estes aspectos estavam associados às séries temporais consideradas, aos modelos candidatos à previsão, à abordagem de meta-aprendizado utilizada e à avaliação dos resultados obtidos com esta abordagem.

Os modelos candidatos à previsão, utilizados pelo Zoomed Ranking, são modelos baseados em técnicas de Redes Neurais (Ver Seção 4.2.2). A seleção entre diferentes modelos neurais se constitui como uma contribuição do presente trabalho. Outras contribuições incluem ainda uma melhoria da técnica original e sua aplicação em problemas do domínio elétrico [Rojas 2004] e dos domínios financeiro e econômico [Hyndman 2006]. No próximo capítulo serão apresentados os experimentos realizados para análise das propostas apresentadas no presente trabalho.

5. ANÁLISE EXPERIMENTAL

Neste capítulo serão apresentados os estudos de caso realizados para a validação da arquitetura apresentada no Capítulo 4 (Ver Figura 4.1). No primeiro estudo de caso foi utilizada a abordagem Zoomed Ranking original, adaptada ao problema de seleção de modelos, na previsão das séries da competição organizada pela EUNITE [Rojas 2004]. No segundo estudo de caso, utilizamos a abordagem Zoomed Ranking com a Rede SOM na fase de *zooming*, modificação anteriormente descrita na Seção 4.2.3, nas mesmas séries do primeiro estudo de caso. O Estudo de Caso III apresenta os resultados obtidos com o uso da abordagem Zoomed Ranking original em uma nova base, com séries do repositório *benchmark* da *Time Series Data Library* (TSDL) [Hyndman 2006]. No Estudo de Caso IV utilizamos o Zoomed Ranking com a Rede SOM [Kohonen 1990] [Kohonen 1997] na mesma base de séries do Estudo de Caso III. Cada um dos estudos de caso será descrito em detalhes nas próximas sessões deste capítulo.

5.1. ESTUDO DE CASO I

Neste estudo de caso foi implementado um protótipo seguindo a arquitetura apresentada no Capítulo 4. Nas próximas sessões apresentaremos detalhes importantes sobre cada um dos módulos da arquitetura implementada: o módulo Extrator de Características (Sessão 5.1.1), o módulo Base de Dados (Sessão 5.1.2) e o módulo Meta-Aprendiz (5.1.3).

Utilizamos neste estudo de caso a abordagem Zoomed Ranking como meta-aprendiz, já adaptada ao problema de seleção de modelos, nos meta-exemplos obtidos a partir das séries extraídas da competição organizada pela EUNITE.

5.1.1. MÓDULO EXTRATOR DE CARACTERÍSTICAS

Foram extraídos oito meta-atributos para a descrição das séries utilizadas neste estudo de caso:

- *Ano*: ano em que as observações da série foram medidas.

- *Mês*: mês, no ano correspondente, em que os dados da série foram observados.
- *Horário*: hora do dia em que as observações da série foram detectadas.
- *Teste de Turning Points para aleatoriedade*: um ponto da série Z_t é considerado um turning point se $Z_{t-1} < Z_t > Z_{t+1}$ ou $Z_{t-1} > Z_t < Z_{t+1}$. A presença de um número muito alto ou muito baixo de turning points sugere que a série não é gerada por um fenômeno puramente aleatório.
- *Média dos valores absolutos das 5 primeiras autocorrelações*: valores altos desse atributo sugerem que o valor de um ponto da série é muito dependente dos valores passados recentes.
- *Teste de significância para autocorrelações*: presença de no mínimo uma autocorrelação positiva entre as cinco primeiras.
- *Tendência básica*: indicado pela estatística t da inclinação do modelo de regressão linear. Quanto mais alto o valor dessa variável, maior é a tendência global da série.
- *Tamanho da série*: corresponde ao número de pontos ou observações da série.

Os primeiros meta-atributos descritos (ano, mês e horário) são características já contidas nos dados iniciais coletados. Os demais meta-atributos são características descritivas calculadas a partir dos dados das séries.

5.1.2. BASE DE DADOS

As séries utilizadas neste estudo de caso foram extraídas de uma competição organizada pela EUNITE. Cada série coletada contém cargas elétricas medidas durante todo um dia, na qual cada observação corresponde a um horário específico. Estas séries não apresentavam variabilidade significativa se comparadas dentro do mesmo mês por terem, por exemplo, a mesma correspondência de horário entre suas observações, o que tornariam os resultados pouco expressivos.

Após o processamento dos dados iniciais, disponibilizados pela competição, foram extraídas 1152 séries temporais. Cada série a ser prevista contém valores de cargas elétricas diárias dos anos de 1997 e 1998. Todas as medidas de carga de uma série estão associadas a um determinado ano, mês e horário específico. Cada observação

da série corresponde a um dia do mês correspondente. Os horários utilizados como intervalos entre as observações apresentam-se com variação de meia hora durante todo um dia, iniciando às 00:30h e terminando às 24h. Na Figura 5.1.1 apresentamos o gráfico representativo de uma das séries da base extraída da EUNITE. Esta série corresponde às observações coletadas no mês de Janeiro do ano de 1997, no horário das 00:30h.

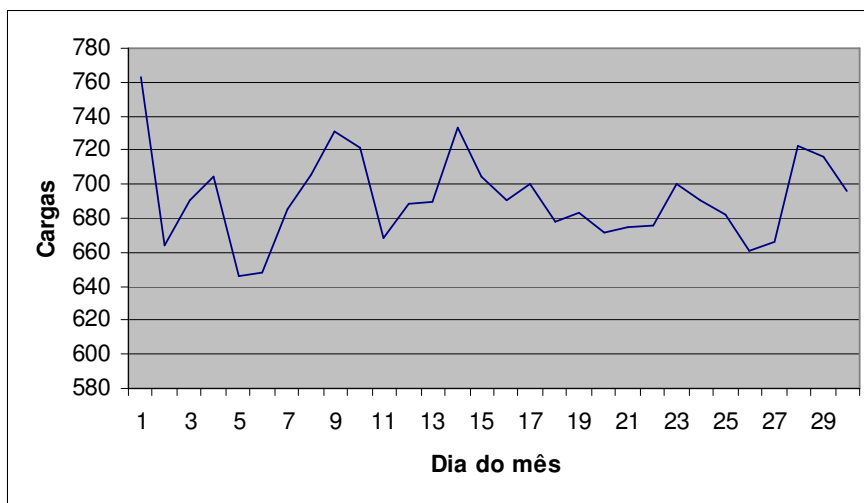


Figura 5.1.1: Exemplo de uma série da EUNITE

As previsões obtidas para estas séries correspondem às cargas elétricas dos últimos dias de cada mês, baseando-se nas informações contidas nos dados da série para os dias anteriores.

5.1.2.1. Meta-exemplos

Os dados extraídos a partir das séries coletadas foram utilizados na obtenção dos exemplos que serão utilizados no Meta-Aprendizado. Para a obtenção de um exemplo, os dados da série são divididos em dados de ajuste e dados de teste, como descrito na Seção 4.2.1. Os dados de ajuste correspondem aos 20 pontos iniciais e os dados de teste correspondem aos demais pontos da série. Os dados de ajuste são utilizados na calibração dos parâmetros de cada um dos modelos para sua posterior utilização na previsão dos dados de teste da série.

Como mencionamos na Seção 4.1, cada exemplo é então formado pelo conjunto das características da série associada ao meta-exemplo seguido pelos resultados de

desempenho de cada modelo candidato na série. As informações de desempenho obtidas foram o tempo de execução do modelo durante a etapa de ajuste dos parâmetros e a medida MAE (Média dos Erros Absolutos) (Ver Equação 2.4) das previsões nos dados de teste.

Os exemplos obtidos foram então divididos em dois conjuntos: o conjunto dos exemplos de treinamento, contendo 864 exemplos, que foram armazenados no módulo BD (Base de Dados), e o conjunto dos exemplos de teste, contendo 288 exemplos, que foram utilizados como entrada para o módulo MA (Meta-Aprendiz).

5.1.2.2. Desempenho dos modelos

Para a obtenção dos resultados de desempenho de previsão dos modelos candidatos, cada uma das redes neurais foi implementada no Matlab. Cada série recebida como entrada pelas redes foi inicialmente normalizada. Esta normalização consistiu na subtração da média e divisão pelo desvio padrão das observações da série. Após este processo, os dados de ajuste da série foram divididos igualmente em dois conjuntos: um conjunto de validação e um conjunto de treinamento.

As definições da topologia e da região de vizinhança da rede SOMTAD (Ver Apêndice B.1) foram realizadas com o uso do *SOM Toolbox* [Vesanto et. al 2000] para Matlab. No treinamento da rede SOMTAD utilizamos mapas retangulares bidimensionais 10x10, com 3 unidades na camada de entrada, taxa de aprendizado inicial η_0 igual a 0.3, parâmetro de feedback das unidades de contexto μ igual a 0.5 e parâmetro de acoplamento espaço-temporal inicial β igual a 0.5. A função de vizinhança utilizada foi a função gaussiana, definida segundo a Equação B.1.4. Os valores de cada um dos parâmetros bem como a topologia da rede foram definidos experimentalmente. Para a inicialização dos pesos da rede utilizamos valores aleatórios num intervalo entre [0;1].

A rede TDNN (Ver Apêndice B.3) foi implementada como uma rede MLP (Multilayer Perceptron) [Haykin 2001], com uma camada oculta. Para a definição da janela de tempo seguimos o teste proposto em [Box, Jenkins 1970], que faz uso da análise de autocorrelações para auxiliar a escolha do tamanho adequado da janela, com um valor máximo igual a 3. A cada série recebida como entrada pela rede, calculamos as 3 primeiras autocorrelações e testamos quais delas eram significativas considerando

um nível de confiança de 95%. O valor escolhido para a janela de tempo era aquele cuja ordem era a mais alta entre as autocorrelações consideradas significativas para o teste, ou seja, aquelas que estavam acima no nível de confiança determinado.

A definição da quantidade de neurônios da camada oculta da rede TDNN seguiu o processo sugerido em [Zhang et. al 2001]. Foi utilizado um procedimento *holdout*, no qual as redes foram treinadas com 1, 2 e 3 neurônios na camada oculta, com 10 execuções para cada configuração escolhida. A rede que obtivesse o menor valor para o SSE (*Sum of Squares Errors*) (Ver Equação 2.2) no conjunto de validação era então escolhida para utilização na previsão da série.

Utilizamos o algoritmo Levenberg-Marquardt [Gill et. al 1981] no treinamento da rede TDNN, disponível no *Nnsysid5 Toolbox* para Matlab [Norgaard 1997]. Foram assumidos três critérios de parada do treinamento: o GL_0 (perda da generalização), o P_5 (parada no progresso do treinamento) e por número máximo de ciclos (1000 ciclos de treinamento).

Para a escolha da melhor topologia da rede RBF (Ver Apêndice B.3), realizamos novamente o procedimento *holdout* seguido para a rede TDNN, com 10 inicializações para cada quantidade de neurônios na camada oculta, com um valor máximo de 3 neurônios. A definição da janela de tempo seguiu o mesmo procedimento utilizado na implementação da rede TDNN, que utiliza a análise das 3 primeiras autocorrelações da série na escolha do valor adequado para o tamanho da janela. Utilizamos a função multiquadrática inversa como função de ativação das unidades ocultas, definida segundo a Equação B.3.1, e a função linear como função de saída da rede. O algoritmo utilizado no treinamento da rede RBF foi o algoritmo de propagação adiante (*forward propagation*) para saídas lineares.

A topologia da rede, incluindo as funções de ativação e de saída, bem como o algoritmo de treinamento utilizado, foram definidos com o auxílio do *Netlab Toolbox* para Matlab [Bishop 1995] [Nabney 2002].

Nas Tabelas 5.1 e 5.2, apresentamos as médias das medidas MAE (Média dos Erros Absolutos) e as médias dos tempos de execução, obtidas para cada modelo candidato no conjunto de treinamento, respectivamente.

Tabela 5.1: Médias das medidas MAE (Média dos Erros Absolutos) no conjunto de treinamento dos meta-exemplos das séries da EUNITE

Algoritmo	MAE
SOMTAD	0.4665
RBF	1.7079
TDNN	6.6769

Tabela 5.2: Média dos tempos de execução dos modelos candidatos no conjunto de treinamento dos meta-exemplos das séries da EUNITE

Algoritmo	Média dos tempos de execução
SOMTAD	0.3567
RBF	0.3053
TDNN	0.4912

Para verificar se os erros de previsão obtidos pelos modelos candidatos são estatisticamente diferentes, realizamos testes de hipótese com os pares possíveis de modelos. Realizamos os testes com nível de significância igual a 5%. A hipótese nula foi rejeitada em todos os testes, indicando que existe diferença significativa entre erros obtidos para os três pares possíveis de modelos candidatos.

5.1.3. MÓDULO META-APRENDIZ

Neste estudo de caso, utilizamos como meta-aprendiz a abordagem Zoomed Ranking descrita na Seção 3.4, já adaptada ao problema de seleção de modelos. Desta forma, os algoritmos para problemas de aprendizado passam a ser visto como modelos de previsão de séries temporais. Os exemplos de entrada são os meta-exemplos obtidos a partir das séries de entrada, conforme descrevemos na Seção 4.1.

Para uma melhor visualização do funcionamento desta abordagem, observe na Figura 5.1 o seu pseudo-código, no qual:

- EXEMPLOS_TESTE: conjunto contendo os exemplos do conjunto de teste;
- EXEMPLOS_TREINAMENTO: conjunto contendo os exemplos do conjunto de treinamento;
- META_ATRIBUTOS: conjunto contendo os meta-atributos descritores da série;
- MODELOS: conjunto dos modelos candidatos;
- e : exemplo de teste;

- $zooming(e, Z)$: aplicação da etapa de *zooming* ao exemplo e ;
- $ArgMinimo(Distancia, n)$: função que retorna o exemplo com a n -ésima menor distância;
- $Distancia[e_1]$: contém a distância do exemplo e de entrada ao exemplo e_1 do conjunto de treinamento;
- Z : corresponde à quantidade de exemplos selecionados na etapa de *zooming*.
- $Selecionados[]$: contém os exemplos de treinamento mais semelhantes ao exemplo e . Possui Z posições;
- $dist(x, e_1, e_2)$: medida *dist* descrita anteriormente pela Equação 3.1;
- $Ranking[m_1]$: contém a posição do modelo m_1 no ranking recomendado;
- $ARR(m_1)$: medida *ARR* do modelo m_1 definida na Equação 3.3;
- $ranking_recomendado(Z)$: retorna o ranking recomendado para a série que representa o exemplo e , após a execução de $zooming(e, Z)$;

```

Para todo  $e_1, e_1 \in EXEMPLOS\_TESTE$  {
   $zooming(e_1, Z)$ ;
   $ranking\_recomendado(Z)$ ;
}

 $zooming(e_1, Z)$  {
  Inicializar  $Distancia$  com zero;
  Para todo  $e_2, e_2 \in EXEMPLOS\_TREINAMENTO$  {
    Para todo  $x, x \in META\_ATRIBUTOS$ 
       $Distancia[e_2] = Distancia[e_2] + dist(x, e_1, e_2)$ ;
  }
  Para todo  $i, i \in \{1, Z\}$  {
     $Selecionados[i] = ArgMinimo(Distancia, i)$ ;
  }
}

 $ranking\_recomendado(Z)$  {
  Para todo  $m_1, m_1 \in MODELOS$  {
    Para todo  $m_2, m_2 \in MODELOS$  {
      Se  $m_2 \neq m_1$ 
        Para todo  $i, i \in \{1, Z\}$ 
           $Ranking[m_1] = ARR(m_1)$ ;
        }
    }
  }
}

```

Figura 5.1.2: Algoritmo do Zoomed Ranking

Seguindo a arquitetura proposta, a abordagem Zoomed Ranking recebe como entrada as características da série à qual o ranking dos modelos candidatos à sua previsão deve ser gerado. A partir dessas características, o Zoomed Ranking recupera da base de dados os exemplos com as menores distâncias em relação à série de entrada. As informações de desempenho contidas nos exemplos selecionados são então utilizadas na geração do ranking dos modelos candidatos. Desta forma, é esperado que o ranking gerado pelo meta-aprendiz reflita um resultado semelhante ao obtido na previsão das séries correspondentes aos exemplos selecionados. Após a geração dos rankings das séries correspondentes aos exemplos de teste, os resultados obtidos são armazenados na base de dados para que possam ser utilizados futuramente em outros problemas do mesmo domínio.

5.1.4. EXPERIMENTOS E RESULTADOS

Nesta seção, serão apresentados os resultados obtidos com a implementação da arquitetura apresentada na Seção 4.1 para este estudo de caso. Conforme descrevemos na seção anterior, utilizamos como meta-aprendiz a abordagem Zoomed Ranking, já adaptada ao problema de seleção de modelos. As séries previstas pelos modelos candidatos, que deram origem aos meta-exemplos utilizados como entrada pelo meta-aprendiz, foram aquelas extraídas da competição organizada pela EUNITE.

Para ilustrar a aplicação do Zoomed Ranking, apresentamos o ranking obtido para uma das séries usadas como teste. Podemos observar na Tabela 5.3 os resultados obtidos pelo Zoomed Ranking com a seleção das 10 séries mais similares à série de teste dada como entrada. Utilizamos os valores 0%, 1%, 10%, 15% e 20% para o parâmetro $AccD$, todos escolhidos experimentalmente. A partir destes resultados, verificamos que a rede SOMTAD lidera todos os rankings gerados por ser tanto a mais precisa, caso seja dada uma importância maior à minimização dos erros de previsão ($AccD$ igual a 0% e 1%), quanto a mais rápida, caso seja dada uma importância maior ao tempo ($AccD$ igual a 10%, 15% e 20%). A rede TDNN passa a ocupar a segunda posição dos rankings quando desejamos obter modelos mais rápidos ($AccD$ igual a 10%, 15% e 20%), por ter apresentado tempo de processamento menor na série de entrada.

Tabela 5.3: Ranking gerado para $Z = 10$ para as séries da EUNITE

AccD	0%		1%		10%		15%		20%	
Rank	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR
1	SOMTAD	1.6803	SOMTAD	1.8199	SOMTAD	1.2788	SOMTAD	1.2773	SOMTAD	1.3382
2	RBF	0.6554	RBF	0.6647	TDNN	0.8911	TDNN	0.8944	TDNN	0.8328
3	TDNN	0.6036	TDNN	0.5008	RBF	0.7507	RBF	0.7507	RBF	0.7325

Na Tabela 5.4, mostramos os resultados obtidos com a seleção das 50 séries mais similares a cada série de entrada. Observamos que os rankings gerados são semelhantes aos obtidos na Tabela 5.3, com exceção do ranking gerado com o parâmetro *AccD* igual a 10%, que manteve a rede RBF na segunda posição do ranking.

Tabela 5.4: Ranking gerado para $Z = 50$ para as séries da EUNITE

AccD	0%		1%		10%		15%		20%	
Rank	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR
1	SOMTAD	2.3030	SOMTAD	2.2605	SOMTAD	2.2509	SOMTAD	2.3772	SOMTAD	2.3667
2	RBF	0.5437	RBF	0.5461	RBF	0.5459	TDNN	0.6513	TDNN	0.6543
3	TDNN	0.4451	TDNN	0.4573	TDNN	0.4615	RBF	0.4245	RBF	0.4257

Podemos observar na Tabela 5.5 os resultados obtidos com o aumento das séries selecionadas na fase de *zooming* para 90 séries. Com isso, verificamos que a rede TDNN só passou a ocupar a segunda posição do ranking quando abrimos mão de 20% da precisão dos modelos candidatos, ou seja, com o parâmetro *AccD* igual a 20%.

Tabela 5.5: Ranking gerado para $Z = 90$ para as séries da EUNITE

AccD	0%		1%		10%		15%		20%	
Rank	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR
1	SOMTAD	2.4104	SOMTAD	2.4087	SOMTAD	2.4238	SOMTAD	2.3859	SOMTAD	2.5593
2	RBF	0.5101	RBF	0.5102	RBF	0.5175	RBF	0.5106	TDNN	0.4699
3	TDNN	0.4547	TDNN	0.4553	TDNN	0.4398	TDNN	0.4632	RBF	0.4691

Verificamos que a quantidade de séries selecionadas na fase de *zooming* (parâmetro Z) influencia diretamente nos rankings recomendados pelo Zoomed Ranking. Com o aumento do parâmetro Z , a rede RBF passa a ocupar a segunda posição na maioria dos rankings com a diminuição simultânea da importância que é dada à precisão (aumento do *AccD*). Isto ocorre porque quanto maior for a quantidade de séries selecionadas na fase de *zooming* mais próximos dos resultados de desempenho obtidos para a série de teste serão os rankings recomendados. Como a rede RBF é mais rápida

que a rede TDNN, ela conseqüentemente passa a ocupar a terceira posição dos rankings neste contexto.

Na Tabela 5.6, apresentamos as médias dos coeficientes de correlação de Spearman obtidas para todas as configurações utilizadas nos experimentos deste estudo de caso. Como descrito no capítulo anterior (Ver Seção 4.2.4), valores próximos de 1 para o coeficiente de correlação de Spearman indicam que todas as posições dos rankings recomendados pelo Zoomed Ranking estão emparelhadas com os rankings ideais correspondentes. Desta forma, verificamos que os resultados obtidos pelo Zoomed Ranking na indicação do melhor modelo candidato são satisfatórios quando aplicados ao problema proposto neste estudo de caso. Os melhores resultados foram obtidos quando selecionamos a maior quantidade de exemplos na fase de *zooming*, entre os valores testados.

Outra forma de validar os resultados de avaliação dos rankings recomendados foi investigada com o uso de um gerador de rankings de referência. A técnica Zoomed Ranking foi utilizada levando em conta todas as séries do conjunto de treinamento na geração dos rankings. Com isso, obtivemos rankings estáticos, gerados a partir de todas as informações de desempenho disponíveis. Comparando os resultados obtidos com os resultados de referência, observamos que os valores dos coeficientes de correlação obtidos neste estudo de caso superam em todas as configurações testadas os resultados do gerador de rankings *default*.

Tabela 5.6: Média dos coeficientes de Spearman para as séries da EUNITE

AccD	r_s médio			
	Z = 10	Z = 50	Z = 90	Default
0%	0.7691	0.8142	0.9063	0.9045
1%	0.7691	0.8142	0.9045	0.6840
10%	0.7708	0.8108	0.9028	0.5990
15%	0.7795	0.8125	0.9080	0.5226
20%	0.7847	0.8177	0.9045	0.4844

5.2. ESTUDO DE CASO II

Neste estudo de caso utilizamos a abordagem Zoomed Ranking como meta-aprendiz, já com a nova adaptação na função de distância utilizada na fase de *zooming*, descrita na Seção 4.2.3. Os rankings foram gerados novamente a partir dos resultados de desempenho obtidos pelos modelos candidatos nas séries da base da EUNITE.

5.2.1. MÓDULO EXTRATOR DE CARACTERÍSTICAS

Os meta-atributos utilizados neste estudo de caso foram os mesmos anteriormente extraídos no Estudo de Caso I (Ver Seção 5.1.1).

5.2.2. BASE DE DADOS

Neste estudo de caso utilizamos novamente as séries da base extraída da competição organizada pela EUNITE, já utilizada no estudo de caso anterior. A descrição detalhada desta base de séries pode ser vista na Seção 5.1.2 deste capítulo.

Os meta-exemplos obtidos no Estudo de Caso I e armazenados pelo módulo BD foram novamente utilizados neste estudo de caso, dado que tanto as características das séries quanto os modelos candidatos à previsão são os mesmos.

5.2.3. MÓDULO META-APRENDIZ

O meta-aprendiz utilizado neste estudo de caso foi o Zoomed Ranking com a substituição da função de distância, proposta na Seção 4.2.3 deste trabalho, que propõe a utilização de uma rede SOM na fase seleção das séries mais similares à série de entrada.

A implementação da Rede SOM foi desenvolvida no Matlab, novamente com a utilização do *SOM Toolbox* para auxiliar a definição da topologia da rede e da função de vizinhança. Utilizamos mapas retangulares bidimensionais 18x18, com função de vizinhança gaussiana (Ver Equação B.3.1) e formato da região de vizinhança também retangular. Durante o treinamento da rede, a taxa de aprendizado e o raio de vizinhança foram continuamente decrementados, procedimento normal para o treinamento de uma rede SOM. A taxa de aprendizado foi decrescida gradualmente com o aumento do tempo segundo a Equação 5.1:

$$\eta(i) = \eta_0 \exp\left(-\frac{i}{\tau}\right) \quad (5.1)$$

Equação 5.1: Taxa de aprendizado para treinamento da rede SOM

onde η_0 é a taxa de aprendizado inicial, τ é uma constante de tempo do algoritmo SOM e i é a iteração em que se encontra o treinamento. Após a realização de

testes experimentais, a taxa de aprendizado inicial η_0 assumiu valor igual a 0.3 e a constante de tempo τ foi inicializada com valor igual a 400.

Durante a fase de ordenação, que na nossa implementação corresponde às 20 primeiras iterações, o tamanho da região de vizinhança foi decrementado e a função de vizinhança foi decrescida gradualmente em intervalos regulares de acordo com o tamanho inicial da região de vizinhança. Assumindo que utilizamos uma grade bidimensional de neurônios, igualamos o tamanho inicial da região de vizinhança ao “raio” da grade, ou seja, à metade da maior dimensão do mapa, como proposto em [Haykin 2001]. Levando em conta que em nossa implementação o mapa da rede SOM possui as duas dimensões iguais a 18, foi associado o valor 9 para o tamanho inicial da vizinhança. O critério de parada adotado foi a parada por número máximo de iterações, com um total de 100 iterações.

Após o treinamento da rede SOM, realizamos um processo de rotulação dos neurônios do mapa para auxiliar a identificação dos grupos formados. Inicialmente, cada exemplo de treinamento foi novamente apresentado à rede e associado ao neurônio vencedor em uma estrutura representativa dos agrupamentos. Esta estrutura continha para cada neurônio uma lista dos padrões que o ativaram. Ao final da apresentação de todos os padrões, os grupos associados à cada neurônio foram então formados.

Após este processo, iniciamos a rotulação propriamente dita para a obtenção dos grupos associados à cada exemplo apresentado. Estes grupos correspondem aos exemplos associados às séries mais similares à série que representa o exemplo de entrada da rede. O neurônio vencedor após a apresentação de cada exemplo era então buscado na estrutura representativa dos agrupamentos e o grupo associado ao mesmo retornado como saída da fase de *zooming*.

O algoritmo do Zoomed Ranking, após a adaptação da fase de *zooming* para a utilização da rede SOM pode ser observado em pseudo-código na Figura 5.2. Os novos termos inseridos neste algoritmo, além dos que já foram mencionados no Estudo de Caso I, são:

- $SOM(e)$: função que representa uma rede SOM, recebendo como entrada o exemplo e ;
- Grupos: estrutura que armazena o conjunto de exemplos agrupados para cada neurônio da rede SOM;

- Seleccionados: vetor que armazena os exemplo agrupados pelo neurônio ativado para o exemplo e de entrada;
- tam_s : tamanho do vetor Seleccionados;
- vencedor: armazena o índice para o neurônio vencedor na estrutura Grupos;
- $length(v)$: retorna o tamanho de um vetor v ;
- $concat(v_1, v_2)$: concatena os elementos do vetor v_1 aos elementos do vetor v_2 ;

```

Para todo  $e_1, e_1 \in EXEMPLOS\_TESTE$  {
   $tam_s = zooming(e_1)$ ;
   $ranking\_recomendado(tam_s)$ ;
}

zooming( $e_1$ ) {
  Para todo  $e_1, e_1 \in EXEMPLOS\_TESTE$  {
    vencedor = SOM( $e_1$ );
    Grupos(vencedor) = concat(Grupos(vencedor),  $e_1$ );
  }
  Para todo  $e_1, e_1 \in EXEMPLOS\_TESTE$  {
    vencedor = SOM( $e_1$ );
    Seleccionados = Grupos(vencedor);
  }
   $tam_s = length(Seleccionados)$ ;
}

ranking_recomendado( $tam_s$ ) {
  Para todo  $m_1, m_1 \in MODELOS$ 
  Para todo  $m_2, m_2 \in MODELOS$ 
  Se  $m_2 \neq m_1$ 
  Para todo  $i, i \in \{1, tam_s\}$ 
  Ranking[ $m_1$ ] = ARR( $m_1$ );
}

```

Figura 5.2: Algoritmo do Zoomed Ranking com uso da Rede SOM

5.2.4. EXPERIMENTOS E RESULTADOS

Os resultados apresentados nesta seção foram obtidos com a abordagem Zoomed Ranking, adaptada na fase de *zooming* para a utilização da rede SOM (Ver Seção 4.2.3), aplicada aos meta-exemplos que representam as séries extraídas da competição organizada pela EUNITE.

Como podemos observar na Tabela 5.7, a rede SOMTAD lidera os rankings gerados nos quais damos uma maior importância à precisão dos modelos, ou seja, nos

rankings em que atribuímos ao parâmetro *AccD* valores iguais a 0%, 1% e 10%. Quando damos uma maior importância ao tempo de execução dos modelos, ou melhor, quando atribuímos ao parâmetro *AccD* valores iguais a 15% e 20%, notamos que a rede RBF passa a liderar os rankings gerados. Isto ocorre porque de fato a rede RBF apresentou tempos de execução menores que os da rede SOMTAD na série sendo analisada.

Tabela 5.7: Ranking recomendado com o uso da Rede SOM para as séries da EUNITE

AccD	0%		1%		10%		15%		20%	
Rank	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR
1	SOMTAD	1.0391	SOMTAD	1.1168	SOMTAD	1.1582	RBF	1.2881	RBF	1.1171
2	RBF	1.0078	RBF	1.0949	RBF	0.9682	SOMTAD	0.9910	SOMTAD	1.0894
3	TDNN	0.8812	TDNN	0.6323	TDNN	0.7126	TDNN	0.6527	TDNN	0.6508

Na Tabela 5.8 podemos observar as médias dos coeficientes de correlação de Spearman para os rankings gerados neste estudo de caso. Notamos com estes resultados que, em média, os rankings recomendados pelo Zoomed Ranking se diferenciam em duas posições em relação ao ranking ideal, dado que todos os valores para o coeficiente de Spearman estão em torno de 0.5.

Utilizamos novamente o gerador de rankings *default* na validação dos resultados aqui apresentados. Verificamos que os resultados obtidos neste estudo de caso superam àqueles obtidos pelo gerador de referência apenas nos casos em que é dada uma importância maior ao tempo. Desta forma, concluímos que a investigação da abordagem adaptada aqui proposta em outros problemas seria relevante, dado que está sendo observada uma melhoria em relação aos rankings estáticos retornados pelo gerador *default* em alguns casos.

Tabela 5.8: Média dos coeficientes de Spearman para as séries da EUNITE com o uso da Rede SOM

AccD	r_s médio	Default
0%	0.5330	0.9045
1%	0.5608	0.6840
10%	0.5556	0.5990
15%	0.5556	0.5226
20%	0.5399	0.4844

5.3. ESTUDO DE CASO III

Neste estudo de caso apresentamos os resultados obtidos com a abordagem Zoomed Ranking como meta-aprendiz, adaptada ao problema de seleção de modelos, nos meta-exemplos obtidos com as séries da base extraída do repositório da *Time Series Data Library* (TSDL).

5.3.1. MÓDULO EXTRATOR DE CARACTERÍSTICAS

Neste estudo de caso foram utilizados cinco meta-atributos para a descrição das séries de entrada, das quais 4 estão entre as características já descritas no Estudo de Caso I (Ver Seção 5.1.1): *tamanho da série*, *teste de turning points para aleatoriedade*, *média dos valores absolutos das 5 primeiras autocorrelações* e *tendência básica*.

Além destas características, utilizamos ainda o *tipo da série*, um atributo categórico representando cada um dos domínios dos quais as séries foram extraídas: o domínio financeiro, o domínio micro-econômico e o domínio macro-econômico.

5.3.2. BASE DE DADOS

Para a obtenção da base de séries utilizada neste estudo de caso, coletamos séries dos domínios financeiro, micro-econômico e macro-econômico no repositório da *Time Series Data Library* (TSDL). Ao todo foram coletadas 83 séries temporais mensais, sendo 9 séries do domínio macro-econômico, 12 do domínio micro-econômico e 62 do domínio financeiro. Os resultados de previsão obtidos a partir destas séries correspondem ao último mês observado.

Após o processamento das séries coletadas, os meta-exemplos foram obtidos a partir das características extraídas das séries de entrada juntamente com os resultados de desempenho dos modelos candidatos nestas séries. Para a obtenção dos resultados de desempenho deste estudo de caso utilizamos o mesmo procedimento adotado no Estudo de Caso I para inicialização e treinamento dos modelos de Redes Neurais, incluindo os valores estimados para seus parâmetros. Os meta-exemplos foram então divididos em um conjunto de treinamento, contendo 62 exemplos e um conjunto de teste, contendo 21 exemplos, utilizados como entrada para o meta-aprendiz.

Na Tabela 5.9 são apresentadas as médias das medidas MAE (Ver Equação 2.4) nos dados de teste das séries da TSDL. Observamos na Tabela 5.10 as médias dos tempos de execução dos modelos candidatos nos dados de ajuste das mesmas séries.

Tabela 5.9: Médias das medidas MAE (Média dos Erros Absolutos) no conjunto de treinamento dos meta-exemplos das séries da TSDL

Algoritmo	MAE
SOMTAD	0.5698
RBF	3.1737
TDNN	116.5986

Tabela 5.10: Média dos tempos de execução dos modelos candidatos no conjunto de treinamento dos meta-exemplos das séries da TSDL

Algoritmo	Média dos tempos de execução
SOMTAD	1.2568
RBF	0.5484
TDNN	3.4761

A verificação da qualidade das previsões obtidas pelos modelos candidatos foi feita com o uso de testes de hipótese com os pares possíveis de modelos. Realizamos testes com nível de significância igual a 5%. Em todos os testes realizados com nível de significância igual a 5% a hipótese nula foi rejeitada, indicando que existe diferença significativa entre erros obtidos para os três pares possíveis de modelos candidatos.

5.3.3. MÓDULO META-APRENDIZ

Utilizamos como meta-aprendiz deste estudo de caso a abordagem Zoomed Ranking com as mesmas adaptações realizadas no Estudo de Caso I.

5.3.4. EXPERIMENTOS E RESULTADOS

Os resultados deste estudo de caso foram obtidos pela aplicação da abordagem Zoomed Ranking, adaptada ao problema de seleção de modelos, nos meta-exemplos obtidos a partir das séries da *Time Series Data Library* (TSDL).

Nos resultados apresentados nas Tabelas 5.11 a 5.13, notamos que a rede SOMTAD lidera todos os rankings recomendados pelo Zoomed Ranking. Como a rede SOMTAD apresenta a menor média entre os erros obtidos para a série de teste sendo

prevista, este resultado é refletido diretamente nos rankings quando damos uma maior importância à precisão dos modelos (*AccD* igual a 0%, 1% e 10%). Já quando é dada uma maior importância ao tempo (*AccD* igual a 15% e 20%), a rede SOMTAD continua a liderar os rankings mesmo apresentando **a maior média** do tempo de execução. Isto ocorre porque os resultados de precisão deste modelo são consideravelmente melhores que os demais, levando a um balanceamento da medida *ARR*.

Os rankings obtidos para os demais modelos, em todas as configurações testadas neste estudo de caso, continuam a refletir os resultados de desempenho aqui observados. A rede TDNN ocupa a segunda posição dos rankings recomendados quando é dada uma maior importância à precisão dos modelos, por ter apresentado um erro menor que o da rede RBF na série de teste sendo avaliada. Já quando damos uma maior importância ao tempo de execução dos modelos, a rede RBF passa a ocupar a segunda posição dos rankings recomendados por ter apresentado resultados melhores que os da rede TDNN.

Apenas quando selecionamos 16 exemplos na fase de *zooming* ($Z = 16$), observamos que a rede RBF também ocupa a segunda posição do ranking quando atribuímos 10% ao parâmetro *AccD*. Isto ocorre porque quanto mais séries forem selecionadas na fase de *zooming* mais próximos dos resultados de desempenho serão os rankings recomendados, neste caso, dos resultados dos tempos de execução.

Tabela 5.11: Ranking gerado para $Z = 10$ para as séries da TSDL

AccD	0%		1%		10%		15%		20%	
Rank	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR
1	SOMTAD	13.1332	SOMTAD	13.1132	SOMTAD	12.9474	SOMTAD	12.8661	SOMTAD	12.7921
2	TDNN	0.1920	TDNN	0.1909	TDNN	0.1809	RBF	0.1791	RBF	0.1845
3	RBF	0.1647	RBF	0.1656	RBF	0.1740	TDNN	0.1759	TDNN	0.1712

Tabela 5.12: Ranking gerado para $Z = 13$ para as séries da TSDL

AccD	0%		1%		10%		15%		20%	
Rank	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR
1	SOMTAD	5.4653	SOMTAD	5.4529	SOMTAD	5.3500	SOMTAD	20.9266	SOMTAD	20.7422
2	TDNN	1.3339	TDNN	1.3196	TDNN	1.2022	RBF	0.1556	RBF	0.1599
3	RBF	0.1404	RBF	0.1413	RBF	0.1493	TDNN	0.1125	TDNN	0.1105

Tabela 5.13: Ranking gerado para $Z = 16$ para as séries da TSDL

AccD	0%		1%		10%		15%		20%	
Rank	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR
1	SOMTAD	5.2074	SOMTAD	5.1979	SOMTAD	13.7329	SOMTAD	13.6073	SOMTAD	9.5021
2	TDNN	0.6189	TDNN	0.6140	RBF	0.2639	RBF	0.2724	RBF	0.2146
3	RBF	0.2009	RBF	0.2020	TDNN	0.0960	TDNN	0.0941	TDNN	0.2095

Tabela 5.14: Médias das medidas MAE (Média dos Erros Absolutos) no conjunto de teste dos meta-exemplos das séries da TSDL

Algoritmo	MAE
SOMTAD	0.6059
RBF	10.1577
TDNN	8.2053

Tabela 5.15: Média dos tempos de execução dos modelos candidatos no conjunto de teste dos meta-exemplos das séries da TSDL

Algoritmo	Média dos tempos de execução
SOMTAD	1.9092
RBF	1.0225
TDNN	1.7336

Podemos observar na Tabela 5.16 as médias dos coeficientes de correlação de Spearman dos rankings recomendados pelo Zoomed Ranking para as séries da TSDL. Como a maioria dos valores para este coeficiente aproximam-se de 1, concluímos que os rankings gerados neste estudo de caso são satisfatórios para as séries da TSDL.

Investigamos os resultados obtidos pelo gerador de rankings *default* para este estudo de caso (Ver Tabela 5.16), como nos estudos de caso anteriores (Ver Seção 5.1.4). Após a comparação das avaliações, vimos que os rankings gerados neste estudo de caso são mais precisos que aqueles obtidos pelo gerador de referência em todas as configurações testadas, o que mais uma vez confirma o bom uso da técnica para os domínios financeiro e econômico.

Tabela 5.16: Média dos coeficientes de Spearman para as séries da TSDL

AccD	r_s médio			
	Z = 10	Z = 13	Z = 16	Default
0%	0.6905	0.6667	0.6429	0.0000
1%	0.6905	0.6667	0.6190	0.0000
10%	0.6905	0.6667	0.6190	0.0000
15%	0.5000	0.6667	0.6190	0.0000
20%	0.5238	0.6429	0.5476	0.0238

5.4. ESTUDO DE CASO IV

Neste estudo de caso apresentamos os resultados obtidos pelo Zoomed Ranking, com a substituição da função de distância utilizada em sua versão original por uma rede SOM (Ver Seção 4.2.3), nas séries extraídas da *Time Series Data Library* (TSDL).

5.4.1. MÓDULO EXTRATOR DE CARACTERÍSTICAS

Os meta-atributos utilizados neste estudo de caso para descrever as séries da TSDL são os mesmos extraídos no estudo de caso anterior.

5.4.2. BASE DE DADOS

Utilizamos neste estudo de caso a mesma base de dados do estudo de caso anterior. Os meta-exemplos obtidos no estudo de caso anterior foram novamente utilizados neste estudo de caso, dado que tanto as características das séries quanto os modelos candidatos à previsão são os mesmos.

5.4.3. MÓDULO META-APRENDIZ

Neste estudo de caso fizemos uso da mesma abordagem utilizada no Estudo de Caso II, ou seja, a abordagem Zoomed Ranking adaptada na fase de *zooming* para a utilização de uma rede SOM em lugar da função de distância utilizada na abordagem original.

Na implementação da rede SOM para este estudo de caso utilizamos mapas retangulares bidimensionais 5x5, função de vizinhança gaussiana (Ver Equação B.3.1) e formato da região de vizinhança também retangular.

A taxa de aprendizado para o treinamento da rede SOM foi decrescida segundo a Equação 5.1, semelhantemente à implementação desenvolvida para o Estudo de Caso II. Após a realização de testes iniciais para a escolha dos valores mais adequados para os parâmetros da rede SOM, atribuímos à taxa de aprendizado inicial η_0 o valor de 0.2 e à constante de tempo τ valor igual a 55.

O critério de parada adotado foi a parada por número máximo de iterações, com um total de 100 iterações. Todos os outros detalhes de implementação apresentados para o Estudo de Caso II (Ver Seção 5.2.3) foram mantidos para este estudo de caso.

5.4.4. EXPERIMENTOS E RESULTADOS

Os resultados obtidos neste estudo de caso correspondem à aplicação da abordagem Zoomed Ranking, utilizando uma rede SOM na fase de *zooming*, nas séries da TSDL.

Podemos observar na Tabela 5.17 os rankings recomendados para este estudo de caso. Notamos que os resultados aqui obtidos são semelhantes àqueles apresentados do estudo de caso anterior. A rede RBF passa a ocupar a segunda posição dos rankings já a partir da perda de 10% da precisão em troca de modelos mais rápidos (*AccD* igual a 10%, 15% e 20%). A rede TDNN ocupa a segunda posição dos rankings quando damos uma maior importância à precisão dos modelos (*AccD* igual a 0% e 1%), por apresentar uma média dos erros de previsão menor que a da rede RBF.

Tabela 5.17: Ranking recomendado com o uso da Rede SOM para as séries da TSDL

AccD	0%		1%		10%		15%		20%	
Rank	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR	Modelo	ARR
1	SOMTAD	2.7767	SOMTAD	3.2062	SOMTAD	4.3683	SOMTAD	4.1272	SOMTAD	5.5011
2	TDNN	1.1303	TDNN	1.3934	RBF	1.2777	RBF	0.4255	RBF	0.6794
3	RBF	0.2815	RBF	0.2256	TDNN	0.0757	TDNN	0.2511	TDNN	0.1030

Na Tabela 5.18 apresentamos as médias dos coeficientes de correlação de Spearman para os rankings gerados pelo Zoomed Ranking utilizando a rede SOM, aplicado às séries da TSDL. Os valores deste coeficiente para os rankings recomendados neste estudo de caso não são tão satisfatórios quanto os obtidos para o estudo de caso anterior. Como os valores obtidos para este coeficiente estão entre 0 e 0.5 verificamos que a maioria dos rankings recomendados possui 2 posições desemparelhadas em relação ao seu respectivo ranking ideal.

Verificamos com o uso do gerador de rankings *default*, que os resultados aqui obtidos superam àqueles alcançados pelo gerador de referência em todas as configurações testadas. Com isso, concluímos que a abordagem Zoomed Ranking adaptada levou a resultados encorajadores para o problema aqui considerado.

Tabela 5.18: Média dos coeficientes de Spearman com o uso da Rede SOM para as séries da TSDL

AccD	r_s médio	Default
0%	0.2143	0.0000
1%	0.4048	0.0000
10%	0.4286	0.0000
15%	0.3333	0.0000
20%	0.4762	0.0238

5.5. CONSIDERAÇÕES FINAIS

Neste capítulo, apresentamos os resultados obtidos com a realização dos 4 estudos de caso que investigaram a solução aqui proposta. Verificamos que a abordagem Zoomed Ranking como proposta anteriormente apresentou bons resultados nos dois domínios investigados, como visto nos Estudos de Caso I e III. De maneira geral, observamos que não houve um valor para a quantidade de séries selecionadas na fase de *zooming* que fosse melhor em todos os casos. Especificamente no primeiro estudo de caso, os resultados apresentaram-se melhores com o aumento do parâmetro Z. Isto indica que os meta-atributos escolhidos para a caracterização das séries previstas não são ideais, o que é visto como uma possível melhoria para trabalhos futuros.

A adaptação realizada na técnica, proposta neste trabalho, apresentou resultados encorajadores tanto para o domínio elétrico como para os domínios financeiro e econômico quando comparados com um gerador de referência. Com isso, achamos que é válida a tentativa de melhorias da abordagem Zoomed ranking adaptada em trabalhos futuros.

6. CONCLUSÃO

Neste trabalho, utilizamos uma abordagem específica de meta-aprendizado na seleção de modelos de previsão de séries temporais. O uso de abordagens de meta-aprendizado pode levar a uma melhoria na seleção dos modelos em problemas atuais a partir do conhecimento adquirido sobre o desempenho dos modelos em problemas anteriores. A abordagem aqui investigada teve seu uso proposto para a seleção de modelos em um trabalho anterior [Santos et. al 2004a] [Santos et. al 2004b]. Os resultados obtidos foram encorajadores, servindo de motivação para o trabalho aqui apresentado.

Foram apresentadas no Capítulo 3 algumas abordagens de meta-aprendizado, propostas inicialmente para a seleção de algoritmos de aprendizado em problemas de classificação. Estas técnicas podem ser utilizadas na seleção de modelos de previsão de séries temporais, com adaptações necessárias para sua adequação ao domínio de previsão. A técnica de meta-aprendizado utilizada na solução aqui investigada foi o Zoomed Ranking, que utiliza mais de um critério de desempenho na geração de rankings dos modelos candidatos à previsão.

Como vimos no Capítulo 4, foram utilizadas no presente trabalho apenas técnicas baseadas em Redes Neurais como modelos de previsão. Considerando que as redes neurais são capazes aprender com os exemplos de entrada, informações futuras podem ser corretamente inferidas, tornando o uso destas técnicas adequadas ao problema de previsão. Os modelos de Redes Neurais utilizados foram a Rede SOMTAD (*SOM with Temporal Activity Diffusion*), a Rede RBF (*Radial Basis Function*) e a Rede TDNN (*Time Delay Neural Network*).

Outro aspecto investigado neste trabalho foi a adaptação da abordagem Zoomed Ranking na tentativa de reduzir a importância dada a características irrelevantes das séries associados aos exemplos de entrada. Originalmente, o Zoomed Ranking fez uso da função de distância definida no algoritmo dos k-vizinhos mais próximos (K-NN) para a seleção dos exemplos mais similares aos exemplos de entrada, utilizados posteriormente na geração dos rankings. Neste contexto, um outro ponto investigado foi

a modificação da técnica original para utilizar uma Rede SOM em lugar da função de distância original.

No Capítulo 5, apresentamos os resultados experimentais realizados para a verificação da viabilidade da solução investigada. Foram realizados 4 estudos de caso, que fizeram uso da abordagem Zoomed Ranking na geração de rankings para os três modelos de Redes Neurais candidatos à previsão:

- 1) No Estudo de Caso I a técnica investigada com as adaptações necessárias ao problema de seleção de modelos na previsão das séries extraídas da competição organizada pela EUNITE.
- 2) No Estudo de Caso II verificamos o comportamento da abordagem Zoomed Ranking após a adaptação da técnica para utilização de uma Rede SOM, o que se constitui como uma contribuição deste trabalho. Os exemplos utilizados pela abordagem foram àqueles obtidos no estudo de caso anterior.
- 3) No Estudo de Caso III utilizamos novamente a técnica investigada, como apresentada no Estudo de Caso I, na seleção dos modelos candidatos à previsão de séries financeiras e econômicas coletadas no repositório *benchmark* da *Time Series Data Library*.
- 4) No Estudo de Caso IV utilizamos o Zoomed Ranking, com a adaptação apresentada no Estudo de Caso II, na geração de rankings dos modelos candidatos, a partir dos exemplos obtidos no estudo de caso anterior.

Os resultados obtidos mostraram que a abordagem Zoomed Ranking, como proposta no trabalho anterior já mencionado, apresentou novamente bons resultados em todos os domínios investigados. A adaptação realizada nesta técnica levou a resultados encorajadores, superando àqueles observados para um gerador de rankings *default* (Ver Seção 5.1.4), em todas as configurações testadas para os domínios financeiro e econômico e em algumas configurações para o domínio elétrico, servindo de motivação para investigações futuras da técnica.

6.1. CONTRIBUIÇÕES

Podemos citar contribuições do trabalho aqui apresentado em várias áreas do conhecimento, como a área de Redes Neurais, a área de previsão de séries temporais e a área de meta-aprendizado.

Neste trabalho, utilizamos apenas técnicas baseadas em Redes Neurais na previsão de séries de diferentes domínios: o domínio elétrico, o domínio financeiro e o domínio econômico. Verificamos ainda uma contribuição na área de meta-aprendizado com as adaptações realizadas na abordagem Zoomed Ranking a partir da proposta original da técnica. A primeira adaptação foi necessária para a adequação da abordagem ao problema de seleção de modelos. Posteriormente, realizamos uma nova adaptação para a utilização de uma rede SOM, na tentativa de minimizar a importância dada a meta-atributos irrelevantes durante a seleção dos exemplos mais similares, utilizados na geração dos rankings.

6.2. TRABALHOS FUTUROS

Destacamos aqui alguns pontos observados como indicadores de trabalhos futuros a partir do trabalho apresentado. Os resultados obtidos com a adaptação da abordagem de meta-aprendizado utilizada neste trabalho para uso de uma rede SOM foram encorajadores, servindo de motivação para futuras investigações em problemas de outros domínios.

Outro ponto a investigar é a escolha adequada do conjunto de meta-atributos descritores das séries a serem previstas. No primeiro estudo de caso, verificamos que os resultados melhoraram à medida que aumentamos a quantidade de exemplos selecionados para a geração dos rankings, mostrando que os atributos escolhidos não foram suficientemente significativos. Este conjunto está intimamente relacionado ao domínio e ao tipo das séries utilizadas.

Vimos que a geração de rankings se apresenta como uma boa alternativa para a solução do problema seleção de modelos, considerando que estas técnicas mostram a relação entre os modelos candidatos por ordem de melhor indicação. Esta é uma característica que possibilita a utilização de um modelo no qual se tenha uma maior experiência, por exemplo, caso o desempenho obtido pelo mesmo seja ligeiramente menor que aquele obtido pelo modelo que lidera o ranking. É possível ainda utilizar

mais de um modelo nas previsões caso recursos suficientes (tempo, poder de processamento, entre outros) estejam disponíveis. Com isso, um último aspecto que pode ser investigado em trabalhos futuros é o uso de novas técnicas de geração de rankings para realização de um estudo comparativo com a técnica aqui utilizada.

APÊNDICE A. REDE SOM

As redes SOM (*Sel-organizing Maps*), foram desenvolvidas por Teuvo Kohonen [Kohonen 1990] [Kohonen 1997] e são baseadas no mapa topológico do córtex cerebral. O cérebro dos animais mais evoluídos possui áreas responsáveis por funções específicas, como a fala, a visão, o controle motor, entre outras. Os neurônios estão espacialmente organizados nestas áreas. Desta forma, os neurônios que estão topologicamente próximos tendem a responder a estímulos semelhantes. Esta ordenação topológica ocorre através do uso de um mecanismo de feedback lateral entre as células do córtex cerebral.

A.1. ARQUITETURA

Em um mapa auto-organizável, os neurônios estão colocados em nós de uma grade que é normalmente uni- ou bidimensional. Os neurônios se tornam seletivamente sintonizados a vários padrões de entrada ou classes de padrões de entrada no decorrer de um processo de aprendizagem [Haykin 2001]. As localizações dos neurônios assim sintonizados se tornam ordenadas entre si de forma que um sistema de coordenadas significativo para diferentes características de entrada é criado sobre a grade [Kohonen 1990]. Um mapa auto-organizável é caracterizado pela formação de um mapa topográfico dos padrões de entrada no qual as localizações espaciais dos neurônios na grade são indicativas das características estatísticas intrínsecas contidas nos padrões de entrada, daí o nome “mapa auto-organizável” [Haykin 2001].

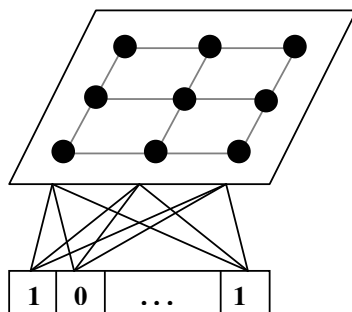


Figura A.1: Arquitetura de uma rede SOM bidimensional

Cada neurônio de uma rede SOM recebe todas as entradas e funciona como um extrator de características (Ver Figura A.1). Quanto maior a semelhança entre a entrada e o vetor de pesos de um neurônio, maior será o valor do neurônio de saída associado ao mesmo.

A.2. TREINAMENTO

A rede SOM utiliza aprendizagem competitiva, de forma que os neurônios de saída do mapa competem entre si para serem ativados, havendo apenas um neurônio vencedor. Esta competição é chamada de *winner-takes-all*, o vencedor leva tudo. A definição do neurônio vencedor ocorre segundo a Equação A.1.

$$i(x) = \arg \min_j \|x(t) - w_j\| \quad (\text{A.1})$$

Equação A.1: Distância mínima euclidiana

onde \mathbf{x} é o vetor de entrada e \mathbf{w}_j é o vetor de peso do neurônio j . Uma maneira de implementar a competição entre os neurônios do mapa é através da utilização de conexões laterais inibitórias entre os nodos de saída. Através da introdução do conceito de vizinhos topológicos dos nodos vencedores, o algoritmo de treinamento utilizado pela rede SOM simula o efeito da função “chapéu mexicano”. Neste processo, apenas o nodo vencedor e seus vizinhos dentro de um certo raio ou área de vizinhança atualizam seus pesos. Durante o treinamento, a taxa de aprendizado e o raio de vizinhança são continuamente decrementados [Braga et. al 2000]. A atualização dos pesos do neurônio vencedor e de seus vizinhos ocorre segundo a Equação A.2.

$$w_j(t+1) = w_j(t) + \eta(t)h_{j,i(x)}(t)(x - w_j(t)) \quad (\text{A.2})$$

Equação A.2: Ajuste do vetor de peso da rede SOM

onde $\mathbf{w}_j(t)$ é o vetor de peso neurônio j no instante de tempo t , $\mathbf{w}_j(t+1)$ é o vetor de peso atualizado no instante $t+1$ e $h_{j,i(x)}$ é a função de vizinhança aplicada ao neurônio j considerando que o neurônio vencedor é dado por $i(\mathbf{x})$ (Ver Equação A.2.1) e \mathbf{x} é o vetor de entrada.

A escolha adequada dos valores associados aos parâmetros do algoritmo de uma rede SOM leva a uma representação organizada dos padrões de entrada. A atualização dos pesos do mapa (Ver Equação A.2.2), ocorre em duas fases: uma fase de ordenação ou auto-organização e uma fase de convergência, descritas em [Kohonen 1982].

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau}\right) \tag{A.3}$$

Equação A.3: Taxa de aprendizado da rede SOM

Durante a fase de ordenação ocorre a ordenação topológica dos vetores de peso, que são inicializados com valores aleatórios. Nesta fase, o treinamento busca agrupar os nodos do mapa topológico em clusters ou agrupamentos, de modo a refletir a distribuição dos padrões de entrada [Braga et. al 2000]. Os parâmetros que devem ser cuidadosamente alterados durante esta fase são a taxa de aprendizagem e a função de vizinhança. A taxa de aprendizagem deve decrescer gradualmente, segundo a Equação A.2.3. O tamanho da região de vizinhança envolve inicialmente todos os neurônios da rede e é reduzido gradualmente até atingir um raio contendo um ou dois neurônios vizinhos. A Figura A.2 mostra possíveis formatos para a região de vizinhança da rede SOM.

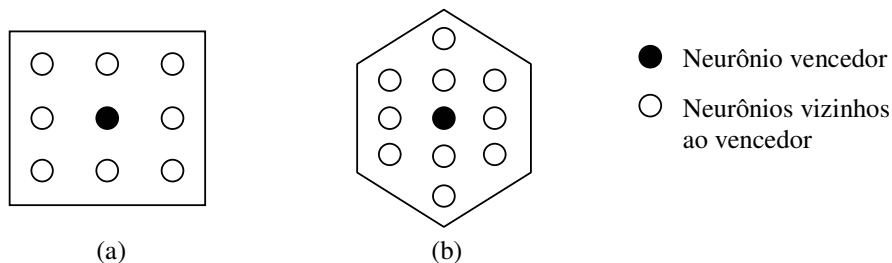


Figura A.2: Formatos possíveis da região de vizinhança para uma rede SOM. (a) Forma retangular. (b) Forma hexagonal

Esta segunda fase do processo adaptativo é necessária para realizar uma sintonia fina do mapa de características e assim produzir uma quantização estatística precisa do espaço de entrada [Haykin 2001]. Durante esta fase, a taxa de aprendizagem deve permanecer baixa, próxima de 0.01 e o raio da vizinhança deve conter apenas um ou

nenhum vizinho. Esta fase sofisticava o mapeamento realizado no estágio anterior, aprimorando o agrupamento realizado [Braga et. al 2000].

O algoritmo para o treinamento de uma rede SOM pode ser definido da seguinte forma:

- I. Inicializar os pesos e os parâmetros da rede
- II. **Para** cada padrão de entrada x **faça**
 - a. Definir neurônio vencedor
 - b. Atualizar os pesos do neurônio vencedor e de seus vizinhos
 - c. **Se** o número do ciclo de execução for múltiplo de M **então**
 - i. Reduza a taxa de aprendizado
 - ii. Reduza o tamanho da vizinhança

APÊNDICE B. MODELOS DE SÉRIES TEMPORAIS

B.1. REDE NEURAL SOMTAD

A maneira simples com que sistemas biológicos tratam informações temporais em nosso cérebro não é observada quando tentamos desenvolver princípios computacionais para aprendizado e auto-organização no espaço-tempo. Desta forma, apresentaremos nas próximas sessões a Rede Neural SOMTAD, que é capaz de representar informação temporal em um modelo auto-organizável.

B.1.1. ESTRUTURAS DE MEMÓRIA PARA SEQUÊNCIAS TEMPORAIS

A interpretação de informações temporais requer pelo menos dois tipos de memória: a memória de curto prazo e a memória de longo prazo.

Na neurobiologia, a memória de longo prazo é utilizada sempre que um estímulo longo está concluindo sua passagem. A maior parte dos modelos de memória de longo prazo estudados na neurocomputação mostram-se estáticos, ou seja, a resposta do sistema depende apenas da entrada atual. As classes mais comuns de memórias de longo prazo em sistemas neurocomputacionais são: memórias endereçáveis pelo conteúdo (CAM) [Anderson 1983], quantização vetorial (VQ) [Garani, Principe 2001] e mapas auto-organizáveis (SOM) [Kohonen 1997]. Estes modelos codificam os padrões de entrada em matrizes de pesos utilizando princípios *Hebbianos* [Hebb 1949]. Um problema observado com a utilização da memória de longo prazo é que, quando o sistema chega a um estado estável, ocorre a perda de informações importantes do passado recente, ou seja, o contexto é perdido.

Assim, um outro tipo de memória, indispensável à extração de informações embutidas na estrutura das séries temporais, é a memória de curto prazo [Elman 1990]. As memórias de curto prazo são utilizadas para reforçar estímulos correntes através das informações do histórico passado recente das séries temporais (informação contextual). Os modelos neurais para memórias de curto prazo são divididos em dois grupos:

mapeamento tempo-espaço, como as redes neurais TDNN [Lang, Hinton 1988]; e o conceito de feedback local, tendo como exemplo os elementos de processamento de contexto (PE).

B. 1.2. AUTO-ORGANIZAÇÃO COM REAÇÕES DE DIFUSÃO

A reação de difusão é um poderoso conceito que tem sido utilizado extensivamente para explicar a formação de padrões na Biologia. Elas explicam como interações locais podem surgir em estruturas espaciais globais.

O gás óxido nítrico (NO) está envolvido em muitos processos do sistema nervoso central, tais como a modificação da força sináptica nos mecanismos primários de aprendizado, que é comumente usado em RNA's. Grandes quantidades de NO em uma sinapse ativa fortalecem a sinapse, o que é chamado de Potenciação de Longo Prazo (LTP). Por outro lado, se o nível de NO está baixo, a força da sinapse é reduzida, o que recebe o nome de Depressão de Longo Prazo (LTD).

O NO é comumente chamado de mensageiro da difusão, por ter a habilidade de carregar informações sem qualquer contato elétrico direto (sinapse) através de distâncias muito mais longas do que as normalmente consideradas (não-local). O mecanismo de difusão tem se mostrado capaz de suportar o desenvolvimento de mapas topográficos sem a necessidade de uma interação lateral (chapéu mexicano). O NO pode agir no rastreamento da memória e permitir a conversão das correlações de entradas temporais em forças de conexões espaciais.

B. 1.3. O MODELO SOMTAD

A arquitetura do modelo SOMTAD [Principe et. al 2002] é baseada em dois elementos principais: a auto-organização de PE's similares em agrupamentos e a difusão da informação sobre o tempo e o espaço. A maioria das redes neurais temporais utiliza memória de curto prazo para transformar tempo em espaço. Estas redes operam como se os padrões temporais fossem padrões espaciais um pouco mais longos. A rede SOMTAD usa difusão para criar auto-organização no tempo e no espaço. A idéia é manter os fundamentos da auto-organização inalterados, mas por outro lado criar uma vizinhança correlacionada temporalmente nas PE's.

A SOMTAD é implementada por um vetor de neurônios integradores associados à saída da rede. A ativação das PE's difunde a informação através do seu espaço de saída, com decaimento no tempo. O vencedor é determinado pela combinação da informação espacial com valores dos pesos, ou seja, a competição é estendida no espaço-tempo. Conseqüentemente, o modelo armazena o passado da entrada no padrão espacial da onda que representa a atividade (memória de longo prazo). Durante a operação da rede, a onda de atividade se mantém forte apenas quando a vizinhança espacial das PE's é ligada pela ordem temporal da seqüência de entrada (memória de curto prazo).

Similarmente à rede SOM, a SOMTAD é treinada com aprendizado competitivo [Rumelhart, Zisper 1985] e utiliza uma função de vizinhança. A entrada é simultaneamente comparada aos pesos de cada PE. A unidade que tiver o casamento mais próximo entre a entrada e os pesos armazenados é a vencedora. As unidades da camada de entrada funcionam como uma janela de tempo, fornecendo os dados de entrada que são utilizados na definição do neurônio vencedor. Desta forma, a saída da rede é definida com base nas informações passadas recebidas como entrada pela rede.

Após a definição do neurônio vencedor, seus pesos e de seus vizinhos são treinados com aprendizado Hebbiano, tornando-os mais próximos da entrada atual. Como podemos observar na Figura B.1, as unidades de contexto na saída da SOMTAD são ligadas aos pesos de todos os seus vizinhos mais próximos.

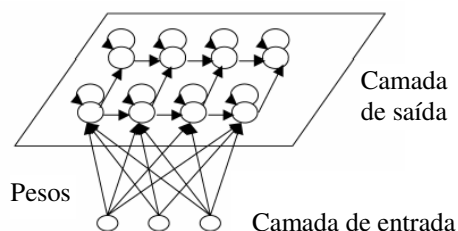


Figura B.1: Arquitetura da rede SOMTAD

Quando a atividade de difusão se espalha pela vizinhança das PE's, seus limiares são reduzidos proporcionalmente, criando uma situação onde a vizinhança das PE's é comumente disparada em seguida.

O *enhancement* é a quantidade pela qual o limiar das PE's é reduzido. O *enhancement* local permite que entradas ordenadas temporalmente disparem saídas ordenadas espacialmente.

B.1.4. DESCRIÇÃO DO ALGORITMO

A onda de atividade no modelo SOMTAD é modelada pela Equação B.1.1:

$$a(x,t) = (1 - \mu) a(x,t - 1) + \mu \|X - W_x\| \quad (\text{B.1.1})$$

Equação B.1.1: Função de ativação no modelo SOMTAD

onde μ é o parâmetro de feedback das PE's e $a(x, t)$ denota a atividade da PE localizada em x com vetor de pesos W_x no instante t . A unidade vencedora é escolhida pelo acoplamento da distância espacial com o decaimento temporal da onda de atividade sendo propagada para a vizinhança das PE's. Pela distância euclidiana descrita na Equação B.1.2, o vencedor k^* é dado por:

$$k^* = \arg \min_k (\|X - W_k\| - \beta a(x,t)) \quad (\text{B.1.2})$$

Equação B.1.2: Definição da unidade vencedora da SOMTAD

onde β é o parâmetro de acoplamento espaço-temporal. O parâmetro β determina a quantidade com que a onda temporal afeta o limiar para determinar o vencedor. O aumento de β leva à redução do limiar da vizinhança do PE, forçando a vizinhança a ser disparada seqüencialmente.

Uma vez que o vencedor é selecionado, ele é treinado com a regra de aprendizado de Kohonen [Kohonen 1997], apresentada na Equação B.1.3:

$$w_{k^*}(n+1) = w_{k^*}(n) + \eta \text{neigh}(X) (\|X - W_{k^*}\|) \quad (\text{B.1.3})$$

Equação B.1.3: Regra de aprendizado de Kohonen

onde $\text{neigh}(X)$ é a função de vizinhança e η é a taxa de aprendizado. A função de vizinhança utilizada pela rede SOMTAD foi a função Gaussiana, definida segundo a Equação B.1.4.

$$z(x) = \exp\left(-\frac{\|x - \mu\|^2}{2\sigma^2}\right) \quad (\text{B.1.4})$$

Equação B.1.4: Função Gaussiana

onde μ e σ são o centro e o raio da vizinhança, respectivamente, e $\|x - \mu\|$ é a norma Euclidiana.

A arquitetura SOMTAD cria uma memória distribuída espacialmente, onde parte da equação de ativação da reação de difusão é implementada pelas equações do SOM tradicional e a difusão é implementada pelo *enhancement* temporal. O *enhancement* é definido segundo a Equação B.1.5:

$$e(x, t) = \sum_{k=0}^n \sum_{\tau=0}^n d(x-k, t-k-\tau) \mu^k (1-\mu)^\tau \quad (\text{B.1.5})$$

Equação B.1.5: *Enhancement* temporal da SOMTAD

A Equação B.1.4 mostra como os resultados da atividade de casamento contribuem para o *enhancement*. A onda de ativação cria dois decaimentos exponenciais, como podemos observar na Figura B.1.2, um sobre o espaço, μ_k , e outro sobre o tempo, $(1-\mu)\tau$. O histórico do passado das PE's é adicionado ao *enhancement* através de um loop auto-recursivo em $(1-\mu)$.

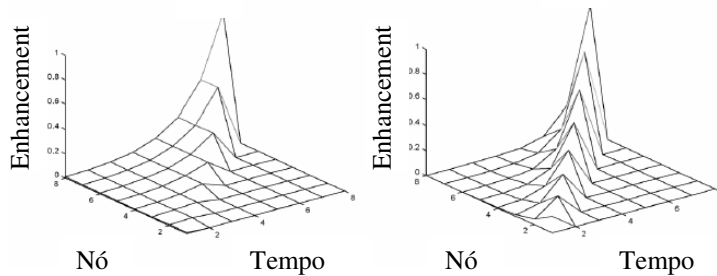


Figura B.1.2: Enhancement com $\mu = 0.5$ (esquerda) e $\mu = 0.75$ (direita) da SOMTAD

Como podemos observar, o parâmetro μ tem efeitos opostos quando disperso sobre o tempo e sobre o espaço. Durante os estágios iniciais, o parâmetro β é muito pequeno, como ocorre no aprendizado do modelo de Kohonen estático. Contudo, com o decorrer do tempo, o parâmetro β é aumentado para possibilitar o aprendizado de

relacionamentos temporais pela rede. Posteriormente, a função *enhancement* novamente decai similarmente ao processo de anelamento do SOM para fino ajuste.

As equações para a taxa de aprendizado η e para o parâmetro de acoplamento β em qualquer instante de tempo, são dadas segundo as Equações B.1.6 e B.1.7:

$$\eta(t) = \eta_0 e^{-\frac{t}{T}} \quad (\text{B.1.6})$$

Equação B.1.6: Equação para a taxa de aprendizado da SOMTAD

$$\beta(t) = \frac{\beta_0}{Max} \left(1 + \sin \frac{t\pi}{2T} \right) \quad (\text{B.1.7})$$

Equação B.1.7: Equação do parâmetro de acoplamento β da SOMTAD

onde T é o número total de épocas de treinamento.

B.2. REDE NEURAL RBF

As funções de base radial foram primeiramente introduzidas na solução do problema de interpolação multivariada real. No contexto de uma rede neural, as unidades ocultas fornecem um conjunto de “funções” que constituem uma “base” arbitrária para padrões de entrada, quando eles são expandidos sobre o espaço oculto: estas funções são chamadas de funções de base radial [Haykin 2001]. O funcionamento da rede RBF é inspirado na propriedade de alguns neurônios biológicos chamada de resposta localmente sintonizada (*locally tuned response*). Estas células nervosas respondem seletivamente a um intervalo finito do espaço de sinais de entrada.

B.2.1. ARQUITETURA

A arquitetura típica de uma rede de função de base radial (RBF, *radial-basis function*) pode ser observada na Figura B.2.

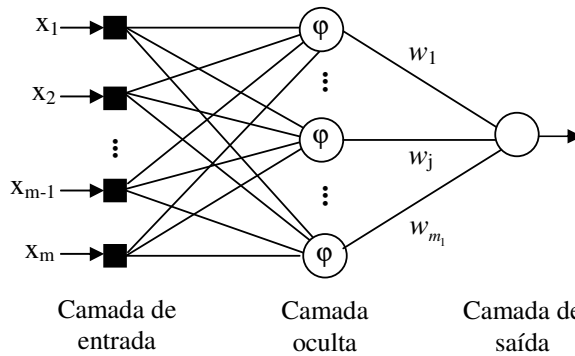


Figura B.2: Rede de função de base radial

As redes RBF são redes *feedforward* com três camadas: camada de entrada, camada intermediária e camada de saída. Cada camada de uma rede RBF desempenha um papel específico para seu comportamento [Braga et. al 2000]. A camada de entrada é constituída por unidades sensoriais que simplesmente propagam o sinal de entrada da rede. Os nodos da camada intermediária são localmente sintonizáveis e utilizam funções de base radial para o agrupamento dos dados de entrada em *clusters*. As funções de base radial representam uma classe especial de funções cujo valor diminui ou aumenta em relação à distância de um ponto central [Braga et. al 2000]. Existem diferentes funções de base radial que podem ser utilizadas em uma rede RBF. Neste trabalho, utilizamos a função multiquadrática inversa, descrita pela Equação B.2.1.

$$z_j(x) = \left[\frac{1}{\left(\|x - \mu_j\|^2 + \sigma_j^2 \right)^{\frac{1}{2}}} \right] \quad (\text{B.2.1})$$

Equação B.2.1: Função multiquadrática inversa

onde \mathbf{x} é o vetor de entrada, $\|\mathbf{x} - \mu_j\|$, que é dado pela distância euclidiana entre o vetor de entrada \mathbf{x} e o neurônio j , μ_j e σ_j representam o centro e a largura da função de base radial, respectivamente. A camada de saída utiliza uma função linear, que fornece a resposta da rede para o padrão de entrada. A resposta gerada por um neurônio de saída é dada segundo a Equação B.2.2.

$$y_j = \sum_{j=1}^h w_{ji} z_j(x) \quad (\text{B.2.2})$$

Equação B.2.2: Função de saída da rede RBF

onde w_{ji} é o peso entre a unidade escondida j e a unidade de saída i e $z_j(\mathbf{x})$ é a saída do neurônio j para o vetor de entrada \mathbf{x} .

B.2.2. TREINAMENTO

O treinamento de uma rede RBF, na maioria dos casos, é feito com aprendizagem híbrida, dividida em dois estágios. No primeiro estágio, ocorre o treinamento da camada escondida, definindo os parâmetros das funções de base radial (localização dos centros e largura dos campos receptivos). Os valores dos centros podem ser definidos por seleção aleatória [Li, Chen 1991], distribuição sobre uma grade regular [Bishop 1995] ou por alguma técnica de agrupamento, como o algoritmo das k -médias [Darken, Moody 1990], por exemplo. Na segunda fase, ocorre a definição dos valores dos pesos entre a camada oculta e a de saída. Os pesos podem ser determinados por diversos métodos como o método dos mínimos quadrados [Rousseeuw 1984] e o método da regra delta [Widrow, Hoff 1960]. Nesta fase, os parâmetros definidos na primeira fase não se modificam. Este tipo de treinamento é o mais utilizado, porque, na maioria dos casos, as saídas desejadas para a rede não são conhecidas inicialmente.

B.3. REDE NEURAL TDNN

A TDNN é uma rede *feedforward* de múltiplas camadas cujos neurônios ocultos e neurônios de saída são replicados ao longo do tempo [Haykin 2001]. Este tipo de rede tem sido utilizada com sucesso em várias aplicações práticas, incluindo reconhecimento de discurso [Waibel et. al 1989] e previsão de séries temporais [Wan 1993].

A topologia de uma rede TDNN (*Time Delay Neural Network*) [Lang, Hinton 1988] está embutida em uma rede MLP (*Multilayer Perceptron*) [Rumelhart et. al 1986] e utiliza atrasos de tempo na realização do processamento temporal. A diferença entre uma rede MLP tradicional e uma rede TDNN é que na rede TDNN as entradas para qualquer nó i podem consistir em saídas de nós anteriores não apenas no instante de

tempo corrente t , mas durante algum número d de passos anteriores ($t - 1, t - 2, \dots, t - d$) [Clouse et. al 1997]. A camada de entrada dessas redes recebe comumente o nome de janela de tempo, uma vez que, fornece a cada instante de tempo apenas uma visão parcial da série [Haykin 1994].

Utilizando uma rede TDNN na previsão de séries temporais, o neurônio de saída retorna a previsão da série em um dado tempo a partir dos valores passados fornecidos na camada de entrada.

REFERÊNCIAS

- [Adya et. al 2001] M. Adya, F. Collopy, J. S. Armstrong and M. Kennedy. Automatic identification of time series fetures for rule-based forecasting. *International Journal of Forecasting*, 17(2): 143-157, 2001.
- [Aha et. al 1991] D. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*. 6: 37-66, 1991.
- [Aha 1992] D. Aha, Generalizing from Case Studies: A Case Study. *Proceedings of the 9th International Workshop on Machine Learning*, Morgan Kaufmann, pages 1-10, 1992.
- [Armstrong, Collopy 1992] J. S. Armstrong and F. Collopy. Error Measures for Generalizing About Forecasting Methods: Empirical Comparisjons. *International Journal of Forecasting*, 8: 69-80, 1992.
- [Armstrong 2001] J. S. Armstrong. Should We Redesign Forecasting Competitions: Commentaries on the M3-Competition. *International Journal of Forecasting*, 17: 542-545, 2001.
- [Anderson 1983] J. Anderson. Cognitive and psychological computation with neural networks. *IEEE Transactions on Systems, Man and Cybernetics*, SMC- 13, pages 799–815, 1983.
- [Arinze 1994] B. Arinze. Selecting Appropriate Forecasting Models Using Rule Induction. *Omega-International Journal of Management Science*, 22(6): 647-658, 1994.
- [Ashley 2003] R. Ashley. Statistically significant forecasting improvements: how much out-of-sample data is likely necessary? *International Journal of Forecasting*, 19(2):229–239, 2003.
- [Atkeson et. al 1997] C. G. Atkeson, A. W. Moore and S. Schaal. Locally Wighted Learning. *Artificial Intelligence Review*, 11: 11-73, Kluwer, 1997.
- [Bishop 1995] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

- [Box, Jenkins 1970] G. E. Box & G. M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, CA, 1970.
- [Brazdil et. al 1994] P. Brazdil, J. Gama and R. J. Henery. Characterizing the applicability of classification algorithms using meta-level learning. *Proceedings of European Conference on Machine Learning (ECML94)*, pages 83-102, Springer-Verlag, 1994.
- [Braga et. al 2000] A. P. Braga, A. C. P. L. F. Carvalho, T. B. Ludermir. *Redes Neurais Artificiais: teoria e aplicações*. LTC, 2000.
- [Brazdil, Soares 2000] P. Brazdil and C. Sores. A comparison of ranking methods for classification algorithms selection. *Machine Learning: Proceedings of the 11th European Conference on Machine Learning (ECML2000)*, pages 63-74, 2000.
- [Brazdil et. al 2003] P. Brazdil, C. Soares and J. P. da Costa. Ranking Learning Algorithms: Using IBL and Meta-Learning on Accuracy and Time Results, *Machine Learning*, 50(3) 251-277, 2003.
- [Li, Chen 1991] J. Li and B. J. Chen. Compiling communication efficient program for massively parallel machines. *IEEE Transactions on Parallel and Distributed Systems*, 2: 361-376, 1991.
- [Chen et. al 2001] B. J. Chen, M.-W. Chang and C.-J. Lin. Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001. *IEEE Transactions on Power Systems*, 19(2004): 1821-1830, 2001.
- [Chatfield 2001] C. Chatfield. *Time Series Forecasting*. Chapman & Hall/CRC, 2001.
- [Clark, Niblett 1989] P. E. Clark and T. Niblett. The CN2 Induction Algorithm. *Machine Learning*, 3: 261-284, 1989.
- [Clouse et. Al 1997] D. S. Clouse, C. L. Giles, B. G. Horne and G. W. Cottrell. Time-Delay Neural Networks: Representation and Induction of Finite State Machines. *IEEE Transactions on Neural Networks*, 8(5): 1065-1070, 1997.
- [Collopy, Armstrong 1992] F. Collopy and J. S. Armstrong. Rule-based forecasting: development and validation of an expert systems approach to combining time series extrapolations. *Managment Science*, 38(10): 1394-1414, 1992.

- [Collopy et. al 1994] F. Collopy, M. Adya and J. S. Armstrong. Principles of Examining Predictive Validity: The Case of Information Systems Spending Forecasts. *Information Systems Research*, 5: 170-179, 1994.
- [Cover, Hart 1967] T. M. Cover and P. E. Hart. Nearest Neighbor Pattern Classification, *In IEEE Transactions on Information Theory*, IT-13, pages 21-27, 1967.
- [Darken, Moody 1990] C. Darken and J. Moody. Fast adaptive k-means clustering: some empirical results. *International Joint Conference on Neural Networks (IJCNN)*, 2: 233-238, 1990
- [Dorffner 1996] G. Dorffner. Neural Networks for Time Series Processing. *Neural Network World*, 6(4): 447-468, 1996.
- [Elman 1990] J. Elman. Finding structure in time, *Cognitive Science*, 14:179, 1990.
- [Engels, Theusinger 1998] R. Engels and C. Theusinger, Using a Data Metric for Offering Preprocessing Advice in Data Mining Applications, *Proceedings of the 13th Biennial European Conference on Artificial Intelligence*, pages 430-434, John Wileys & Sons, 1998.
- [Fildes 1989] R. Fildes. Evaluation of Aggregate and Individual Forecast Method Selection Rules. *Management Science*, 15(9): 1056-1065, 1989.
- [Garani, Principe 2001] S. Garani and J. Principe. Dynamic vector quantization of speech. *Workshop on Self-Organizing Maps (WSOM'01)*, London: Springer, 2001.
- [Gill et. al 1981] P. R. Gill, W. Murray and M. H. Wright. The Levenberg-Marquardt Method. London: Academic Press, pages 136-137, 1981.
- [Giraud-Carrier 2004] C. Giraud-Carrier, R. Vilalta and P. Brazdil. Introduction to the special issue on mete-learning, *Machine Learning*, 54(3): 187-193, 2004.
- [Harvey 1993] A. C. Harvey. *Time Series Models*. MIT Press, Cambridge, MA, 1993.
- [Haykin 1994] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company, New York, 1994.
- [Haykin 2001] S. Haykin. *Redes Neurais: princípios e prática*. Bookman, Porto Alegre, 2001.

-
- [Hebb 1949] D.O. Hebb. *The organization of Behavior: A Neuropsychological Theory*. New York, Wiley, 1949.
- [Hyndman 2006] Hyndman, R.J. *Time Series Data Library*, [<http://www.robhyndman.info/TSDL/>], Accessed on 23/04/2006.
- [Kalousis, Theoharis 1999] A. Kalousis and T. Theoharis. NOEMON: Design, Implementation and Performance Results of an Intelligent Assistant for Classifier Selection, *Intelligent Data Analysis*, 3(5): 319-337, 1999.
- [Kohonen 1982] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43: 59-69, 1982.
- [Kohonen 1990] T. Kohonen. The self-organizing map. *Proceedings of the Institute of Electrical and Electronics Engineers*, 78: 1464-1480, 1990.
- [Kohonen 1997] T. Kohonen. *Self-organizing maps*. (2nd ed.), Berlin: Springer, 1997.
- [Lang, Hinton 1988] K. J. Lang and G. E. Hinton. A time-delay neural network architecture for speech recognition. *Technical Report CMU-DS-88-152*, Department of Computer Science, Carnegie Mellon University, 1988.
- [Lindner, Studer 1999] G. Lindner and R. Studer. Ast: Support for algorithm selection with a CBR approach. *Proceedings of the 3rd International Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-99)*, Czech Republic, pages 418-423, 1999.
- [Makridakis et al 1982] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler. The accuracy of extrapolation (time series) methods; Results of a forecasting competition. *Journal of Forecasting*, 1: 111-153, 1982.
- [Makridakis et al 1993] S. Makridakis, C. Chatfield, M. Hibon, M Lawrence, T. Mills, J. K. Ord and L. Simmons. The M2-Competition: A real-life judgmentally based forecasting study with discussion. *International Journal of Forecasting*, 9:5-29, 1993.
- [Makridakis, Hibon 2000] S. Makridakis and M. Hibon. The M3-Competition: Results, Conclusions and Implications. *International Journal of Forecasting*, 16(4): 451-476, 2000.

- [Mascaat 2006] MASCAAT: Meta-Learning for Selection and Combination of Clustering Algorithms Applied to Temporal Series. [<http://algorithmics.molgen.mpg.de/probral/>], Accessed in 31 de julho de 2006.
- [Mathworks 1996] The Mathworks. *MATLAB Language Reference Manual*. The Mathworks Inc., Natick, MA, 1996.
- [Merz 1996] C. J. Merz. Dynamical Selection of Learning Algorithms, In: D. Fisher and H.-J. Lenz (eds.), *Learning from Data: Artificial Intelligence and Statistics*. Springer-Verlag, New York, 1996.
- [Metal 2003] ESPRIT METAL Project. [<http://www.cs.bris.ac.uk/~cgc/METAL>], 2003, Acessado em 07/05/2006.
- [Mitchell 1997] T. Mitchell. *Machine Learning*. MacGraw Hill, New York, NY, 1997.
- [Montgomery et. al 1990] D. Montgomery, L. Johnson, and J. Gardiner. *Forecasting & Time Series Analysis*. McGraw Hill, 1990.
- [Morettin, Toloí 1987] P. A. Morettin and C. M. Toloí. *Séries Temporais, Métodos Quantitativos*. Atual, São Paulo, SP, 1987.
- [Nabney 2002] I. T. Nabney. *Netlab: Algorithms for Pattern Recognition*. ser. Advances in pattern recognition. Springer, New York, NY, USA, 2002.
- [Norgaard 1997] M. Norgaard. Neural Network based system identification toolbox version 1.1 for use with Matlab. *Technical Report 97-E-851*, Department of Automation, Technical University of Denmark, 1997.
- [Oliveira 2000] E. M. J. Oliveira. Previsão da Série Temporal do Índice da Bolsa de Valores de São Paulo Usando Redes Neurais e Estatística. Dissertação de Mestrado, Centro de Informática, Universidade Federal de Pernambuco, Recife, Brasil, 2000.
- [Principe et. al 2002] J. Principe, N. Euliano, S. Garania. Principles and networks for self-organization in space–time. *Neural Networks*, 15: 1069–1083, 2002.
- [Prudêncio, Ludermir 2002] R. B. C. Prudêncio and T. B. Ludermir. Selection of models for time series prediction via meta-learning. *In Proceeding of the Second International Conference on Hybrid Systems (HIS 2002)*, pages 74-83, IOS Press, 2002.

- [Prudêncio, Ludermir 2003] R. B. C. Prudêncio and T. B. Ludermir. Selecting and ranking time series models using the NOEMON approach. *In Lecture Notes in Computer Science*, 2714: 654-661, 2003.
- [Prudêncio, Ludermir 2004] R. B. C. Prudêncio and T. B. Ludermir. Meta-learning approaches to selecting time series models. *Neurocomputing*, 61: 121-137, 2004.
- [Prudêncio 2004] R. B. C. Prudêncio. Meta-Aprendizado para Seleção e Combinação de Modelos de Séries Temporais. Tese de Doutorado, Centro de Informática, Universidade Federal de Pernambuco, Recife, Brasil, 2004.
- [Punronen, Terziyan 2000] S. Punronen, V. Terziyan. A similarity evaluation technique for data mining with an ensemble of classifiers. *11th International Workshop on Database and Expert Systems Applications (DEXA 2000)*, pages 1155-1159, London, UK, 2000.
- [Quinlan 1993] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, 1993.
- [Rojas 2004] H. P. Rojas. Soft-computing techniques for time series forecasting. *12th European Symposium On Artificial Neural Networks (ESANN)*, pages 93-102, Bruges, Belgium, 2004.
- [Rousseeuw 1984] P. J. Rousseeuw. Least Median of Squares Regression. *Journal of the American Statistical Association*, 79(388): 871-880, 1984.
- [Rumelhart et. al 1986] D. E. Rumelhart, G. E. Hinton and R. J. Williams. Learning representations by backpropagation errors. *Nature*, 323: 533-536, 1986.
- [Rumelhart, Zisper 1985] D. E. Rumelhart and D. Zisper. Feature Discovery by competitive learning. *Cognitive Science*, 9: 75-112, 1985.
- [Santos et. al 2004a] P. M. dos Santos, T. B. Ludermir and R. B. C. Prudêncio. Selection of Time Series Forecasting Models based on Performance Information. *In Proceedings of the Fourth International Conference on Hybrid Intelligent Systems (HIS 2004)*, pages 366-371, 2004.
- [Santos et. al 2004b] P. M. dos Santos, T. B. Ludermir e R. B. C. Prudêncio. Seleção de Modelos de Previsão de Séries Temporais baseada em Informações de Performance. *In Anais do VIII Simpósio Brasileiro de Redes Neurais (SBRN 2004)*, 2004.

-
- [Schmidhuber et. al 1996] J. Schmidhuber, J. Zhao and M. Wiering. Simple principles of meta-learning. *Technical Report IDSIA-69-96*, 1996.
- [Soares, Brazdil 2000] P. Brazdil and C. Soares. Zoomed Ranking: Selection of Classification Algorithms based on Relevant Performance Information. *Lecture Notes in Computer Science*, 1910: 126-135, 2000.
- [Tashman 2000] L. J. Tashman. Out-of-sample Tests of Forecasting Accuracy: An Analysis and Review. *International Journal of Forecasting*, 16: 437-450, 2000.
- [Todorovski, Dzeroski 1999] S. Todorovski and L. Dzeroski. Experiments in meta-level learning with ILP. *In Proceedings of the Third European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD99)*, pages 98-106, 1999.
- [Tsoumakas et. al 2004] G. Tsoumakas, D. Vrakas, N. Bassiliades and I. Vlahavas. Lazy Adaptive Multicriteria Planning. *In Proceedings of the 16th European Conference on Artificial Intelligence (ECAI04)*, pages 693-697, 2004
- [Tukey 1980] J. M. Tukey. Can we predict where 'time series' should go next? Directions in Time Series. (Eds. D. R. Brillinger and G. C. Tiao). *Institute of Mathematical Statistics*. Hayward, CA, pages 1-31, 1980.
- [Newman et. al 1998] D.J. Newman, S. Hettich, C.L. Blake and C.J. Merz. UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.
- [Valença 2005] M. J. S. Valença. *Aplicando Redes Neurais: um guia completo*. Ed. Do Autor, 2005.
- [Valença 2001] M. J. S. Valença and T. B. Ludermir. Redes Compostas por Blocos de Regressões Sigmóides Não-lineares: Uma eficiente rede de alta ordem com aplicações na previsão de séries temporais, *V Congresso Brasileiro de Redes Neurais (CBRN)*, 1: 67-72, 2001;
- [Vilalta, Drissi 2002] R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *Journal of Artificial Intelligence Review*. 18(2): 77-95, 2002.

- [Vesanto et. al 2000] J. Vesanto, J. Himberg, E. Alhoniemi and J. Parhankangas. SOM Toolbox for Matlab 5 (Technical Report). SOM Toolbox Team, Helsinki University of Technology.
- [Waibel et. al 1989] A. Waibel, T. Hanazawa, K. Shikano and K. Lang. Phonem recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(3): 328-339, 1989.
- [Wan 1993] E. A. Wan. Time series prediction by using a connectionist network with internal delay lines. *Time Series Prediction: Forecasting the Future and Understanding the Past*, A. S. Weigend and N. A. Gershenfeld, Eds., pages 195–217. Addison-Wesley Publishing Company, Inc, Reading, MA,1993.
- [Widrow, Hoff 1960] B. Widrow and M. E. Hoff. Adaptative switching circuits. *IRE WESCON Convention Records*, 96-104, 1960.
- [Zhang et. al 1998] G. P. Zhang, B. E. Patuwo, and M. Y. Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecast*, 14: 35-62, 1998.
- [Zhang et. al 2001] G. P. Zhang, B. E. Patuwo, and M. Y. Hu. A simulation study of artificial neural networks for nonlinear time series forecasting. *Computers and Operation Research*, 28(4): 381-396, 2001.