

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Previsão de Séries Temporais via Modelos Baseados em Regras

Cláudio Paulo Faustino

Itajubá, março de 2011

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Cláudio Paulo Faustino

Previsão de Séries Temporais via Modelos Baseados em Regras

Dissertação submetida ao Programa de Pós-Graduação em
Ciência e Tecnologia da Computação como parte dos requisitos
para obtenção do Título de Mestre em Ciência e Tecnologia
da Computação

Área de Concentração: Matemática da Computação

Orientador: Prof. Dr. Carlos Alberto M. Pinheiro

Março de 2011

Itajubá - MG

*Dedico este trabalho a meus pais, aqueles a quem simplesmente devo tudo que sou;
minha noiva Camila, aquela que mudou minha vida e apoiou-me muito nos momentos mais difíceis;
minhas amigas e segundas mães Dalva e Lili por cuidarem tão bem de mim;
aos colegas de república com quem passei muito bons momentos juntos.*

Da esperança nasce a Luz.

Sr. Madruga

Agradecimentos

Agradeço primeiramente à Deus, aquele que sempre ilumina nosso caminho e enche nossos corações de esperança e fé.

Agradeço a todos os professores do Instituto de Engenharia de Sistemas e Tecnologia da Informação da UNIFEI que, de forma direta ou indireta, contribuíram para meu crescimento humano e científico.

Agradeço ao meu professor, orientador e amigo Dr. Carlos Alberto M. Pinheiro que me conduziu até aqui.

Agradeço especialmente ao meu pai, a minha mãe e a minha noivinha, que estiveram junto a mim nos vários momentos difíceis dessa trajetória de estudos.

Resumo

Aplicações que envolvam previsões podem ser encontradas nos mais diversos campos do conhecimento. Frequentemente, áreas da engenharia, ciências naturais, ciências humanas, economia, finanças, dentre outras, vêm recorrendo a modelos de previsão para que informações relacionadas a estimativas de um sistema possam ser inferidas. Os processos de previsão muitas vezes estão relacionados ao conceito de séries temporais, estabelecendo as principais informações disponíveis para que modelos de previsões possam ser determinados. Este trabalho tem como objetivo a construção e a comparação de modelos que sejam capazes de descrever o comportamento de séries temporais e inferir valores futuros das mesmas. Em especial, é de interesse estudar uma classe especial de modelos, os *modelos baseados em regras*. Esses modelos estão fundamentados em conceitos da lógica nebulosa e dos conjuntos aproximados, consagradas ferramentas da inteligência artificial. Além dessas abordagens, outros métodos de previsão como as redes neurais artificiais (*RNA*) e os métodos estatísticos foram considerados. A finalidade dessas técnicas, comprovadamente eficientes em aplicações de previsão, é gerar modelos que servirão de base de comparação para os modelos baseados em regras, foco desse trabalho. Os modelos baseados em regras estão fundamentados a partir de modelos *neuro-fuzzy Adaptive Network Based Fuzzy Inference System (ANFIS)* e de uma proposta sugerida neste trabalho e que foi baseada em pesquisas recentes. Já para os modelos neurais foi considerada a aplicação de *RNA's* do tipo perceptrons de múltiplas camadas (*MLP*). Por fim, os modelos estatísticos foram obtidos por diversos métodos como o Auto Regressivo com Médias Móveis para séries não estacionárias (*ARIMA*) e o *Holt Winters* por exemplo. A fim de testar as metodologias empregadas neste trabalho, considerou-se a aplicação da previsão em cinco séries temporais práticas. Procurou-se selecionar séries que apresentassem variadas características e comportamentos para que a avaliação do desempenho dos modelos pudesse ser realizada nas mais diferentes situações. A avaliação de cada modelo foi realizada com base na análise dos vários indicadores de erros amplamente aplicados em trabalhos relacionados.

Abstract

Applications involving predictions can be found in various fields of knowledge. Often, areas of engineering, natural sciences, humanities, economics, finance, among others, have appealed to forecasting models in order to obtain relevant future informations about their processes. Forecast processes are often related to the concept of time series, which is the best set of data for forecasting model determination. This work aims at the construction and comparison of models that are able to describe behavior of the time series and to infer their future values. In particular, it is of interest to study a special class of models, the *rule-based* models. These models are based on concepts of fuzzy logic and rough sets, consecrated artificial intelligence tools. Besides these approaches, other forecasting methods as artificial neural networks (ANN) and statistical methods have been considered. The purpose of these techniques, proven to be effective in forecasting applications, is to generate models that serve as a basis of comparison for the rule-based models, focus of this work. Two techniques have been considered in order to obtain the rules-based models. The first one applies a hybrid architecture based on a neuro-fuzzy approach called *Adaptive Network Based Fuzzy Inference System (ANFIS)*. The second one consists of a proposal suggested in this work which was based on recent researches. The neural model chosen for this work was *multilayer perceptron (MLP)* because it's widely applied in forecasting applications. Finally, the statistical models were obtained by several methods such as Autoregressive integrated moving average (*ARIMA*) and the *Holt Winters*. In order to test the methodologies employed in this work, five time series were considered. Different time series with different characteristics and behaviors were selected. So, the model performance evaluation could be accomplished in many different situations. The evaluation of each model was based on analysis of various error indicators widely applied in works related.

Sumário

Lista de Figuras

Lista de Tabelas

1	Introdução	p. 16
1.1	Objetivos	p. 17
1.2	Justificativa	p. 18
1.3	Organização da Dissertação	p. 19
2	Séries Temporais e Métodos de Previsão	p. 21
2.1	Series Temporais	p. 21
2.2	Métodos Estatísticos de Previsão de Séries Temporais	p. 24
2.2.1	Método de Médias	p. 24
2.2.1.1	Método de Média Simples e Média Móvel	p. 25
2.2.1.2	Métodos Média Móvel Exponencial	p. 26
2.2.1.3	Método de Holt Winters	p. 27
2.2.2	Métodos de Decomposição	p. 28
2.2.2.1	Determinando a Componente de Tendência	p. 29
2.2.2.2	Cálculo dos Índices Sazonais	p. 30
2.2.3	Modelos Box-Jenkins	p. 31
2.2.3.1	Médias Móveis (MA)	p. 31
2.2.3.2	Auto Regressivo (AR)	p. 32
2.2.3.3	Auto-Regressivo com Médias Móveis (ARMA)	p. 32

2.2.3.4	Modelo ARIMA	p. 32
2.3	Métodos Não-Lineares de Previsão	p. 34
2.3.1	Redes Neurais Artificiais	p. 35
2.3.2	Tópicos Sobre <i>MLP</i>	p. 35
3	Previsão de Séries Temporais via Lógica Nebulosa	p. 38
3.1	Tópicos sobre Lógica Nebulosa	p. 38
3.2	Introdução a Sistemas Neuro-Fuzzy	p. 42
3.3	Trabalhos Relacionados	p. 44
3.4	Adaptive Network-Based Fuzzy Inference System	p. 46
3.4.1	Arquitetura do ANFIS	p. 47
3.4.2	Treinamento do <i>ANFIS</i>	p. 49
3.4.2.1	Aplicação do Método dos Mínimos Quadrados no Pro- cesso de Treinamento (<i>etapa “forward”</i>)	p. 50
3.4.2.2	Aplicação do <i>Backpropagation</i> no Processo de Treina- mento (<i>etapa “backward”</i>)	p. 51
3.5	O Algoritmo Fuzzy C-Means	p. 52
3.6	Metodologia de Aplicação da Lógica Nebulosa em Processos de Previsão	p. 54
4	Conjuntos Aproximados e Previsões	p. 58
4.1	Introdução a Conjuntos Aproximados	p. 58
4.2	Conceitos Fundamentais de Conjuntos Aproximados	p. 61
4.3	Aplicações de Conjuntos Aproximados	p. 64
4.4	Uma Proposta de Metodologia de Previsão Baseada em Conjuntos Apro- ximados	p. 68
4.4.1	Conjuntos Aproximados como Aproximadores de Funções	p. 68
4.4.2	Aplicação do Método em Processos de Previsão	p. 70
4.5	Considerações Sobre a Aplicação da Metodologia	p. 71

5	Metodologia de Processos de Previsão	p. 73
5.1	Definição dos conjuntos de treinamento e teste	p. 73
5.2	Normalização dos dados	p. 74
5.3	Pré Processamento dos Dados	p. 75
5.4	Configuração do sistema previsor a ser empregado	p. 76
5.5	Previsão e análise dos erros de previsão	p. 77
6	Experimentos	p. 79
6.1	Experimento 1 - Série de Consumo de Sacas de Café no Brasil	p. 80
6.1.1	Modelo Fuzzy Aplicado à Previsão da Série de Demanda de Café .	p. 81
6.1.2	Modelo Neural Aplicado à Previsão da Série de Consumo de Café	p. 83
6.1.3	Discussão dos Resultados (Série Consumo de Café)	p. 84
6.2	Experimento 2 - Série de Preço da Arroba de Boi	p. 85
6.2.1	Modelo Fuzzy Aplicado à Previsão da Série de Preço da Arroba de Boi	p. 86
6.2.2	Modelo Neural Aplicado à Previsão da Série de Preço da Arroba de Boi	p. 86
6.2.3	Discussão dos Resultados (Série de Preço da Arroba do Boi Gordo)	p. 87
6.3	Experimento 3 - Série de Demanda de Carga Elétrica	p. 88
6.3.1	Modelo Neural Aplicado à Previsão da Série de Demanda de Carga Elétrica	p. 89
6.3.2	Modelo Fuzzy Aplicado à Previsão da Série de Demanda de Carga Elétrica	p. 89
6.3.3	Modelagem via Conjuntos Aproximados Aplicada à Previsão da Série de Demanda de Carga Elétrica	p. 91
6.3.4	Previsão da Série de Demanda de Carga Elétrica via Método de <i>Holt Winters</i>	p. 93
6.3.5	Discussão dos Resultados (Série Demanda de Carga Elétrica) . . .	p. 94

6.4	Experimento 4 - Série de Atividade Pluviométrica	p. 95
6.4.1	Modelo Neural Aplicado à Previsão da Série de Atividade Pluviométrica	p. 96
6.4.1.1	Modelo Neural Aplicado à Previsão da Série de Atividade Pluviométrica com Informação Temporal	p. 96
6.4.1.2	Modelo Neural Aplicado à Previsão da Série de Atividade Pluviométrica sem Informação Temporal	p. 97
6.4.2	Modelo Fuzzy Aplicado à Previsão da Série de Atividade Pluviométrica	p. 98
6.4.3	Modelo Fuzzy Aplicado à Previsão da Série de Atividade Pluviométrica com Informação Temporal	p. 98
6.4.4	Modelo Fuzzy Aplicado à Previsão da Série de Atividade Pluviométrica sem Informação Temporal	p. 99
6.4.5	Modelagem via Conjuntos Aproximados Aplicada à Previsão da Série de Atividade Pluviométrica	p. 99
6.4.6	Previsão da Série de Atividade Pluviométrica via Método de <i>Holt Winters</i>	p. 101
6.4.7	Discussão dos Resultados (Série Atividade Pluviométrica)	p. 102
6.5	Experimento 5	p. 102
6.5.1	Modelos Fuzzy Aplicados na Previsão da Série de Fundo de Ações do Banco do Brasil	p. 103
6.5.2	Modelos Neurais Aplicados na Previsão da Série de Fundos de Ações do Banco do Brasil	p. 105
6.5.3	Discussão dos Resultados (Série Financeira)	p. 106
7	Conclusão	p. 108
7.1	Considerações finais	p. 108
7.2	Contribuições e Sugestões para trabalhos futuros	p. 109
	Apêndice A - Programa para tratamento das regras geradas pelo Rosetta	p. 111

Apêndice B - Programa de processamento de regras para simulação dos modelos aproximados	p. 117
Apêndice C - Programa para construção de modelos fuzzy para previsão	p. 122
Referências	p. 124

Lista de Figuras

1	Exemplo de representação gráfica de uma série temporal.	p. 22
2	Organização dos métodos clássicos para análise de séries temporais. . . .	p. 25
3	Arquitetura geral de um FIS.	p. 42
4	Exemplo de uma rede neural adaptativa.	p. 47
5	Arquitetura <i>ANFIS</i> de 5 camadas.	p. 48
6	Arquitetura de um <i>ANFIS</i> para série com quatro clusters.	p. 56
7	Diagrama de atividades da metodologia de previsão baseada em modelo fuzzy.	p. 57
8	Objetivo geral da teoria de <i>Conjuntos Aproximados</i>	p. 59
9	Esquema geral da metodologia proposta.	p. 71
10	Gráfico da série de consumo de café.	p. 81
11	Gráfico da adequação do modelo fuzzy aos dados de treinamento da série de consumo de café.	p. 83
12	Gráfico da série de preços da arroba do boi gordo.	p. 85
13	Estrutura dos modelos neurais hierárquicos empregados.	p. 90
14	Gráfico da previsão da série de demanda de carga elétrica via redes neurais.	p. 90
15	Estrutura do <i>ANFIS</i> aplicado no processo de previsão da série de de- manda de carga elétrica.	p. 91
16	Gráfico da adequação do modelo <i>fuzzy</i> aos dados de treinamento da série de demanda de carga elétrica.	p. 91
17	Gráfico da previsão da série de demanda de carga elétrica realizada pelo modelo baseado em regras <i>fuzzy</i>	p. 92

18	Gráfico da adequação do modelo aproximado aos dados de treinamento da série de demanda de carga elétrica.	p. 93
19	Gráfico da previsão da série de demanda de carga elétrica realizada pelo modelo aproximado.	p. 94
20	Gráfico da previsão da série de demanda de carga elétrica realizada pelo método de Holt Winters.	p. 94
21	Esquemática do sistema de monitoramento de enchentes.	p. 96
22	Gráfico da previsão da série de atividade pluviométrica realizada pelo modelo neural com informação temporal.	p. 97
23	Gráfico da previsão da série de atividade pluviométrica realizada pelo modelo neural sem informação temporal	p. 98
24	Gráfico da previsão da série de atividade pluviométrica realizada pelo modelo fuzzy com informação temporal	p. 99
25	Gráfico da previsão da série de atividade pluviométrica realizada pelo modelo fuzzy sem informação temporal	p. 100
26	Gráfico da modelagem da série de atividade pluviométrica realizada pelo modelo baseado em regras gerado via conjuntos aproximados	p. 100
27	Gráfico da previsão da série de atividade pluviométrica realizada pelo modelo baseado em regras gerado via conjuntos aproximados	p. 101
28	Gráfico da previsão da série de atividade pluviométrica realizada pelo método de Holt Winters.	p. 101
29	Gráfico da série de fundo de ações do Banco do Brasil.	p. 103
30	Gráfico da previsão da série de ações do Banco do Brasil (Normalização da diferença).	p. 104
31	Gráfico da previsão da série de ações do Banco do Brasil (Retornos). . .	p. 105
32	Gráfico da previsão da série de ações do Banco do Brasil via Modelo Neural (Normalização da diferença).	p. 106
33	Gráfico da previsão da série de ações do Banco do Brasil via Modelos Neurais (Retornos).	p. 106

Lista de Tabelas

1	Representação tabular de um sistema de informação.	p. 61
2	Representação de um <i>SD</i> para previsão de um passo à frente.	p. 70
3	Tabela com as observações da série de consumo de sacas de café.	p. 81
4	Antecedentes das regras do modelo fuzzy para série de consumo de café. .	p. 82
5	Consequentes das regras do modelo fuzzy para série de consumo de café.	p. 83
6	Tabela de erros da previsão da série de consumo de sacas de café.	p. 84
7	Tabela de erros da previsão da série de preços da arroba de boi.	p. 88
8	Regras do modelo aproximado para previsão de demanda de carga elétrica.	p. 93
9	Tabela de erros da previsão da série de demanda de carga elétrica.	p. 95
10	Tabela de erros da previsão da série de atividade pluviométrica.	p. 102
11	Tabela de erros da previsão da série financeira.	p. 107

Lista de Listagens

A.1	Classe principal do programa para tratamento das regras	p. 111
A.2	Classe principal do programa para tratamento das regras	p. 112
B.1	Listagem do programa de simulação de modelos aproximados	p. 117
C.1	Listagem do programa de para construção de modelos fuzzy para previsão de enchentes	p. 122

1 Introdução

Pode-se definir previsão como o processo através do qual o comportamento futuro de um sistema, seja ele dinâmico ou estático, é estimado. Tal estimativa é, na maioria das vezes, baseada no entendimento e caracterização do comportamento do sistema ao longo do tempo (CASTILHO; MELIN, 2002).

Esse entendimento e por conseguinte a caracterização de um sistema pode ser conseguido por meio de modelos gerados através das mais diversas técnicas. Desta forma, o desafio central da área da atividade de se prever é justamente escolher o método apropriado e aplicá-lo de forma adequada a fim de gerar modelos suficientemente eficientes que sejam capazes de descrever o comportamento do sistema em estudo.

Neste trabalho, o estudo de um determinado sistema é realizado por meio da análise da *série temporal* gerada a partir dos dados do mesmo. Uma série temporal, conforme a conceituação fornecida no Capítulo 2, pode ser definida como um conjunto de observações regularmente dispostas em um período de tempo passado do fenômeno em estudo.

Uma vez determinado, o modelo em questão pode ser usado para que valores futuros da série a qual ele está relacionado possam ser inferidos.

Vários métodos podem ser considerados para se gerar um modelo de previsão. Basicamente, os métodos de análise e previsão podem ser classificados em dois grandes grupos: “*Soft Computing*” e “*Hard Computing*”

Os métodos classificados como *Hard Computing* estão fortemente fundamentados em teorias estatísticas e uma discussão sobre vários desses métodos pode ser encontrada no Capítulo 2.

Os métodos classificados como *Soft Computing* são implementados através da aplicação de paradigmas da inteligência artificial e uma descrição dos métodos considerados nesse trabalho pode ser encontrada ao longo dos Capítulos 3 e 4.

1.1 Objetivos

Estabelecido o conceito de previsão, formaliza-se o objetivo geral do trabalho que é prever o comportamento futuro de uma série temporal.

Contudo, é de interesse que a previsão seja realizada a partir de modelos inseridos em uma classe especial de modelos gerados via métodos *Soft Computing*. Esses modelos são os *modelos baseados em regras*.

Sendo assim, o desafio principal é obter um conjunto de regras no formato *SE - ENTAO* que seja capaz de aproximar o comportamento de uma série temporal e que, a partir dessa aproximação, inferências de valores futuros da mesma sejam realizadas.

Os modelos baseados em regras tem uma característica importante em relação aos modelos gerados pelos métodos estatísticos e os modelos gerados via redes neurais artificiais (*RNA*). Essa característica consiste na facilidade com que o conhecimento embutido em cada modelo, necessário para o funcionamento do mesmo, pode ser interpretado. Por exemplo, o conhecimento das *RNA's* do tipo *Multilayer Perceptron* está abstraído nos pesos que unem os neurônios de uma camada com os neurônios de outra (HAYKIN, 2000) (ver Capítulo 2). Contudo, a interpretação do que cada peso representa em termos de conhecimento é muito complicada. Entretanto, esta condição é factível de ser interpretada em modelos baseados em regras visto que o conhecimento está todo abstraído em regras cuja sintaxe é muito próxima da linguagem humana. Essa característica confere a esses modelos a capacidade de serem facilmente inseridos em sistemas computacionais gerais.

Para gerar modelos baseados em regras, duas abordagens baseadas em duas reconhecidas ferramentas da inteligência artificial serão consideradas.

A primeira abordagem é baseada na *lógica nebulosa* (ZADEH, 1965) cuja aplicação em processos de previsão de séries temporais vem crescendo ao longo das últimas duas décadas principalmente depois de propostas de algoritmos como o de Wang e Mendel (WANG; MENDEL, 1992) e do *Adaptive Network Based Fuzzy Inference System* (JANG, 1993).

A segunda abordagem considera a aplicação dos conceitos de *conjuntos aproximados* (“*Rough Sets*”) (PAWLAK, 1982) cuja aplicação em processos de previsão de séries temporais raramente é considerada.

A fim de validar e avaliar as propostas de metodologias de previsão utilizadas neste trabalho, cinco séries temporais práticas relacionadas a diferentes áreas foram considera-

das. Cada série temporal apresenta características que lhe são peculiares, como tendência, sazonalidade, aleatoriedade, dentre outras que serão melhores discutidas no Capítulo 2. Pretende-se desta forma, avaliar o desempenho de cada modelo de previsão em séries com diferentes comportamentos inseridas nas mais diferentes áreas de conhecimento.

Para determinar o quão eficiente é um modelo, comparações com métodos de previsão consagrados como aqueles baseados em *Hard Computing* e redes neurais artificiais (*RNA*) serão realizadas.

Sendo assim, os objetivos do trabalho podem ser resumidos em duas perguntas sendo que a razão da segunda depende da resposta da primeira:

1. *Dada uma série temporal, existe um modelo baseado em regras que seja capaz de descrever, de forma eficaz, o comportamento passado da série e, uma vez aprendido esse comportamento, inferir valores futuros da mesma?*
2. Caso a resposta à primeira pergunta seja sim, pode-se então buscar uma resposta para a segunda questão: *Quão preciso é tal modelo baseado em regras?*

Conforme citado anteriormente, a resposta para a segunda questão pode ser obtida por meio de comparações com modelos consagrados de previsão.

1.2 Justificativa

A importância da previsão de séries temporais pode ser verificada a partir de uma análise, mesmo que superficial, de trabalhos relacionados a esta área.

Engenharia (LIAO; TSAO, 2007), ciências atmosféricas (NAYAK et al., 2004), ciências humanas (CHEN; YING; PAN, 2010a), agronomia (FAHIMIFARD et al., 2009), finanças (LI; XIONG, 2005), economia (TAY; SHEN, 2002), são algumas, dentre várias outras áreas, onde a aplicação de processos de previsão vem sendo empregada com sucesso.

Segundo Liao e Tsao (2007), a previsão de cargas elétricas desempenha um papel fundamental em sistemas de potência. De acordo com os autores, previsões eficientes têm significativo impacto na eficiência operacional dos sistemas de potência auxiliando em etapas como a programação anual de manutenção de equipamentos, gerenciamento da demanda de carga, controle de processos de avaliação de segurança entre outros. Além disso, para Syed-Ahmad et al. (2007) a previsão de demanda de carga elétrica auxilia

companhias elétricas na geração eficiente de energia elétrica bem como no fornecimento da mesma.

Em Hidrologia, as previsões de séries temporais vêm sendo empregadas em problemas relacionados ao fluxo de água em rios, ao controle de atividades pluviométricas, determinação de períodos de seca, planejamento da construção de reservatórios de água e estimação de concentração de sedimentos (FIRAT; GÜNGÖR, 2007). Para Nayak et al. (2004) prever variáveis hidrológicas representa um importante passo para o planejamento e análise operacional de recursos hídricos.

Na área das ciências humanas, frequentemente encontram-se trabalhos que abordam a previsão de chegada de turistas em uma determinada região. Segundo Chen, Ying e Pan (2010b), a necessidade de previsões precisas na área é crucial. Para os autores, tanto organizações não governamentais quanto companhias privadas buscam meios eficientes de previsão para que decisões operacionais, táticas ou estratégicas, possam ser mais eficientemente tomadas. Ainda segundo Chen, Ying e Pan (2010b), as empresas podem usar os resultados das técnicas de previsão para planejar a demanda de turistas e realizar investimentos em infraestrutura para garantir que toda a demanda seja atendida.

No mundo financeiro, previsões de preços dos mais variados produtos (CASTILHO; MELIN, 2002), de ações financeiras (LI; XIONG, 2005) e de variações na taxa de cambio são as mais frequentemente realizadas. A previsão de ações financeiras serve como ferramenta de apoio para investidores auxiliando no planejamento futuro de quais ações poderiam ser compradas ou vendidas. Saber o preço futuro de um determinado bem ou serviço é interesse de qualquer consumidor já que isso permitiria um planejamento mais preciso da aquisição do bem desejado. Estimar variações na taxa de cambio poderia ser uma ferramenta auxiliar no planejamento de importação e exportação de mercadorias.

Por fim, Ming-bao e Xin-ping (2008) defendem a importância da previsão do tráfego urbano de veículos. Para os autores, a previsão eficiente do fluxo de tráfego precisa ser realizada a fim de que medidas concretas no que diz respeito ao controle do mesmo possam ser tomadas.

1.3 Organização da Dissertação

Os próximos capítulos estão organizados conforme descrito a seguir.

O capítulo 2 define formalmente séries temporais, descreve os métodos estatísticos

mais frequentemente empregados em processos de previsão e discute a questão de se utilizarem métodos lineares para modelagem de processos não lineares. No final, uma rápida revisão sobre redes neurais artificiais é realizada.

O capítulo 3 faz uma revisão sobre conjuntos aproximados e propõe uma abordagem baseada na técnica, fundamentada sobre a proposta de Pinheiro et al. (2010a, 2010b), para que modelos de previsão de séries temporais possam ser gerados e aplicados.

No capítulo 4, pode-se encontrar uma revisão sobre lógica nebulosa e uma discussão de como modelos híbridos *neuro-fuzzy* podem ser aplicados em processos de previsão.

A metodologia empregada nos processos de previsão está descrita no capítulo 5. Neste capítulo, discutem-se aspectos gerais de processos de previsão bem como os procedimentos necessários para aplicação de cada método.

A descrição de cada uma das cinco séries temporais consideradas nesse trabalho pode ser encontrada no capítulo 6 juntamente com os experimentos realizados sobre cada uma delas.

Por fim, o capítulo 7 conclui a pesquisa e propõe sugestões para trabalhos futuros.

2 Séries Temporais e Métodos de Previsão

As séries temporais são os objetos de estudo principal deste trabalho. Deseja-se que, após aplicado um método de previsão, o comportamento de uma determinada série temporal em estudo possa ser reproduzido. Espera-se que a reprodução seja suficientemente precisa para que previsões de valores futuros possam ser realizadas respeitando-se limites de erros estipulados. Para conseguir atingir tal objetivo é importante estudar as características das séries temporais bem como os métodos de previsão disponíveis, de modo que a aplicação destes seja feita da forma mais eficiente possível.

Sendo assim, esse capítulo apresenta conceitos fundamentais de séries temporais e descreve brevemente os principais métodos quantitativos frequentemente empregados em processos de análise e previsão.

Em um segundo momento, discute-se como as técnicas de inteligência artificial podem ser empregadas em processos de previsão. Tal discussão foca o uso das *redes neurais artificiais*, meio utilizado para fins de comparação com os modelos baseados em lógica nebulosa e conjuntos aproximados propostos nesse trabalho. As principais abordagens de previsão desse trabalhos são descritas em capítulos específicos ao longo do texto.

2.1 Series Temporais

Uma série temporal pode ser matematicamente definida conforme a equação (2.1).

$$Z_t = \{Z_t \in \mathfrak{R} | t = 1, 2, 3 \dots N\} \quad (2.1)$$

Onde t é um índice temporal, e N é o número de observações.

De acordo com Palit e Popovic (2005) e a partir da análise da equação (2.1), uma série temporal pode ser definida como sendo um conjunto de observações discretas sendo que essas observações são realizadas em períodos geralmente equidistantes e apresentam

dependência serial entre si. Ainda segundo Palit e Popovic (2005), o principal objetivo do estudo de uma série temporal é permitir que a análise da dinâmica do sistema que gerou a série seja realizada, permitindo assim, reproduzi-lo ou mesmo prever seus valores futuros.

Graficamente, tem-se as observações de uma série dispostas no eixo das ordenadas. O eixo das abscissas fornece uma representação da noção temporal das observações. A Figura 1 ilustra graficamente uma série temporal financeira.

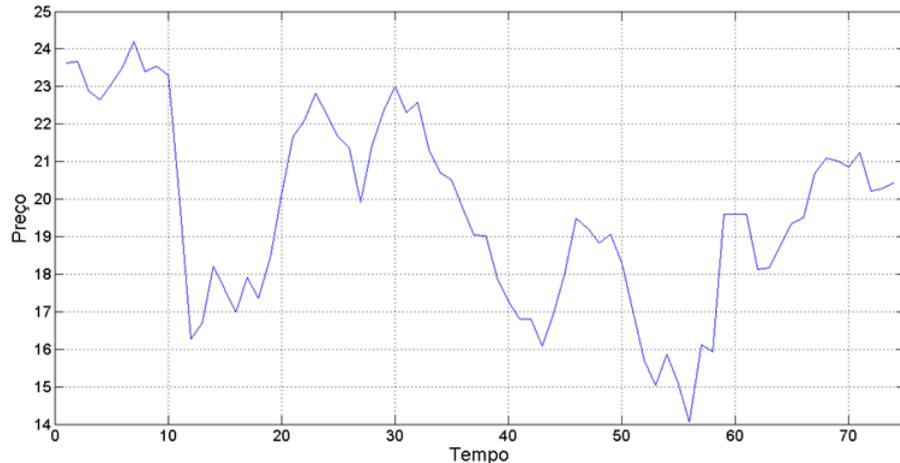


Figura 1: Exemplo de representação gráfica de uma série temporal.

Moreira (2004) afirma que uma série temporal pode ser composta por até quatro componentes distintas sendo que várias dessas componentes podem estar presentes ao mesmo tempo na caracterização da série. Segundo Bowerman, O'Connell e Koehler (2005) essas componentes são:

- *Componentes de tendência*: Confere à série a tendência de crescer ou decrescer ao longo do tempo. Pode ser identificada a partir de uma inspeção no gráfico da série. Caso um crescimento ou decrescimento acentuado seja verificado, a componente de tendência se faz presente e a série é dita ser não *estacionária*. Caso contrário, diz-se que ela é *estacionária*, ou seja, ela varia sempre em torno de um valor médio. Séries que representam o valor de ações financeiras em períodos de início ou fim de crise são exemplos de séries que apresentam forte componente de tendência.
- *Componentes cíclicas*: Provocam oscilações de subida e de queda nas séries de forma suave e repetitiva ao longo da componente de tendência. Geralmente, os efeitos cíclicos em uma série são causados por mudanças aleatórias nos fatores que influenciam a determinação das observações da série. Uma série temporal que representa o preço

de um determinado produto ao longo do tempo seria um exemplo de série com componente cíclica caso a influência do fator inflação fizesse-se presente.

- *Componentes sazonais*: São padrões de comportamento que se repetem ao longo das observações da série temporal em intervalos regulares de tempo. Diferentemente das componentes cíclicas, as sazonais possuem movimentos facilmente previsíveis. Exemplos de séries que possivelmente apresentariam acentuada componente sazonal são aquelas que caracterizam a demanda por carga elétrica ou a atividade pluviométrica de uma determinada região.
- *Componente de erro*: São variações ocorridas devido a causas não identificadas. Por ocorrerem ao acaso, não podem ser previstas. Moreira (2004) afirma ainda que, caso fosse retirado as três componentes descritas anteriormente, o que restaria seria a componente de erro.

Com base nas componentes anteriormente destacadas, Palit e Popovic (2005) sugerem que as séries temporais podem ser classificadas em:

- Estacionária ou não estacionária;
- Sazonal ou não sazonal;
- Linear ou não linear;
- Univariada ou multivariada;
- Caótica.

As séries não estacionárias, como discutido anteriormente, caracterizam-se pela presença da componente de tendência e, sendo assim, apresentam comportamento de crescimento ou queda. De forma oposta, nas séries estacionárias as observações variam em torno de uma média apresentando baixa variância.

As séries sazonais, como discutido anteriormente, apresentam padrões de comportamento semelhantes em períodos regulares de tempo, o que já não acontece com as não sazonais.

Uma série pode ser dita linear caso seja possível a regressão da mesma para uma função linear. A maioria das séries que descrevem fenômenos do mundo real não podem ser aproximadas por simples funções lineares e, desta forma, são ditas não lineares.

As séries univariadas são aquelas obtidas levando-se em consideração a análise de uma única variável de um contexto. Já as multivariadas, são séries geradas a partir da observação simultânea de duas ou mais variáveis.

Por fim, as séries caóticas são aquelas em que os dados parecem estar aleatoriamente distribuídos, sem periodicidade. Sua principal característica é não apresentar padrão definido das observações ao longo do tempo. Como exemplo de série caótica, pode-se citar a de *Mackey-Glass*, definida em sua forma discreta pela equação (2.2).

$$x(t) = 0.9x(t-1) + \frac{0.2x(t-\tau)}{(1+x(t-\tau))^{10}} \quad (2.2)$$

onde τ é um parâmetro que define o grau de caoticidade da série. Quanto maior o valor de τ para valores maiores de 17, mais caótica será a série.

2.2 Métodos Estatísticos de Previsão de Séries Temporais

Os métodos de modelagem de séries temporais podem ser usados para estimar valores futuros das mesmas. Geralmente, tais métodos podem ser aplicados quando dispõe-se de um histórico da variável do sistema a ser analisado. Além disso, o uso de tais métodos também é aconselhado quando um modelo de regressão para um determinado sistema é difícil de ser estimado ou não gera resultados bons o suficiente para serem aproveitados (BOWERMAN; O'CONNELL; KOEHLER, 2005).

Os métodos de séries temporais podem ser divididos em: métodos baseados em médias, métodos *Naive*, métodos baseados em decomposição e os métodos *Box-Jenkins*.

A Figura 2 ilustra uma organização dos métodos clássicos.

As seções seguintes estão dedicadas a descrever cada um dos métodos citando suas características, vantagens, desvantagens e os mecanismos matemáticos necessários para sua aplicação.

2.2.1 Método de Médias

A previsão obtida através do uso desses métodos utiliza-se de algum tipo de média que leva em consideração valores reais anteriores da própria variável que se quer prever (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998).

Dentre os métodos de médias mais comuns, encontram-se os métodos de Média Móvel

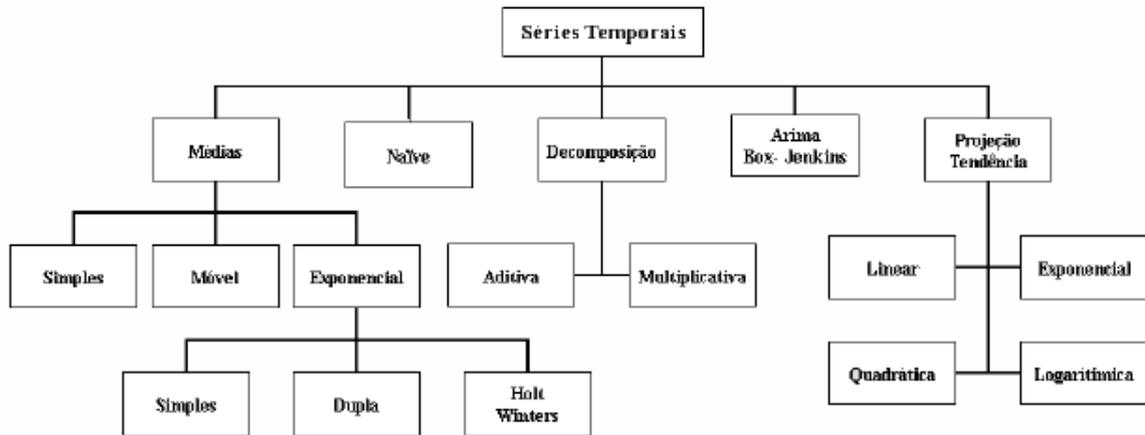


Figura 2: Organização dos métodos clássicos para análise de séries temporais.

Simple, Média Móvel Ponderada e Média Móvel Exponencial, sendo esta última dividida em Simple Suavização Exponencial, Dupla Suavização Exponencial e Métodos de *Holt Winters*.

2.2.1.1 Método de Média Simple e Média Móvel

O método de Média Móvel Simple prevê um valor para o período imediatamente futuro de t , tomando-se a média aritmética dos n valores passados observados (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998).

O método pode ser aplicado a partir da fórmula (2.3)

$$F_t = \frac{A_{t-1} + A_{t-2} + \dots + A_{t-n}}{n} \quad (2.3)$$

onde F_t é o valor previsto no tempo t , A_{t-i} ($i \leq n$) é o valor observado no tempo $t - i$ e n é o número de períodos considerados na média.

Segundo Moreira (2004), o método de Média Móvel Simple tem a vantagem de ser simples de ser implementado e de apresentar bons resultados quando a série apresenta poucas observações. No entanto, o resultado para séries não estacionárias ou sazonais não apresenta uma precisão satisfatória. Além disso, a dificuldade da determinação de um valor adequado para n é proporcional ao número de observações da série (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998).

De forma análoga ao método Média Móvel Simple, o método Média Móvel Ponderada toma as últimas observações da série para definição do modelo de previsão. A diferença

entre os métodos é a presença de pesos atribuídos às observações consideradas para previsão. Desta forma, pode-se atribuir maior importância aos valores mais recentes da série (MOREIRA, 2004).

O método Média Móvel Ponderada pode ser usado a partir da equação (2.4).

$$F_t = W_{t-1}A_{t-1} + W_{t-2}A_{t-2} + \dots + W_{t-n}A_{t-n} \quad (2.4)$$

onde W_i para $i \leq n$ são os pesos atribuídos aos valores da série.

A maior dificuldade na aplicação do método é determinação dos valores dos pesos. De forma análoga ao método Média Móvel Simples, a presença de componentes de tendência e sazonais inviabiliza o uso destes métodos (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998).

2.2.1.2 Métodos Média Móvel Exponencial

De acordo com a Figura 2, na classe de métodos de média móvel exponencial podem ser encontrados os métodos Simples Suavização Exponencial e Dupla Suavização Exponencial.

O método Simples Suavização Exponencial pode ser aplicado a partir da equação (2.5).

$$F_{t+1} = F_t + \alpha(Y_t - F_t) \quad (2.5)$$

Onde F_{t+1} é a previsão em $t + 1$, F_t é a previsão em t , Y_t é a observação em t e α é uma constante de suavização cujo valor geralmente varia em torno do intervalo de 0 a 1, e que é usada para quantificar a importância que será dada ao erro obtido na previsão anterior (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998).

Uma das maiores dificuldades desse método é estimar o valor de α . Geralmente, o uso de programas de computador como ferramenta de apoio é considerado para gerar modelos de previsão com vários valores para α escolhendo-se ao final, o valor que minimiza o erro ao máximo.

Este método, igualmente aos métodos Média Móvel Simples e Média Móvel Ponderada, não considera as componentes de tendência e de sazonalidade e deve ser utilizado para prever valores a curto prazo (BOWERMAN; O'CONNELL; KOEHLER, 2005).

Já o método Dupla Suavização Exponencial, diferentemente dos métodos anterior-

mente descritos, incorpora a componente de tendência no cálculo da previsão e pode ser utilizado para o cálculo de previsões a médio prazo (BOWERMAN; O'CONNELL; KOEHLER, 2005).

O método Dupla Suavização Exponencial pode ser utilizado a partir da aplicação das equações (2.6), (2.7) e (2.8).

$$Y_t = \alpha R_t + (1 - \alpha)(Y_{t-1} - T_{t-1}) \quad (2.6)$$

$$T_t = \delta(Y_t - Y_{t-1}) + (1 - \delta)T_{t-1} \quad (2.7)$$

$$Y_{t+m} = Y_t + T_t m \quad (2.8)$$

Onde Y_t é a previsão suavizada em t , R_t é observação em t , T_t é a estimativa da tendência em t e δ é a constante de suavização de 2ª ordem cujo valor varia de 0 a 1.

De forma análoga ao método Simples Suavização Exponencial, a determinação das constantes de suavização do método geralmente é feita por meio de testes. Seleciona-se os valores de α e δ que, em conjunto, minimizam o erro da previsão. Na maioria dos casos, essa estimativa é feita por meios de recursos computacionais.

2.2.1.3 Método de Holt Winters

O método de *Holt Winters* é uma extensão do método Dupla Suavização Exponencial. Além de fazer a suavização da média (equação 2.9) e da tendência (equação 2.10), este método também considera a componente de sazonalidade sendo que uma suavização da mesma também é realizada (equação 2.11) (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998).

O método é composto pelas equações (2.9), (2.10), (2.11) e (2.12) apresentadas a seguir.

$$Y_t = \alpha \frac{\bar{Y}_t}{S_{t-s}} + (1 - \alpha)(Y_{t-1} + T_{t-1}) \quad (2.9)$$

$$T_t = \delta(Y_t - Y_{t-1}) + (1 - \delta)T_{t-1} \quad (2.10)$$

$$S_t = \beta \frac{\bar{Y}_t}{T_t} + (1 - \beta)S_{t-s} \quad (2.11)$$

$$Y_{t+m}^{\hat{}} = (Y_t + L_t m)S_{t-s+m} \quad (2.12)$$

Sendo que Y_t é o resultado da previsão, T_t é a tendência, S_t é a componente sazonal, \bar{Y}_t é

o valor observado em t , \hat{Y}_{t+m} é a previsão m períodos a frente, L_t é a média dos valores até o instante t e s corresponde à duração da sazonalidade.

Para a inicialização do método, faz-se necessário que se tenha um valor inicial de Y_t (geralmente o 1º valor observado da série), da tendência T_t (geralmente o 1º valor observado da série) e do índice sazonal S_t (média da primeira sessão de dados) (BOWERMAN; O'CONNELL; KOEHLER, 2005).

De forma análoga ao método de Dupla Suavização Exponencial, o processo de determinação dos parâmetros α , β e δ do método de *Holt Winters* deve ser realizada por meio de testes sucessivos apoiados por meios computacionais a fim de encontrar uma configuração ótima que minimize o erro da previsão o máximo possível.

2.2.2 Métodos de Decomposição

Nos métodos de decomposição, as séries são analisadas como sendo compostas pelas quatro componentes definidas na seção 2.1

Segundo Moreira (2004), a idéia principal dos métodos de decomposição é isolar as componentes e tratá-las separadamente.

O método pode ser descrito pela equação (2.13)

$$Y = (T) \cdot (S) \cdot (C) \cdot (i) \quad (2.13)$$

onde Y é o valor da previsão, T é a componente de tendência, S é a componente de sazonalidade, C é a componente cíclica e i é a componente de erro.

A fim de simplificar o método, geralmente adota-se $C = 1$. Tal suposição pode ser feita desde que se admita que o horizonte de previsão seja curto. Desta forma, as variações cíclicas que possam vir a surgir ao longo de um período ainda estariam presentes no período em que a previsão está sendo realizada. Além disso, é comum unir as variações irregulares e a componente sazonal em uma única componente (MOREIRA, 2004). Sendo assim, a equação (2.13) pode ser reduzida à equação (2.14).

$$Y = (T) \cdot (S) \quad (2.14)$$

Dessa forma o problema fica reduzido em determinar a componente de tendência e de sazonalidade.

2.2.2.1 Determinando a Componente de Tendência

A determinação da componente de tendência pode ser feita a partir da regressão da função responsável pela geração dos dados da série para uma determinada função $y = f(x)$ (MOREIRA, 2004).

Tal função $f(x)$ pode assumir uma forma linear

$$y = a + bx \quad (2.15)$$

parabólica

$$y = a + bx + cx^2 \quad (2.16)$$

ou exponencial

$$y = ab^x \quad (2.17)$$

onde a , b e c são constantes e devem ser determinadas.

Se a equação escolhida tiver a forma da equação (2.15), o método de regressão é denominado de *Método de Regressão Linear*, caso contrário, o método é denominado de *Método de Regressão não Linear com Função Parabólica ou Exponencial*, de acordo com as equações (2.16) e (2.17), respectivamente.

Deve-se portanto, escolher uma dentre as três funções acima apresentadas que melhor represente o comportamento dos dados da série ao longo do tempo. O comportamento pode ser identificado com relativa facilidade por meio de inspeção do gráfico da série.

Uma vez definida qual função melhor se adapta ao comportamento da série, deve-se estimar os parâmetros que compõem a função.

O método tradicionalmente usado para estimar tais parâmetros é o método dos *Mínimos Quadrados*. Sem entrar em maiores detalhes, é apresentado a seguir as equações obtidas a partir da aplicação do método para cada uma das situações representadas pelas equações (2.15), (2.16) e (2.17).

No caso da regressão linear, tem-se as seguintes equações:

$$\sum Y = na + b \sum X \quad (2.18)$$

$$\sum XY = a \sum X + b \sum X^2 \quad (2.19)$$

Sendo que Y corresponde ao conjunto dos valores da variável dependente, X ao conjunto dos valores da variável independente e a e b são os parâmetros a serem estimados. Para fins

de simplificação, os índices dos somatórios foram omitidos. Vale ressaltar que a aplicação do tal método leva a um sistema composto por duas equações.

No caso da regressão não linear com função parabólica, as seguintes equações são obtidas a partir da aplicação do método dos mínimos quadrados:

$$\sum Y = na + b \sum X + c \sum X^2 \quad (2.20)$$

$$\sum XY = a \sum X + b \sum X^2 + c \sum X^3 \quad (2.21)$$

$$\sum X^2Y = a \sum X^2 + b \sum X^3 + c \sum X^4 \quad (2.22)$$

No caso da regressão não linear com função exponencial, a ideia é reduzir a equação (2.17) para a equação (2.15), aplicando-se o logaritmo em ambos os lados da equação (2.17). Este procedimento gera a equação (2.23)

$$\log_{10} Y = \log_{10} a + X \log_{10} b \quad (2.23)$$

que pode ser escrita na forma

$$Y' = A + BX' \quad (2.24)$$

onde $\log_{10} Y = Y'$, $\log_{10} a = A$ e $\log_{10} b = B$.

A partir deste ponto, pode-se estimar os parâmetros A e B usando as equações estimadas para regressão linear (equações 2.18 e 2.19) e, posteriormente, realizar as operações inversas: $a = 10^A$ e $b = 10^B$ para se determinar os parâmetros a e b necessários.

2.2.2.2 Cálculo dos Índices Sazonais

Uma vez calculada a linha de tendência através dos métodos discutidos na seção anterior, pode-se utilizar a equação da componente de tendência para estimar os valores da série.

Uma vez estimados esses valores, os mesmos devem ser ajustados a fim de que os efeitos sazonais sejam considerados. Para tanto, deve-se calcular para todos os k períodos da série, a razão entre o valor real observado Y_t e o valor T_t estimado pela equação da componente de tendência, gerando um valor R_t para a observação no período t .

Para estimar a influência da componente sazonal, deve-se identificar um numero n que

represente o intervalo de períodos da série onde o efeito da sazonalidade se faz presente, por exemplo, em cada trimestre de um ano (nesse caso $n = 4$), ou em cada dia de uma semana ($n = 7$), ou ainda, em cada mês de um ano ($n = 12$). Uma vez identificado o intervalo, calcula-se a média aritmética entre os R_{i-n} valores ($i = 1, 2, 3, \dots$ e $in \leq k$ onde k é o valor total de observações). Tem-se ao final desse processo, um conjunto A de n fatores que servirão como ajuste para os dados da previsão.

Estes fatores de ajuste serão utilizados na previsão futura. O resultado final da previsão inserido no i -ésimo período sazonal é dado pelo produto do valor obtido através da equação que representa a componente de tendência e do i -ésimo fator de sazonalidade do conjunto A .

2.2.3 Modelos Box-Jenkins

Segundo Bowerman, O'Connell e Koehler (2005), dentro das classes de métodos *Box-Jenkins*, os seguintes métodos podem ser encontrados: Auto Regressivo (AR), Médias Móveis (MA), Auto Regressivo com Médias Móveis (ARMA) e Auto Regressivo com Médias Móveis para processos não estacionários (ARIMA). Ainda de acordo com os autores, os três primeiros métodos anteriormente citados devem ser aplicados em sistemas lineares cuja série gerada apresenta comportamento estacionário. O último, pode ser aplicado em séries com acentuada componente de tendência.

Cada um dos métodos anteriormente citados serão discutidos nas seções que se seguem. A ideia principal é apresentar o processo matemático básico envolvido em cada um dos métodos sem, no entanto, entrar em maiores detalhes estatísticos de sua concepção ou de como a validação do mesmo pode ser realizada.

2.2.3.1 Médias Móveis (MA)

Segundo Mills (1999), um modelo de Médias Móveis pode ser descrito pela equação (2.27).

$$Y_t = \mu + \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} - \dots - \theta_q\varepsilon_{t-q} \quad (2.25)$$

Onde ε_{t-i} ($i = 0, 1, 2, \dots, q$) são perturbações aleatórias (erros gerados por eventos do i -ésimo período); θ_i ($i = 1, 2, \dots, q$) são os parâmetros do modelo a ser estimado; μ é a média da série e q é a ordem do modelo.

O valor de q especifica quantos parâmetros de θ deverão ser estimados, ou seja, determina o número de amostras do passado que deverão ser considerados (BOWERMAN; O'CONNELL; KOEHLER, 2005).

2.2.3.2 Auto Regressivo (AR)

Segundo Mills (1999), um modelo Auto-Regressivo pode ser descrito pela equação (2.26).

$$Y_t = c + \varepsilon_t + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} \quad (2.26)$$

Onde Y_{t-i} ($i = 1, 2, \dots, p$) são observações no instante i (muitas vezes denominados de regressores); ϕ_i ($i = 1, 2, \dots, p$) são os parâmetros do modelo a ser estimados; c é uma constante (muitas vezes omitida para fins de simplificação) e ε_t é um resíduo (BOWERMAN; O'CONNELL; KOEHLER, 2005).

2.2.3.3 Auto-Regressivo com Médias Móveis (ARMA)

A combinação dos dois modelos apresentados nas duas seções anteriores constitui o modelo Auto-Regressivo com Médias Móveis. O modelo é geralmente definido como $ARMA(p, q)$ onde p e q são as ordens do modelo auto regressivo e de médias móveis respectivamente (MILLS, 1999). Um modelo $ARMA$ é definido pela equação 2.27.

$$Y_t = c + \varepsilon_t + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (2.27)$$

2.2.3.4 Modelo ARIMA

Um modelo $ARIMA$ difere dos modelos anteriormente descritos por possibilitar trabalhar com séries não estacionárias (MILLS, 1999). Pode-se considerar este modelo como uma extensão do modelo $ARMA$ (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998).

O modelo $ARIMA$ pode ser reduzido para um modelo $ARMA$ a partir da transformação da série não estacionária que se está analisando para uma série estacionária (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998).

Esta transformação é realizada fazendo-se uma diferenciação na série original. A diferenciação calcula para cada instante t , um novo valor que é resultante da subtração

do t -ésimo valor da série com o t -ésimo - 1 valor da série (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998). A equação (2.28) ilustra a operação.

$$Z_t = \Delta Y_t = Y_t - Y_{t-1} \quad (2.28)$$

A partir da equação (2.28) pode-se verificar, que a nova série resultante Z_t terá o número de observações da série original subtraído de 1 (um).

O processo de diferenciação é realizado até que a série seja considerada estacionária. Desta forma, em muitas situações mais de uma diferenciação pode se fazer necessária para que a componente de tendência seja totalmente removida.

Segundo (MILLS, 1999), basicamente dois testes podem ser realizados para verificar se uma série é estacionária ou não. Um primeiro, simples e eficiente, consiste em plotar as observações da série em um gráfico e analisar sua componente de tendência procurando identificar uma inclinação positiva ou negativa da mesma.

Um teste mais consistente pode ser realizado analisando-se o gráfico da função de autocorrelação da série. Os valores das autocorrelações devem se aproximar estatisticamente de zero. O teste de *Barlett* pode ser usado para verificar se um valor de correlação pode ser ou não considerado estatisticamente igual a zero. Tal teste consiste em verificar se cada valor de correlação é menor que um dado valor x , obtido pela equação a seguir

$$x = \frac{2}{\sqrt{n}} \quad (2.29)$$

onde n é o número de observações da série.

Um modelo *ARIMA* é frequentemente enunciado da seguinte forma: *ARIMA* (p, d, q) (MILLS, 1999). Tais índices correspondem respectivamente ao parâmetro do modelo de auto regressão, ao fator de integração (numero de diferenciações) e ao parâmetro do modelo de médias.

Segundo (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998) certas heurísticas para a determinação dos parâmetros p e q podem ser adotadas, contudo, ainda segundo o mesmo autor, nenhum método é definitivo. Essas heurísticas consistem em “dicas” que podem conduzir a uma escolha acertada ou pelo menos próxima da correta para os parâmetros do modelo. Essas heurísticas envolvem complexas teorias estatísticas que fogem do escopo desse trabalho. Frequentemente, a determinação dos parâmetros é apoiada por recursos computacionais. O software *Minitab* por exemplo, é capaz de gerar a equação de um mo-

delo *ARIMA* desde que os parâmetros p , q e d e as observações da série sejam informados. Pode-se, desta forma, estimar vários modelos *ARIMA* apenas variando-se os parâmetros que os definem. Em uma etapa final, seleciona-se aquele modelo que resultou em menor erro.

2.3 Métodos Não-Lineares de Previsão

Os métodos de análise e previsão descritos nas seções anteriores são frequentemente empregados nos mais variados processos de previsão. Porém, tais métodos geram modelos que se enquadram em uma classe denominada de *Modelos Lineares*. Segundo Zhang, Patuwo e Hu (1998) esses modelos têm a vantagem de serem relativamente fáceis de se gerar e entender, contudo, apresentam a desvantagem de não serem apropriados para modelagem de sistemas não lineares.

Frequentemente, os processos que geram as observações das séries temporais são não lineares e, sendo assim, segundo Fahimifard et al. (2009), a aplicação de métodos lineares em processos de previsão pode gerar resultados não satisfatórios.

A abordagem mais indicada para o tratamento desse tipo de problema seria, portanto, considerar o emprego de métodos não lineares de previsão.

Muitos métodos não lineares de previsão, ainda inseridos no campo da estatística, foram propostos. Como exemplo, pode-se citar o *TAR* (Threshold autoregressive) (TSAY, 1989), *Markov Switching* (MCCULLOCH; TSAY, 1994) e o *GARCH* (Auto regressive conditional heteroscedastic) (HENTSCHEL, 1998). Contudo, segundo Zhang, Patuwo e Hu (1998) esses modelos não lineares apresentam problemas de precisão quando a quantidade de dados disponíveis é relativamente pequena.

A fim de usar procedimentos não lineares em processos de previsão, a comunidade científica frequentemente recorre à técnicas de inteligência artificial. As diversas arquiteturas de redes neurais artificiais (*RNA*) e a lógica nebulosa são as ferramentas mais frequentemente empregadas e serão consideradas nesse trabalho juntamente com uma proposta baseada em conjuntos aproximados.

Para fins de contextualização, esta seção apresenta tópicos básicos sobre as redes neurais do tipo *MLP* que, nesse trabalho, serviram de base de comparação para os modelos baseados em lógica *fuzzy* e *conjuntos aproximados*, focos principais da pesquisa.

2.3.1 Redes Neurais Artificiais

As redes neurais artificiais têm sua fundamentação baseada no modelo neural biológico. A idéia é criar um modelo matemático que represente, de forma simplificada, o funcionamento das redes neuronais biológicas (HAYKIN, 2000).

Segundo Haykin (2000) algumas das áreas mais comuns onde as redes neurais vêm sendo intensamente aplicadas são: reconhecimento de padrões, classificação de dados, predição, aproximação de funções, processamento de sinais e análise de imagem e voz.

Para resolver problemas do mundo real, várias arquiteturas de redes neurais foram propostas ao longo das ultimas três décadas. Cada uma dessas arquiteturas pode ser utilizada para resolver determinados tipos de problemas sendo que algumas arquiteturas se adequam mais precisamente a certos tipos de problemas do que outras. As arquiteturas mais comuns são as *Multilayer Perceptron* (*MLP*) e as *Self-Organization Map* (*SOM*) sendo a primeira amplamente empregada em processos de previsão. Uma outra arquitetura cuja aplicação em previsão é crescente é a *Support Vector Machine* (*SVM*) ((FEI; LIU; MIAO, 2009) e (TAY; CAO, 2001)).

Por ser a arquitetura aplicada nos testes desse trabalho, esta seção apresenta conceitos fundamentais relacionados às redes *MLP*.

2.3.2 Tópicos Sobre MLP

A estrutura de uma *MLP* compõe-se de uma camada de entrada, uma camada de saída e uma ou mais camadas intermediárias (também conhecidas como camadas ocultas)(HAYKIN, 1994). As camadas estão conectadas entre si por meio de pesos sinápticos. Tais conexões se dão camada a camada, ou seja, uma determinada camada está ligada através de pesos somente às camadas imediatamente anterior e posterior. Os pesos armazenam o conhecimento da rede neural. Sendo assim, o objetivo principal é ajustar os pesos de forma que a rede responda corretamente à estímulos a ela apresentados (ZHANG; PATUWO; HU, 1998). Esse tipo de rede é conhecido como rede *feedforward*.

Cada camada da rede neural é composta por neurônios. A saída de cada neurônio (exceto se este for da camada de entrada) é ativada através de uma função. Tal função é denominada de função de ativação e pode assumir diversas formas.

As funções de ativação mais usadas são as funções lineares, logística sigmoidal e tangente sigmoidal, descritas pelas equações (2.30), (2.31) e (2.32), respectivamente.

$$s_i = P'(E_i) = bE_i \quad (2.30)$$

$$s_i = P'(E_i) = \frac{1}{1 + e^{-E_i}} \quad (2.31)$$

$$s_i = P'(E_i) = \tanh(E_i) = \frac{e^{E_i+b} - e^{-(E_i+b)}}{e^{E_i+b} + e^{-(E_i+b)}} \quad (2.32)$$

Onde E_i é dado por

$$E_i = \sum_j w_{ij} s_j \quad (2.33)$$

sendo que w_{ij} é o peso entre o j -ésimo neurônio da camada anterior com o neurônio cujo valor de saída deseja-se calcular.

O ajuste dos pesos de uma rede neural se faz por meio de algoritmos de treinamento. Tais algoritmos podem ser classificados como algoritmos supervisionados e não supervisionados (HAYKIN, 2000).

Basicamente, um algoritmo supervisionado é aquele que simula um procedimento de aprendizado. Neste tipo de algoritmo, o treinamento é realizado tomando-se vários vetores de entrada e seus respectivos vetores de saída. Para cada vetor de entrada apresentado à rede, um vetor resposta é calculado. O vetor resposta é comparado com o vetor saída correspondente ao vetor de entrada apresentado para treinamento e um erro é calculado. Com base neste erro, ajustam-se os pesos da rede de forma a minimizar o erro o máximo possível (HAYKIN, 1994). Uma *época* de treinamento é completada quando todos os pares de vetores são apresentados à rede. Um processo completo de treinamento pode ser composto por tantas épocas quantas forem necessárias para que o erro atinja uma tolerância previamente determinada ou até que o número de épocas definido a priori seja alcançado (HAYKIN, 1994).

Um algoritmo não supervisionado, diferentemente do supervisionado, não apresenta a ideia de supervisão no processo de treinamento e, por isso, somente os vetores de entrada são necessários para treinamento. A ideia é ajustar os pesos da rede de forma que entradas semelhantes gerem respostas semelhantes após serem processadas pela rede (HAYKIN, 1994).

O algoritmo de treinamento mais popular de uma rede *MLP* é o “*Backpropagation*”, que é classificado como um algoritmo supervisionado. Tal algoritmo é composto por duas fases. Na primeira fase, (conhecida como “*forward*”), as entradas são apresentadas e propagadas camada a camada. Tem-se ao final do processo, um valor calculado para

cada neurônio na camada de saída. Com base nestes valores, um erro pode ser calculado a partir da equação (2.34) (HAYKIN, 2000).

$$Erro = \sum_{i=1}^J (s_i^* - s_i^{(N)})^2 \quad (2.34)$$

Onde J é o número de neurônios na camada de saída, s_i^* é o valor desejado para o i -ésimo neurônio da camada de saída e $s_i^{(N)}$ é o valor obtido para o i -ésimo neurônio na camada de saída (o superescrito (N) representa a camada cujos valores dos neurônios estão sendo considerados, neste caso, N diz respeito à camada de saída).

Na segunda fase (conhecida como “*backward*”), o erro calculado é propagado da camada de saída para as camadas anteriores por meio do cálculo do gradiente descendente. Este cálculo direciona à determinação de um valor de ajuste para os pesos da rede (HAYKIN, 2000). Desta forma, para a camada de saída tem-se:

$$\delta_i^{(N)} = (s_i^* - s_i^{(N)})P'(E_i^{(N)}) \quad (2.35)$$

onde o $P'(E_i^{(N)})$ é a derivada da função de ativação do neurônio.

E para as camadas anteriores, tem-se (2.36).

$$\delta_i^{(n)} = \sum_j \delta_j^{(n+1)} w_{ji}^n P'(E_i^{(n)}) \quad (2.36)$$

Assim, os pesos da rede são modificados com base no valor Δw_{ij} calculado em função do gradiente conforme visto na equação (2.37)

$$\Delta w_{ij} = \eta \cdot \delta_i \cdot s_j \quad (2.37)$$

onde η é a taxa de aprendizado do treinamento.

O fim da etapa *backward* marcará o fim de uma época e o início de um outra etapa *forward* que definirá o início de uma outra época de treinamento.

3 Previsão de Séries Temporais via Lógica Nebulosa

Este capítulo tem como objetivo introduzir conceitos básicos sobre lógica nebulosa, essenciais para o entendimento dos sistemas *neuro-fuzzy* frequentemente empregados nas previsões desse trabalho.

Uma vez que certos conceitos fundamentais forem apresentados, uma breve revisão sobre sistemas *neuro-fuzzy* e sobre trabalhos que utilizam esses sistemas para previsão é realizada nas seções 3.2 e 3.3 respectivamente.

As seções 3.4 e 3.5 descrevem, respectivamente, o *Adaptive Network Based Fuzzy Inference System (ANFIS)* e o algoritmo de agrupamento de dados *Fuzzy C-Means* que, em conjunto, compõem o cerne da metodologia de previsão baseada em lógica nebulosa aplicada nos experimentos desse trabalho (seção 3.6).

3.1 Tópicos sobre Lógica Nebulosa

Na teoria clássica dos conjuntos, o conceito de pertinência de um elemento x a um dado conjunto A em um universo de discurso X (que relaciona um conjunto de valores permitidos para o elemento x em questão) associa a cada elemento deste universo X um valor binário que determina se o elemento x pertence ou não pertence àquele conjunto A . Esse procedimento pode ser representado pela função característica f_A definida pela equação 3.1.

$$f_A(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases} \quad (3.1)$$

Contudo, no mundo real existem informações que são vagas, incertas ou imprecisas e, portanto, impossíveis de serem caracterizadas por predicados da lógica clássica. A teoria

da lógica nebulosa (ou lógica *fuzzy*) foi criada justamente para lidar com esse tipo de informação, tratando graus de pertinência intermediários entre a pertinência total e a não-pertinência de elementos de um universo de discurso com relação a um dado conjunto (ZADEH, 1965). Na lógica nebulosa, a função característica $f_A(x)$ foi generalizada de forma que ela pudesse assumir um número infinito de valores no intervalo $[0,1]$. Dessa forma, um conjunto nebuloso A em um universo X pode ser caracterizado por uma função de pertinência $\mu_A(x) : X \rightarrow [0, 1]$, onde $\mu_A(x)$ é denominado como o grau de pertinência do elemento x ao conjunto A , indicando o quanto x é compatível com o conjunto A (KLIR; YUAN, 1995).

Os elementos que podem ser mapeados para conjuntos nebulosos definem as variáveis do mundo real que servem de entrada para um sistema *fuzzy*. A cada uma dessas variáveis, pode-se associar um conjunto de valores linguísticos que caracterizam os valores que estas variáveis podem assumir. Sendo assim, define-se como variável linguística aquela cujos valores são conjuntos de termos, terminologias, nomes ou rótulos, ao invés de números, permitindo uma modelagem similar ao pensamento humano (ZADEH, 1996).

Formalmente, uma variável linguística pode ser definida por uma quintupla denotada por:

$$\langle v, T, X, g, m \rangle;$$

onde v é o nome da variável; T é o conjunto de termos linguísticos de v (cada elemento de $T(v)$ representa um rótulo L dos termos que a variável v pode assumir); X é o universo de discurso da variável v ; g é uma regra sintática para gerar termos ou rótulos linguísticos; m é uma regra semântica que associa a cada rótulo L , um conjunto nebuloso no universo de discurso, ou seja $m(L)$.

A quantidade de valores linguísticos define a granularidade, isto é, a especificação e distribuição dos termos linguísticos e, por conseguinte, a partição nebulosa do universo de discurso correspondente. Um número pequeno de termos linguísticos define uma partição esparsa ou grossa, ao passo que um número maior resulta numa partição fina.

Para cada variável linguística, seja ela de entrada ou saída, tem-se uma função cuja denominação é função de pertinência. O papel dessa função é determinar o grau de pertinência do valor real da variável em questão em relação ao valor linguístico representado pela função (KLIR; YUAN, 1995).

O valor gerado pela função de pertinência varia no intervalo $[0,1]$ sendo que, quanto mais próximo de um, maior é a relação do valor real da variável com a variável linguística

que a função representa.

As funções de pertinência podem ser de diversos tipos (ZIMMERMANN, 2001). Os tipos mais usados são triangular, trapezoidal e gaussiana e estão representadas respectivamente pelas equações (3.2), (3.3), (3.4).

$$A(x) = \begin{cases} 0, & \text{se } x < a \\ \frac{x-a}{b-a}, & \text{se } x \in [a, b] \\ 1, & \text{se } x \in [b, c] \\ \frac{d-x}{d-c}, & \text{se } x \in [d, c] \\ 0, & \text{se } x > d \end{cases} \quad (3.2)$$

$$A(x) = \begin{cases} 0, & \text{se } x < a \\ \frac{x-a}{m-a}, & \text{se } x \in [a, m] \\ 1, & \text{se } x \in [m, n] \\ \frac{b-x}{b-n}, & \text{se } x \in [n, b] \\ 0, & \text{se } x > b \end{cases} \quad (3.3)$$

$$A(x) = e^{-k(x-m)^2} \quad (3.4)$$

A esse processo de mapear o valor real de uma variável do sistema para valores do conjunto *fuzzy*, ou seja, calcular o grau de pertinência da variável do mundo real para cada uma das variáveis linguísticas consideradas, dá-se o nome de *fuzificação* (ZIMMERMANN, 2001).

Além das funções de pertinência, um sistema *fuzzy* também é composto por um conjunto de regras. De forma geral, as regras *fuzzy* podem assumir a seguinte forma

$$\text{Se } x = A \text{ E } y = B \text{ E } \dots \text{ entao } z = C$$

onde A, B e C são rótulos linguísticos que representam cada um dos conjuntos nebulosos considerados. Frequentemente, $x = A \text{ E } y = B \dots$ é denominado de antecedente da regra enquanto $z = C$ é denominado de consequente da regra. Um conjunto de regras *fuzzy* tem como objetivo representar o conhecimento do especialista do problema do processo ou modelo abordado (DUBOIS; PRADE; YANGER, 1996).

Ao fim do processo de *fuzificação*, pode-se verificar quais regras do sistema *fuzzy* foram ativadas. Entende-se por regras ativadas, aquelas cujo processamento do antecedente

para suas entradas gerou graus de pertinência não-nulos. A associação das informações das pertinências não-nulas pode ser realizada por meio da aplicação de um operador *T-Norma*. Os dois operadores mais utilizados são o *Produto* ($\text{PROD}(A,B,\dots)$) e o *Mínimo* ($\text{MIN}(A,B,\dots)$) cujos resultados consistem, respectivamente, do produto dos valores de pertinência das variáveis de entrada nos conjuntos definidos nos termos do antecedente da regra, e do menor valor de pertinência dentre as variáveis de entrada para os conjuntos representados nos termos do antecedente das regras (KLIR; YUAN, 1995).

Existem vários modelos de regras que podem ser usados em sistemas *fuzzy*. Os dois mais utilizados são o *Modelo Mamdani* e o *Modelo Takagi-Sugeno*.

No *Modelo Mamdani*, as regras do sistema assumem o seguinte formato:

$$\text{Se } x_1 = X_1^{(i)} \text{ E } x_2 = X_2^{(i)} \text{ E } \dots \text{ E } x_k = X_k^{(i)} \text{ entao } Y_n = Y^{(m)}$$

Já no *Modelo Takagi-Sugeno*, as regras do sistema assumem o seguinte formato:

$$\text{Se } x_1 = X_1^{(i)} \text{ E } x_2 = X_2^{(i)} \text{ E } \dots \text{ E } x_k = X_k^{(i)} \text{ entao } Y_n = c_{0i} + c_{2i}x_1 + c_{2i}x_2 + \dots + c_{ki}x_k$$

Em ambos os casos, o antecedente das regras relaciona o valor de uma variável de entrada com um valor linguístico definido por uma função de pertinência. Com relação ao conseqüente, no *Modelo Mamdani* o valor de saída constitui uma ponderação da composição das regras ativadas. No *Modelo Takagi-Sugeno* o valor do termo conseqüente de cada regra será o valor resultante da função polinomial presente no conseqüente da regra ativada (ZIMMERMANN, 2001).

Uma vez determinado quais regras foram ativadas, pode-se realizar a “*defuzificação*”. “*Defuzificação*” é o processo de mapear valores do conjunto *fuzzy* para valores do mundo real, consistindo em um método de ponderação (ZIMMERMANN, 2001).

Existem vários métodos de *defuzificação* sendo que o mais empregado é o “*Método de Centro de Área*”, definido pela equação (3.5).

$$z = \frac{\sum_{n=1}^N \nu_n \mu(\nu_n)}{\sum_{n=1}^N \mu(\nu_n)} \quad (3.5)$$

Onde ν_n é o valor do consequente da regra n e $\mu(\nu_n)$ é o grau de ativação da regra n .

Uma vez definido o modelo de regras a ser aplicado e uma vez determinada a base de conhecimento composta por regras *fuzzy* e as funções de pertinência para os conjuntos *fuzzy* que irão compor a base de dados, um “*Sistema de Inferência Fuzzy*” (*FIS*), pode ser proposto.

O *FIS* funcionará como um sistema que recebe valores numéricos e transforma esses valores em inferências de variáveis linguísticas (*fuzificação*) (DUBOIS; PRADE; YANGER, 1996). Em seguida, a variável *fuzzy* é processada por um motor de inferência que faz uso das regras nebulosas do sistema *fuzzy*. No final do processo, a variável *fuzzy* resultante é novamente transformada em uma variável do mundo real podendo então ser utilizada em processos externos à computação do modelo *fuzzy* em questão. A Figura 3 ilustra a arquitetura de um *FIS* genérico.

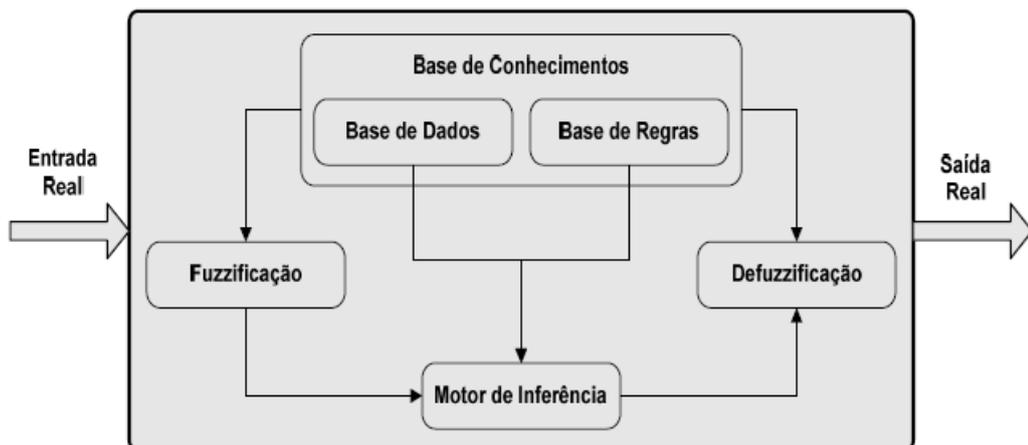


Figura 3: Arquitetura geral de um FIS.

3.2 Introdução a Sistemas Neuro-Fuzzy

Wang e Mendel (1992) introduziram em 1992 um algoritmo que permite a geração de modelos *fuzzy* em processos de modelagem em geral, incluindo aplicações em previsões. Tal algoritmo, denominado de “*Algoritmo de Wang e Mendel*”, está dividido em cinco etapas. Na primeira etapa, os dados da série temporal são divididos em conjuntos *fuzzy*. Na segunda etapa, as regras *fuzzy* são geradas. Na terceira etapa, defini-se um grau para cada regra gerada. Na quarta etapa, a base de conhecimento é gerada. Na quinta parte, ocorre a *defuzzificação*, ou seja a inferência do valor da previsão. A importância do “*Algoritmo de*

Wang e Mendel” se deve ao fato de ser um dos pioneiros na aplicação da lógica nebulosa em processos de previsão.

Apesar de simples, o processo de determinação de modelos *fuzzy* via *Algoritmo de Wang e Mendel* pode ser bastante tedioso e demorado. Portanto, a fim de diminuir esforços e estimar modelos de previsão mais eficientes, sistemas *neuro-fuzzy* (JANG; SUN; MIZUTANI, 1997) podem ser aplicados.

A idéia básica de um sistema *neuro-fuzzy* é implementar *Sistemas de Inferência Fuzzy* (seção 3.1) sobre arquiteturas de redes neurais artificiais, caracterizando um sistema híbrido, de forma que os paradigmas de aprendizado comuns às redes neurais possam ser aplicados. Dessa forma, aproveita-se os benefícios das redes neurais, que são apropriadas para a criação de modelos a partir de conhecimento implícito embutido em um conjunto de dados, e dos sistemas *fuzzy*, que são adequados para a criação de modelos a partir de um conhecimento explícito, originário de especialistas humanos.

Dentre os vários *Sistemas Neuro-Fuzzy* propostos, pode-se destacar o *ANFIS* (*Adaptive Network Based Fuzzy Inference System*), o *FSOM* (*Fuzzy Self-Organization Map*), o *NEFClass* (*Neuro Fuzzy Classification*) e o *DENFIS* (*Dynamic Evolving Neural-Fuzzy Inference System*).

O *FSOM*, proposto por Mitra e Pal (1994), é um *Sistema Neuro-Fuzzy* baseado no modelo de mapas auto-organizáveis de *Kohonen*, capaz de realizar classificações por meio da aplicação de técnicas de lógica nebulosa. A arquitetura de uma *FSOM* é composta por um conjunto de cinco camadas cujo funcionamento é semelhante à de um *ANFIS* (seção 3.4).

O *NEFClass*, proposto por Nauck e Kruse (1995), é um *Sistema Neuro-Fuzzy* cuja aplicação básica também é em sistemas de classificação. A diferença principal entre o *NEFClass* e o *FSOM* é que o primeiro apresenta valores crisp na camada de saída, conforme ilustrado por (3.6) que generaliza o formato das regras *fuzzy* que compõem os modelos gerados pelo *NEFClass*.

$$SE x_1 \in \mu_1 E x_2 \in \mu_2 \dots ENTAO padrao(x_1, x_2, \dots) pertence a i \quad (3.6)$$

Kasabov e Song (2001) propuseram o *Denfis*, que consiste em um *Sistema Neuro Fuzzy* capaz de evoluir de forma dinâmica adicionando novas regras ao *FIS* antes ou durante o processo de treinamento, e removendo regras durante ou após o processo de treinamento.

O *ANFIS* (JANG, 1993) é o *Sistema Neuro-Fuzzy* aplicado na construção dos modelos de previsão baseados em lógica nebulosa desse trabalho. Sendo assim, uma descrição mais detalhada sobre o *ANFIS* será realizada na seção 3.4.

3.3 Trabalhos Relacionados

Esta seção tem como objetivo apresentar alguns trabalhos que utilizaram o *ANFIS* para construção de modelos baseados em regras *fuzzy* capazes de atuar em processo de previsão.

Fahimifard et al. (2009) realizaram um estudo de desempenho entre métodos lineares e não lineares quando aplicados em processos de previsão. A técnica linear considerada foi o *ARIMA* e o não linear foi a *ANFIS*. A fim de comparar ambas as técnicas, uma série que define uma variável agrícola iraniana foi considerada. O processo considerou a previsão em três horizontes distintos com uma, duas e quatro semanas respectivamente. A análise dos resultados permitiu concluir que o *ANFIS* obteve um desempenho consideravelmente maior em todas os experimentos realizados.

Liao e Tsao (2007) propuseram um sistema híbrido baseado em lógica nebulosa e algoritmos genéticos para a previsão de demanda de carga elétrica a curto prazo. A finalidade da aplicação do algoritmo genético consistiu em se estimar os parâmetros de treinamento para o *ANFIS* considerado para determinação dos modelos *fuzzy* de previsão. Como forma de avaliação do modelo obtido, uma comparação foi realizada com modelos obtidos via aplicação de redes neurais. Os modelos baseados em lógica nebulosa obtiveram resultados mais precisos.

De forma similar, Syed-Ahmad et al. (2007) e Ying e Pan (2008) utilizaram-se de modelos *fuzzy* para prever a demanda de carga elétrica. O primeiro a curto e o segundo a longo prazo. Syed-Ahmad et al. (2007) considerou observações horárias de consumo de energia elétrica para compor o conjunto de treinamento usado para treinamento do *ANFIS*. O modelo *fuzzy* obtido foi aplicado na previsão de demanda vinte e quatro horas a frente. Ying e Pan (2008), utilizaram observações anuais de 1981 até 1996 para previsão de carga para 1997 até 2000 em diferentes regiões de Taiwan. Em ambos os casos os modelos obtidos via *ANFIS* obtiveram resultados compatíveis com os modelos neurais usados para fins de comparação.

Em Hidrologia, é de interesse o estudo de variáveis relacionadas a atividade pluviométrica, seca, fluxo de água, dentre vários outros. Nesse sentido, Nayak et al. (2004) e

Firat e Güngör (2007) estudaram, respectivamente, o fluxo de água nos Rios *Baitarani* na Índia e *Great Menderes* na Turquia. Nayak et al. (2004) propuseram seis modelos *fuzzy* sendo que cada um deles foi gerado por meio de treinamento que considerou de um até seis observações do passado para previsão de um passo à frente. Comparações com modelos obtidos via aplicação de redes neurais e *ARMA* foram realizadas, o *ANFIS* obteve melhores resultados. Já Firat e Güngör (2007) construíram quatro subconjuntos de dados a partir das 5844 observações diárias disponíveis aplicando três dos quatro conjuntos para treinamento do *ANFIS* e um para teste. Para cada subconjunto de treinamento, foram criados modelos que levaram em consideração até sete observações do passado. A validação dos modelos foi realizada com base na comparação de desempenho de modelos obtidos via redes neurais. Mais uma vez, os modelos *fuzzy* obtiveram melhores resultados.

Chen, Ying e Pan (2010b), a partir do uso do *ANFIS*, estimaram um modelo *fuzzy* capaz de prever a quantidade de turistas que chegam em Taiwan vindos dos Estados Unidos, Japão e Hong Kong. Para construção do modelo, um conjunto de treinamento que levava em consideração observações mensais que iam de 1989 a 2000 foi construído. O objetivo consistia em estimar o número de turistas para os anos de 2001 a 2003.

En-ANFIS foi o nome dado ao sistema híbrido proposto por Chen e Zhang (2005) para previsão de séries temporais. O *En-ANFIS* utiliza o paradigma de treinamento *Ensemble* (daí a origem do nome do sistema) que consiste em treinar vários componentes para a realização da mesma tarefa. O treinamento de cada componente pode ser realizado por meio de conjuntos construídos de forma aleatória ou por meio do método *Bootstrap Sampling* com ou sem repetições. O valor de previsão é resultado da combinação do resultado de previsão de cada componente isolado sendo que essa combinação pode ser baseada em pesos ou não. Caso a opção por pesos seja a considerada, atribui-se pesos maiores a valores de previsão de componentes que apresentaram menores erros durante o processo de treinamento.

Na última década, houve um aumento no interesse de previsões da condição de luminosidade em ambientes como salas de prédios ou cômodos de apartamentos por exemplo. A fim de criar controladores de iluminação automáticos, Kurian et al. (2006) desenvolveram um modelo *fuzzy* a partir do treinamento de um *ANFIS* para previsão de um, seis e dez passos a frente. O conjunto de treinamento foi construído com o auxílio de um software de simulação que simula as condições de iluminação de um ambiente segundo parâmetros como dimensão da sala, obstruções, posicionamento, condições do céu dentre outros.

Ming-bao e Xin-ping (2008) aplicaram um sistema híbrido composto pelo *ANFIS* e

algoritmo genético para prever o fluxo de tráfego de automóveis. Para obtenção dos dados de treinamento, um software de simulação de propósito específico foi utilizado para gerar novecentas observações de fluxo sob determinadas condições. Das novecentas observações, oitocentas foram usadas para criação do conjunto de treinamento e cem foram usadas para compor o conjunto de testes. Os dados de treinamento foram agrupados por meio do algoritmo “*Subtractive Clustering*” e, com base nesse agrupamento, o processo de treinamento foi inicializado. A definição de um *raio de clusterização* ótimo foi realizada por meio da aplicação de um algoritmo genético. Os resultados foram comparados aos resultados obtidos via aplicação de modelos gerados por redes neurais *MLP* e *ANFIS*. O sistema híbrido apresentou melhores resultados.

Muitos trabalhos também podem ser encontrados na área financeira. Como exemplo, pode-se citar previsões de ações de mercado, variações de cambio e preços de produtos em geral.

Castilho e Melin (2002) consideraram a aplicação do *ANFIS* em processos de previsão do preço de caixas de cebolas e de tomates e na previsão da variação do Peso Mexicano em relação ao Dólar. A comparação foi realizado com os resultados de modelos neurais obtidos via redes *MLP* treinadas com o algoritmo “*Lovenberg Marquardt*” e “*Back Propagation*” com termo de momento. Os modelos neurais obtiveram melhores resultados quando aplicados na previsão da variação da Peso Mexicano em relação ao Dólar.

Por fim, Li e Xiong (2005) utilizaram modelos *fuzzy* gerados a partir do *ANFIS* para prever o mercado de ações de Changai. Para realizar tal procedimento, foram consideradas 244 observações diárias de ações inseridas no intervalo de março de 2004 à março de 2005. Para construção dos conjuntos de treinamento e testes, foram utilizados 200, e 44 observações respectivamente. Os resultados indicam modelagem e previsão razoáveis da série apresentando erro de 1% e 5% respectivamente.

3.4 Adaptive Network-Based Fuzzy Inference System

O *ANFIS* é um sistema híbrido que funciona como um *FIS*, o qual é gerado a partir do treinamento de uma rede neural adaptativa e simulado a partir da propagação de sinal aplicada à rede adaptativa treinada (JANG, 1993).

Uma rede neural adaptativa é uma rede multi-camadas com propagação de sinal onde cada neurônio implementa uma função denominada *função de ativação*. Uma função de ativação é definida em função dos valores de entrada do neurônio. A aplicação da função

resulta no valor de saída do neurônio que a implementa. A natureza das funções de ativação pode variar de neurônio para neurônio e suas escolhas dependem do contexto onde o problema a ser resolvido está inserido (JANG, 1993).

A Figura 4 ilustra uma rede neural adaptativa. A partir dela, pode-se notar a presença de dois tipos distintos de neurônios. Frequentemente, os neurônios representados por quadrados são denominados neurônios adaptativos enquanto que os representados por círculos são chamados de neurônios padrão. Convém ressaltar que as conexões entre neurônios indicam somente o fluxo da direção do sinal e que nenhum peso está associado a essas conexões.

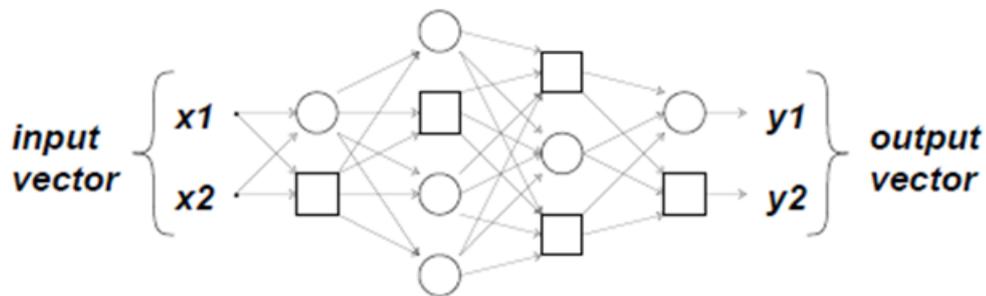


Figura 4: Exemplo de uma rede neural adaptativa (JANG, 1993).

Os neurônios adaptativos têm como característica a presença de um ou mais parâmetros que, juntamente com os valores de entrada, definem a sua função de ativação. O processo de treinamento de uma rede adaptativa consiste em determinar um valor para os parâmetros dos neurônios adaptativos de forma a minimizar o erro final o máximo possível.

O *ANFIS*, portanto, servirá de base para construção de um conjunto de regras nebulosas e funções de pertinência apropriadas. As regras e os parâmetros das funções de pertinência são estimados via um processo de treinamento híbrido realizado por meio de pares de entrada e saída de dados. O processo de treinamento do *ANFIS* é descrito na seção 3.4.2.

3.4.1 Arquitetura do ANFIS

Podem-se encontrar diferentes arquiteturas para o *ANFIS*, cada qual baseada nos modelos de regras *fuzzy* existentes (JANG; SUN, 1995). Neste trabalho, considerou-se a arquitetura de um *ANFIS* que implementa o modelo de regras do tipo *Takagi-Sugeno*.

A Figura 5 ilustra a arquitetura de um *ANFIS* composto por duas regras e que

recebe duas variáveis de entrada mapeando cada uma delas para dois conjuntos nebulosos distintos.

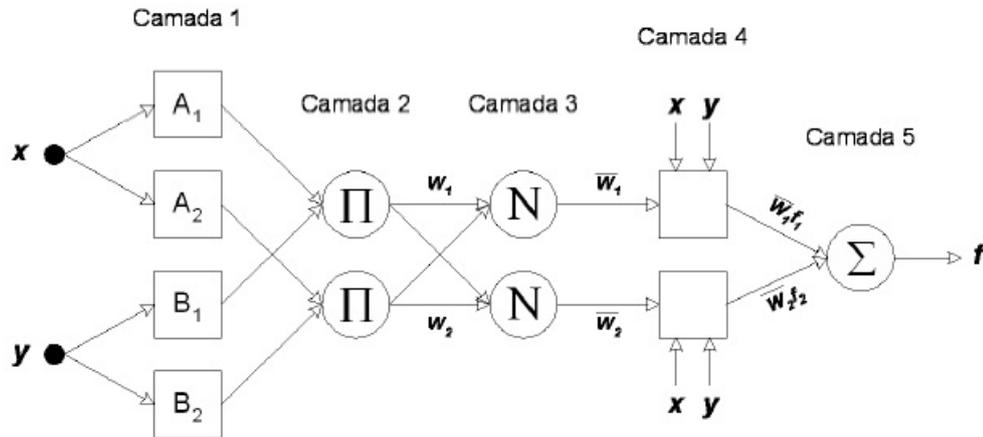


Figura 5: Arquitetura *ANFIS* de 5 camadas. Fonte: (JANG, 1993)

O *ANFIS* recebe as variáveis de entrada e, através de interações em suas camadas, realizando as etapas de *fuzificação* e *defuzificação* resultando ao final, no valor de saída correspondente ao padrão de entrada.

Na primeira camada ocorre o mapeamento das variáveis de entrada do universo de discurso para universo *fuzzy*, ou seja, é realizado o processo de *fuzificação* por meio da aplicação de uma função de pertinência. Como pode ser notado pela Figura 5, os neurônios da primeira camada são todos do tipo adaptativo, sendo assim, os parâmetros, que nesse caso dizem respeito às constantes das funções de pertinência, serão estimados via processo de treinamento.

A segunda camada representa o antecedente das regras *fuzzy* que compõem o *FIS*. O valor resultante (w_i , onde i representa o índice do neurônio) dos neurônios dessa camada é obtido por meio da aplicação de um operador *T-Norma* como o $Min(A, B)$ ou $Prod(A, B)$ por exemplo (seção 3.1), sob os valores de entrada desses neurônios.

A terceira camada é responsável pela normalização dos valores gerados na segunda camada. A saída dos neurônios que compõem essa camada indica o grau de ativação de uma regra em relação às outras. O valor de saída desses neurônios é dado pela equação (3.7).

$$\bar{w}_i = \frac{w_i}{\sum_{k=1}^K w_k} \quad (3.7)$$

Na quarta camada encontram-se as composições dos consequentes das regras *Takagi-Sugeno* cujo formato é dado por uma equação polinomial. No caso do *ANFIS* da figura 5, tem-se a equação $(p_i x + q_i y + r_i)$ (seção 3.1). Os valores de saída dos neurônios dessa camada são calculados por meio da equação (3.8). Convém ressaltar que os neurônios dessa camada são todos do tipo adaptativo e, sendo assim, seus parâmetros serão determinados por meio do processo de treinamento. Nesse caso, os parâmetros a serem estimados correspondem aos valores de p_i , q_i e r_i .

$$f_i = \bar{w}_i g_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (3.8)$$

Por fim, a quinta camada finaliza o processo de *defuzificação* através da aplicação da equação (3.9) que consiste da soma dos valores obtidos na quarta camada.

$$z_i = \sum_i f_i = \sum_i \bar{w}_i g_i \quad (3.9)$$

3.4.2 Treinamento do ANFIS

Conforme mencionado na seção 3.4.1, o processo de treinamento do *ANFIS* consiste em se estimar os parâmetros das funções de pertinência encapsuladas nos neurônios da primeira camada e os parâmetros que compõem as funções polinomiais lineares que representam os consequentes das regras *Takagi-Sugeno* representadas pelos neurônios da quarta camada.

O processo de treinamento pode ser realizado adotando-se um dentre os três procedimentos que se seguem (JANG, 1993):

- *Backpropagation* (propaga-se o erro por meio do cálculo do gradiente descendente para as camadas anteriores ajustando-se os parâmetros dos nós adaptativos da primeira e quarta camadas);
- *Backpropagation + Método dos Mínimos Quadrados* (usa o *backpropagation* para ajustar os parâmetros dos neurônios da primeira camada e o *Método dos Mínimos Quadrados* para ajustar os da quarta camada);
- *Backpropagation + Método dos Mínimos Quadrados* uma única vez (usa o *backpropagation* para ajustar os parâmetros dos neurônios da primeira camada e o *Método*

dos Mínimos Quadrados para ajustar os da quarta camada somente na primeira interação).

Por ser o método empregado na obtenção de todos os modelos baseados em regras *fuzzy* desse trabalho, esta seção descreve como o algoritmo “*Backpropagation* (etapa “*backward*”) e o *Método dos mínimos quadrados* (etapa “*forward*”) podem ser combinados a fim de gerar o processo de treinamento híbrido aplicados na etapa de experimentos.

3.4.2.1 Aplicação do Método dos Mínimos Quadrados no Processo de Treinamento (etapa “forward”)

Considerando-se a estrutura do *ANFIS* representada pela Figura 5 pode-se definir a saída f_{out_i} do i -ésimo par de treinamento como uma combinação linear dos parâmetros dos consequentes das regras (p_i, q_i, r_i) como segue (CHEN; YING; PAN, 2010a)

$$\begin{aligned}
 f_{out_i} &= \sum \bar{w}_j f_j \\
 &= \bar{w}_1 f_1 + \bar{w}_2 f_2 \\
 &= \bar{w}_1(p_1 x_i + q_1 y_i + r_1) + \bar{w}_2(p_2 x_i + q_2 y_i + r_2) \\
 &= (\bar{w}_1 x_i) p_1 + (\bar{w}_1 y_i) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x_i) p_2 + (\bar{w}_2 y_i) q_2 + (\bar{w}_2) r_2 \quad (3.10)
 \end{aligned}$$

onde $\langle (x_i, y_i), f_{out_i} \rangle$ constituem o i -ésimo par de treinamento. Sendo n o numero total de pares de treinamento disponíveis, considera-se as matrizes seguintes.

$$f = \begin{bmatrix} f_{out_1} \\ f_{out_2} \\ \cdot \\ \cdot \\ \cdot \\ f_{out_n} \end{bmatrix}, \theta = \begin{bmatrix} p_1 \\ q_1 \\ r_1 \\ p_2 \\ q_2 \\ r_2 \end{bmatrix} \text{ e } B = \begin{bmatrix} \bar{w}_1 x_1 & \bar{w}_1 y_1 & \bar{w}_1 & \bar{w}_2 x_1 & \bar{w}_2 y_1 & \bar{w}_2 \\ \bar{w}_1 x_2 & \bar{w}_1 y_2 & \bar{w}_1 & \bar{w}_2 x_2 & \bar{w}_2 y_2 & \bar{w}_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \bar{w}_1 x_n & \bar{w}_1 y_n & \bar{w}_1 & \bar{w}_2 x_n & \bar{w}_2 y_n & \bar{w}_2 \end{bmatrix}$$

Pode-se então, expressar a equação (3.10) em forma matricial por meio da equação

$$f = B\theta \quad (3.11)$$

onde θ é uma matriz desconhecida cujos elementos representam o conjunto de parâmetros dos consequentes das regras.

Através do *Método dos Mínimos Quadrados*, determinam-se os parâmetros do modelo por meio da equação (3.12) (DENAI; PALIS; ZEGHBIB, 2007).

$$\theta^* = (B^T B)^{-1} B^T f \quad (3.12)$$

Pode-se, a partir desse ponto, concluir a etapa “*forward*” do treinamento.

3.4.2.2 Aplicação do Backpropagation no Processo de Treinamento (etapa “backward”)

Denomina-se E , o erro médio quadrático obtido por meio da aplicação da equação (3.13).

$$E = \frac{1}{2}(d - f)^2 \quad (3.13)$$

Onde d é o resultado estimado e f é o valor esperado. Nesta etapa do treinamento, o erro E é propagado para as demais camadas do *ANFIS* e, com base nesse valor, os parâmetros dos nós adaptativos da primeira camada vão sendo ajustados.

Sendo assim, o valor de correção para um determinado parâmetro ρ da curva de pertinência da variável *fuzzy* de entrada pode ser obtido por $\Delta\rho$, que por sua vez, é definido pela equação (3.14) (MITRA et al., 2004).

$$\Delta\rho = \eta \frac{\partial E}{\partial \rho} = \eta \frac{\partial E}{\partial f} \frac{\partial f}{\partial f_i} \frac{\partial f_i}{\partial w_i} \frac{\partial w_i}{\partial \mu_{ij}} \frac{\partial \mu_{ij}}{\partial \rho} \quad (3.14)$$

Onde η é a taxa de aprendizagem. Da equação (3.13) pode-se resolver $\frac{\partial E}{\partial f}$ conforme a equação (3.15).

$$\frac{\partial E}{\partial f} = f - d \quad (3.15)$$

Da equação (3.9) ($f = \sum_{i=1}^n f_i$), resolve-se o termo $\frac{\partial f}{\partial f_i}$ resultando na equação (3.16)

$$\frac{\partial f}{\partial f_i} = 1 \quad (3.16)$$

Sabe-se que $f_i = \frac{w_i}{\sum_{i=1}^n w_i} (p_i x + q_i y + r_i)$, portanto, o termo $\frac{\partial f_i}{\partial w_i}$ resulta na equação (3.17).

$$\frac{\partial f_i}{\partial w_i} = \frac{(p_i x + q_i y + r_i) - f_i}{\sum_{i=1}^n w_i} \quad (3.17)$$

Da segunda camada, considerando que a T-Norma $Prod(A, B...)$ seja o operador aplicado, tem-se $w_i = \prod_{j=1}^M \mu_{A_{ji}}$. Sendo assim, uma vez resolvido o termo $\frac{\partial w_i}{\partial \mu_{ij}}$ obtém-se a equação (3.18).

$$\frac{\partial w_i}{\partial \mu_{ij}} = \frac{w_i}{\mu_{ij}} \quad (3.18)$$

Por fim, tem-se a derivada parcial da função de pertinência propriamente dita, representada pelo primeiro termo da equação (3.14). Considerando uma função trapezoidal, como aquela representada pela equação (3.3), pode-se estimar o valor dos parâmetros a , b e c após resolver as equações (3.19), (3.20) e (3.21).

$$\frac{\partial \mu_{ij}}{\partial a} = -\frac{\mu_{ij}(x_i)^2}{a} \left(\frac{(x_i - c)^2}{a} \right)^b \quad (3.19)$$

$$\frac{\partial \mu_{ij}}{\partial b} = -b\mu_{ij}(x_i)^2 \left(\frac{(x_i - c)^2}{a} \right)^{b-1} \quad (3.20)$$

$$\frac{\partial \mu_{ij}}{\partial c} = \frac{2b\mu_{ij}(x_i)^2}{x_i - c} \left(\frac{(x_i - c)^2}{a} \right)^b \quad (3.21)$$

O procedimento anteriormente descrito ocorre para todos os pares de dados disponíveis para treinamento. Uma vez que todos os pares de dados de treinamento são apresentados, uma interação ou uma época é caracterizada. O processo de treinamento segue até que o número de épocas determinado a priori seja atingido ou até que um erro objetivo, dado pela equação (3.13), calculado no final de cada interação seja alcançado.

3.5 O Algoritmo Fuzzy C-Means

O algoritmo *Fuzzy C-Means* é utilizado para determinar agrupamentos e seus centros segundo a norma euclidiana existente entre um dado e os centros dos agrupamentos (JANG; SUN; MIZUTANI, 1997). A idéia para entender a relação entre um dado e o agrupamento é o seguinte: quanto mais próximo do centro de um agrupamento o dado estiver, maior

será seu grau de pertinência a esse agrupamento.

Considerando um grupo de dados, que para esse caso corresponde aos dados de treinamento, $X = x_1, \dots, x_n$, onde cada ponto x_k ($k = 1, \dots, p$) é um vetor em \mathfrak{R}^p e considerando U_{cn} como um grupo de matrizes reais $c \times n$, onde c é um inteiro que obedece a relação $2 \leq c < n$, define-se a partição *fuzzy* para o conjunto X conforme equação (3.22).

$$M_{fnc} = U \in U_{cn} : u_{ik} \in [0, 1], \sum_{i=1}^c u_{ik} = 1, 0 < \sum_{k=1}^n u_{ik} < n \quad (3.22)$$

Onde U_{ik} é o grau de pertinência de x_k em um agrupamento i ($i = 1, \dots, c$).

A equação (3.23) pode ser utilizada para determinar se o algoritmo *Fuzzy C-Means* encontrou uma partição ótima.

$$J_m(U, V; X) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m \|x_k - v_i\|^2 \quad (3.23)$$

Da equação (3.23), tem-se que $V = (v_1, \dots, v_c)$ é a matriz que contém os centros dos agrupamentos onde todo $v_i \in \mathfrak{R}^p$, m é um peso constante entre $[1, \infty)$ que influencia no valor de pertinência.

O centro $v_i^{(t)}$ de cada agrupamento i ($i = 1, \dots, c$) para uma interação t pode ser calculada segundo a equação (3.24).

$$v_i^{(t)} = \frac{\sum_{k=1}^n (u_{ik}^t)^m x_k}{\sum_{k=1}^n (u_{ik}^t)^m} \quad (3.24)$$

De posse dos novos centros $v_i^{(t)}$, os valores de pertinência podem ser obtidos pela equação (3.25).

$$u_{ik}^{(t+1)} = \frac{1}{\sum_{j=1}^c \left(\frac{\|x_k - v_i^{(t)}\|^2}{\|x_k - v_j^{(t)}\|^2} \right)^{\frac{2}{m-1}}} \quad (3.25)$$

De forma esquemática, o algoritmo pode ser descrito conforme a listagem seguinte:

1. Escolher um valor para c , m , ε ;

2. Gerar a partição fuzzy U^0 de forma aleatória obedecendo às restrições impostas por (3.22);
3. Atribuir o valor 0 ao contador de interações;
4. Atribuir $J_m^{(t)} = 0$;
5. Calcular os centros $v_i^{(t)}$ segundo a equação (3.24);
6. Calcular a função objetivo $J_m^{(t+1)}$ por meio da equação (3.23);
7. Calcular os graus de pertinência $u_{ik}^{(t+1)}$ segundo a equação (3.25);
8. Calcular $\delta = J_m^{(t+1)} - J_m^{(t)}$;
9. Incrementar o contador de interações t ;
10. *Se a condição de saída for falsa, então retorna-se ao passo 5, senão, finaliza-se o algoritmo.*

Onde a *chamada condição de saída* é atingida quando $\delta \leq \varepsilon$ ou um pré determinado número de interações é alcançado.

3.6 Metodologia de Aplicação da Lógica Nebulosa em Processos de Previsão

A fim de gerar modelos baseados em regras *fuzzy* capazes de atuar nos processos de previsão das séries temporais consideradas nesse trabalho, um *ANFIS* que implementa o modelo de regras *Takagi-Sugeno*, conforme descrito na seção 3.4, foi utilizado.

Contudo, pode-se verificar a partir da seção 3.4 que o número de regras do *FIS* implementado por um *ANFIS* “*full conected*” na camada 2 é diretamente proporcional ao número de variáveis externas consideradas e ao número de conjuntos *fuzzy* considerados para cada variável de entrada. Sendo assim, o total de regras é dado por (3.26).

$$n = m^p \tag{3.26}$$

Onde n é o total de regras, m é o número de variáveis de entrada e p é o número de conjuntos fuzzy para onde cada variável de entrada pode ser mapeada.

É fácil perceber por meio de uma análise na equação (3.26) que o total de regras cresce exponencialmente em função do número de variáveis de entrada e conjuntos fuzzy. Esse crescimento elevado leva a um problema denominado de “*Maldição de Dimensionalidade*” (JANG; SUN; MIZUTANI, 1997).

A fim de lidar com o problema, um processamento inicial via técnicas de agrupamento nos dados que compõem o conjunto de treinamento será realizado. A técnica de agrupamento a ser aplicado é o algoritmo *Fuzzy C-Means*, descrito na seção 3.5.

O algoritmo, uma vez executado, gera como resposta uma matriz que define o quanto um determinado grupo de valores é pertencente a um determinado agrupamento. A partir dessa matriz um *FIS*, aqui denominado de *FIS inicial* pode ser projetado. O número de regras desse *FIS* inicial é igual ao número de “*clusters*” em que os dados foram agrupados. Vale ressaltar que o número de agrupamentos é um parâmetro informado para inicialização do algoritmo *Fuzzy C-Means*.

O *FIS inicial* é utilizado para inicializar a estrutura de um *ANFIS* que, nesse ponto, assume a estrutura genérica ilustrada na Figura 6, que, nesse caso em particular, representa a estrutura de um *ANFIS* para uma série temporal cujos dados foram agrupados em quatro “*clusters*” e cujas regras estão definidas conforme segue:

Se $x \in cluster1$ E $y \in cluster1$ ENTÃO $z \in cluster1$

Se $x \in cluster2$ E $y \in cluster2$ ENTÃO $z \in cluster2$

Se $x \in cluster3$ E $y \in cluster3$ ENTÃO $z \in cluster3$

Se $x \in cluster4$ E $y \in cluster4$ ENTÃO $z \in cluster4$

O *ANFIS* em questão receberá um treinamento gerando um *FIS* final cujo processo de *fuzificação* determina o quanto um determinado padrão de entrada pertence a cada um dos n agrupamentos gerados.

O Apêndice 3 traz uma listagem que ilustra como um modelo de previsão baseado em regras *fuzzy* foi obtido para um dos experimentos deste trabalho.

Ao fim do processo de treinamento, tem-se como resultado o *FIS final* que definirá o modelo baseado em regras *fuzzy* que poderá ser aplicado no processo de previsão.

Nos processos de previsão desse trabalho, os m neurônios na camada de entrada do *ANFIS* correspondem às m observações do passado que são consideradas para determinar

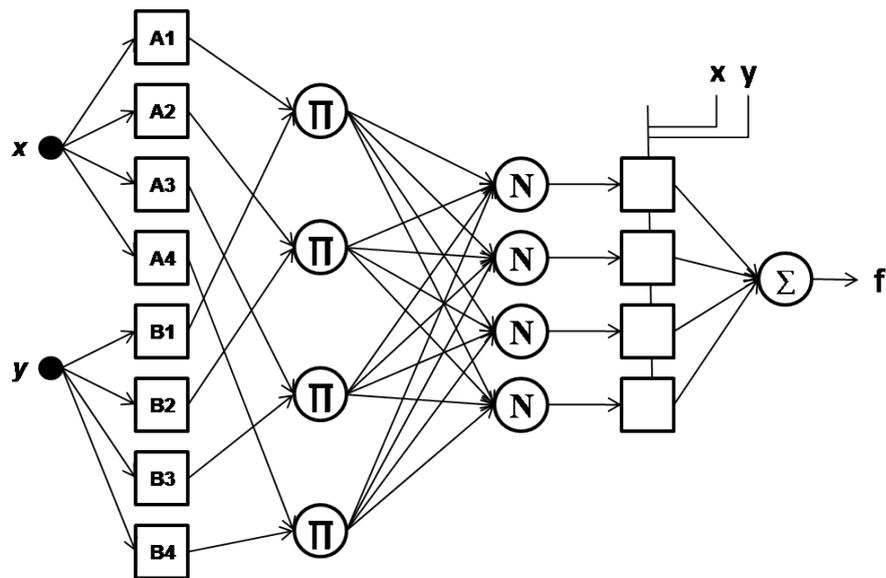


Figura 6: Arquitetura de um ANFIS para série com quatro clusters.

o valor do único neurônio na camada de saída cujo valor consiste na previsão um passo a frente (PALIT; POPOVIC, 2005). Esse valor previsto é interativamente usado como entrada para previsão no passo seguinte.

O diagrama de atividades da Figura 7 ilustra cada uma das etapas da metodologia de previsão discutida nessa seção, que vai desde a preparação dos dados, passa pela determinação do modelo e encerra-se com a previsão propriamente dita.

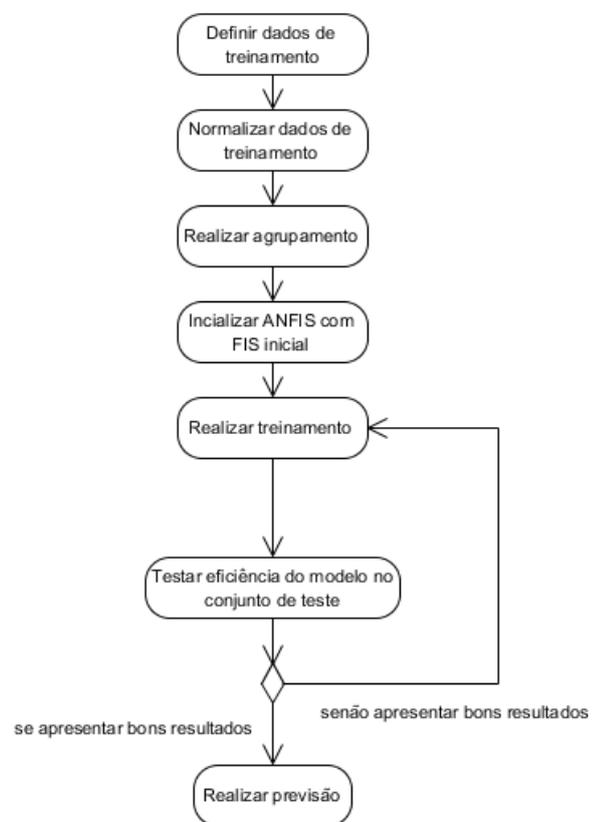


Figura 7: Diagrama de atividades da metodologia de previsão baseada em modelo fuzzy.

4 Conjuntos Aproximados e Previsões

O objetivo desse capítulo é propor uma metodologia de aplicação da teoria de *conjuntos aproximados* na previsão de séries temporais.

Para alcançar esse objetivo será introduzido, em um primeiro momento, conceitos básicos relacionados a *conjuntos aproximados* a fim de que uma contextualização da metodologia seja realizada.

Em seguida, alguns trabalhos que aplicam a teoria de *conjuntos aproximados* em processos gerais de previsão serão apresentados. Contudo, convém ressaltar que nos artigos pesquisados, praticamente nada relacionado à previsão de séries temporais propriamente dita foi encontrado. A previsão em cada trabalho consiste em se determinar uma classificação final que é referente ao contexto de cada problema tratado.

Uma vez apresentados os conceitos básicos de *conjuntos aproximados*, uma proposta de previsão de séries temporais será apresentada. Conforme descrito posteriormente, a metodologia é baseada nos trabalhos de Pinheiro et al. (2010a, 2010b) que propuseram uma técnica baseada em *conjuntos aproximados* que pode ser aplicada em problemas que envolvem aproximação de função e modelagem de sistemas dinâmicos.

Por fim, uma visão de como a proposta de previsão pode ser aplicada nos problemas considerados nesse trabalho será dada.

4.1 Introdução a Conjuntos Aproximados

A teoria de *Conjuntos Aproximados* (*Rough Sets*) foi introduzida por Pawlak com dois objetivos principais: revelar importantes estruturas em conjuntos de dados e classificar objetos (PAWLAK, 1982). Diferentemente dos métodos de análise de dados convencionais, que frequentemente utilizam-se de mecanismos estatísticos, a abordagem de conjuntos aproximados é baseada em técnicas de *data-mining* para descobrir conhecimento (TAY; SHEN, 2002).

A fim de aplicar os conceitos da teoria, um tipo especial de *Tabela de Informação* denominado de *Tabela de Decisão* é frequentemente considerado (PAWLAK et al., 1995). Estas *Tabelas de Decisão* são sistemas bi-dimensionais formados por *objetos* no formato *atributo-valor*. Tais *objetos* encontram-se dispostos na *Tabela de Decisão* agrupados em *classes* (PAWLAK, 2003). Os conceitos referentes a esses tipo especiais de tabelas serão melhores explorados nas seções seguintes.

Um *Sistema de Apoio à tomadas de Decisão*, geralmente baseado em uma *Tabela de Informação*, pode ser proposto com o objetivo de auxiliar no processo de tomada de decisão independentemente do contexto do problema em análise.

Contudo, é comum a situação em que nem todos os *atributos* presentes na *Tabela de Informação* em questão sejam úteis no processo de tomada de decisão, não se fazendo necessário portanto, a análise de todos os *atributos*.

Estas informações desnecessárias e que não modificam os resultados com relação à decisão a ser tomada são denominados de *atributos irrelevantes*. Dessa forma, para a construção de um modelo de auxílio ao processo de tomada de decisão, é de interesse considerar somente aqueles *atributos* fundamentais para a tomada da decisão, separando esses *atributos* daqueles não relevantes.

A teoria dos *Conjuntos Aproximados* fornece meios sistemáticos para eliminar variáveis irrelevantes através de um processo de redução da *Tabela de Informação*, baseado na definição de *redutos* que, conforme descrito nas seções seguintes, são conjuntos de atributos capazes de manter as mesmas propriedades da representação de conhecimento quando esta é feita utilizando todos os atributos originais (PAWLAK, 1982).

A Figura 8 representa o processo de redução de um forma geral.

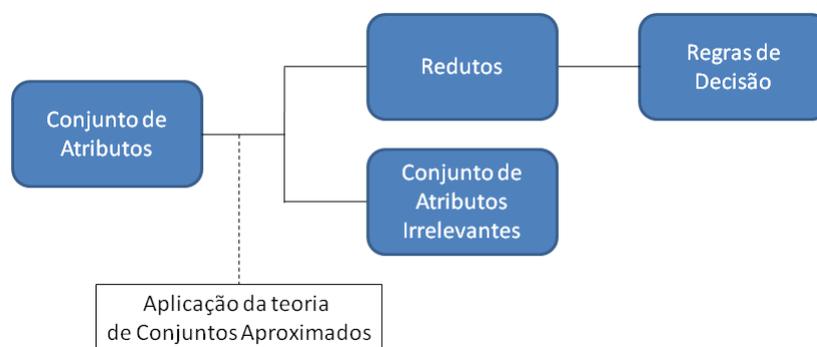


Figura 8: Objetivo geral da teoria de *Conjuntos Aproximados*.

Com o intuito de empregar a teoria em processos de classificação, a aplicação da

mesma atua sobre os *objetos* agrupando-os em classes distintas. *Objetos* agrupados em uma mesma classe são indiscerníveis entre si (PAWLAK et al., 1995). Contudo, é comum encontrar nos dados que compõem os *Sistemas de Informações*, *objetos* cujos dados são inconsistentes. Essa inconsistência impede que a discernibilidade entre objetos seja realizada visto que existem *objetos* com mesmos valores dos atributos classificados em classes diferentes. A administração dessa imprecisão é realizada por meio dos conceitos de *aproximação inferior* e *aproximação superior* (PAWLAK; SKOWRON, 2007), conforme explicado na seção 4.2.

Como motivação para o estudo dos *conjuntos aproximados*, é apresentado a seguir uma série de vantagens da aplicação da teoria na solução de problemas, conforme levantado por Tay e Shen (2002).

- A construção de modelos de decisão tratados por esta teoria necessita somente dos dados originais referentes ao problema em estudo e não de dados externos, como é o caso dos modelos estatísticos que frequentemente necessitam de informações da distribuição de probabilidade e dos modelos *fuzzy* que necessitam que funções de pertinência sejam determinadas;
- O modelo baseado em regras resultante fornece informações capazes de analisar não somente dados quantitativos mas também qualitativos;
- A aplicação da teoria permite a descoberta de informações representativas ocultas nos dados e expressa esses fatos na forma de regras de decisão cuja sintaxe é bem próxima à linguagem natural;
- O conjunto de regras que compõem os modelos dão uma descrição generalizada do conhecimento presente no conjunto de dados eliminando as redundâncias;
- As regras de decisão obtidas são baseadas em informações pertinentes visto que foram geradas a partir de conjuntos de exemplos reais;
- Os modelos gerados por meio da teoria pertencem à uma classe de modelos denominados modelos *caixa branca* cuja característica principal é a facilidade de entendimento e análise visto que são formados por regras cuja sintaxe é muito próxima da linguagem natural, diferentemente de outras técnicas como as redes neurais artificiais por exemplo que são modelos *caixa preta* cuja interpretação daquilo que representa o conhecimento é geralmente muito difícil.

4.2 Conceitos Fundamentais de Conjuntos Aproximados

A forma mais usual de se representar os dados disponíveis para análise na teoria de *conjuntos aproximados* é por meio de uma *tabela de informação* (PAWLAK et al., 1995). Desse ponto em diante, toda referência a uma *tabela de informação* será feita pela expressão *sistema de informação* (*SI*) conforme adotado na maioria dos trabalhos correlatos.

Um *SI* é composto por uma coleção de *objetos*. Cada *objeto* é caracterizado por um conjunto de *atributos* que são os mesmos para todos os *objetos*. Os valores que cada atributo pode assumir diferencia um determinado *objeto* de outro (PAWLAK et al., 1995).

Formalmente, o *SI* pode ser definido como um par ordenado $S = (U, A)$ onde U é um conjunto finito e não vazio de *objetos* chamado *Universo* e A é um conjunto finito e não vazio de elementos chamados de *atributos*. Os elementos do *Universo* são referenciados como *objetos*. Cada *atributo* $a \in A$ é uma função de classificação $a : U \rightarrow V_a$, onde V_a é o conjunto dos valores válidos para o atributo a (sua faixa de valores).

A classificação relacionada é dada por um atributo especial denominado de *atributo de decisão*. Sendo assim, pode-se estender a definição de um *Sistema de Informação* para *Sistema de Decisão* acrescentando o *atributo de decisão*.

De uma forma mais formal, define-se um *Sistema Decisão* (*SD*) como qualquer *SI* da forma $S = (U, A \cup d)$, onde $d \notin A$ é o *atributo de decisão*, e os elementos de A são os *atributos condicionais*.

A Tabela 1 consiste de uma representação tabular de um *SD* genérico.

Tabela 1: Representação tabular de um sistema de informação.

$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	\dots	$\mathbf{x_N}$	\mathbf{y}
$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	\dots	$x_N^{(1)}$	$y^{(1)}$
$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$	\dots	$x_N^{(2)}$	$y^{(2)}$
\dots	\dots	\dots	\dots	\dots	\dots
$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	\dots	$x_N^{(k)}$	$y^{(k)}$
\dots	\dots	\dots	\dots	\dots	\dots
$x_1^{(m)}$	$x_2^{(m)}$	$x_3^{(m)}$	\dots	$x_N^{(m)}$	$y^{(m)}$
\dots	\dots	\dots	\dots	\dots	\dots
$x_1^{(v)}$	$x_2^{(v)}$	$x_3^{(v)}$	\dots	$x_N^{(v)}$	$y^{(v)}$

Um outro conceito importante da teoria é o conceito de *Relação de Não-Discernimento*. Se essa relação existe entre dois *objetos* tem-se, para um conjunto de *atributos* desses *objetos*, que os valores desses *atributos* são idênticos entre si, não possibilitando que se

destingua um *objeto* do outro (GOH; LAW, 2003).

Formalmente, tem-se a seguinte definição para uma *Relação de Não-Discernimento*: Para cada subconjunto de atributos $B \subseteq A$ no *SI* $S = (U, A)$, uma relação de equivalência $IND_S(B)$ é associada, chamada de *Relação de Não-Discernimento* e é definida conforme expressão (4.1) (PAWLAK, 1982).

$$IND(B) = \{(x, y) \in U^2 | \forall a \in B, a(x) = a(y)\} \quad (4.1)$$

O conjunto de todas as classes de equivalência na relação $IND_S(B)$ é denotado por $U/IND_S(B)$.

Das classes de equivalência, emergem dois outros conceitos importantes da teoria: *Aproximação Inferior* e *Aproximação Superior* dadas pela equação (4.2). Dado que X é um conjunto tal que $X \subseteq U$ a *Aproximação Inferior* de X , (B_*) , é o conjunto de possíveis *objetos* que podem ser classificados com certeza como membros de X utilizando o conjunto de *atributos* B . Já a *Aproximação Superior* de X , (B^*) , é o conjunto de *objetos* que podem ser classificados como membros de X utilizando um conjunto de *atributos* B (PAWLAK; SKOWRON, 2007). No caso em que $B^*(X) = B_*(X)$, o conjunto X é denominado preciso (*crisp*), no caso contrário, ele é definido como impreciso, grosseiro (*rough*) ou aproximado.

$$\begin{aligned} B_*(X) &= \{x \in U | U/IND(X) \subseteq X\} \\ B^*(X) &= \{x \in U | U/IND(X) \cap X \neq \emptyset\} \end{aligned} \quad (4.2)$$

A partir da *Aproximação Inferior* e *Superior*, pode-se definir a *Região de Fronteira* que é dado pela equação (4.3) e define os objetos impossíveis de serem classificados em X (PAWLAK, 1982).

$$RF(X) = B^*(X) - B_*(X) \quad (4.3)$$

As aproximações anteriormente discutidas podem ter sua qualidade medida em termos dos próprios elementos que a definem (PAWLAK; SKOWRON, 2007). Três importantes medidas de qualidade são:

- O coeficiente de imprecisão, $\alpha_B(X)$, define a qualidade da aproximação de X e é dado pela equação (4.4)

$$\alpha_B(X) = \frac{|B_*(X)|}{|B^*(X)|} \quad (4.4)$$

onde $|B_*(X)|$ e $|B^*(X)|$ denotam a cardinalidade das *Aproximações Inferior* e *Superior* respectivamente. O valor $\alpha_B(X)$ pertence ao intervalo $[0, 1]$. Quanto mais próximo de um, mais preciso é X em relação ao conjunto de atributos B e quanto mais próximo de zero, mais impreciso (*rough*) é X em relação ao conjunto de atributos B

- O coeficiente da qualidade da *Aproximação Superior*, $\alpha_B|B^*(X)|$, pode ser interpretado como sendo o percentual de todos os objetos possivelmente classificados como pertencentes a X e é dado pela equação (4.5).

$$\alpha_B(B^*(X)) = \frac{|B^*(X)|}{|U|} \quad (4.5)$$

Onde $|U|$ é a cardinalidade do conjunto de objetos do SI

- O coeficiente da qualidade da *Aproximação Inferior*, $\alpha_B|B_*(X)|$, pode ser interpretado como sendo o percentual de todos os objetos certamente classificados como pertencentes a X e é dado pela equação 4.6.

$$\alpha_B(B_*(X)) = \frac{|B_*(X)|}{|U|} \quad (4.6)$$

A *Matriz de Discernibilidade* é o próximo conceito referente a *conjuntos aproximados* a ser discutido. Uma *Matriz de Discernibilidade* é uma matriz na qual as classes são índices e os *atributos condicionais*, que podem ser utilizados para distinguir entre as classes, são inseridos nas células correspondentes às classes a serem discernidas.

Para esse conceito, tem-se a seguinte definição mais formal: Seja B um conjunto de atributos tal que $B \subseteq A$ para o $SI S = (U, A)$, a *Matriz de Discernibilidade* é dada por $M_D(B) = m_D(i, j)_{n \times n}$, $1 \leq i, j \leq n$ onde $n = |U/IND(B)|$ e $m_D(i, j)$ é obtido conforme a expressão (4.7):

$$m_D(i, j) = a \in B | a(E_i) \neq a(E_j) \text{ para } i, j = 1, 2, \dots, n \quad (4.7)$$

Ou seja, o elemento $m_D(i, j)$ na *Matriz de Discernibilidade* é o conjunto de atributos de B que discerne as classes de objetos $E_i, E_j \in U/IND(B)$.

Uma vez estabelecida a definição de *Matriz de Discernibilidade*, pode-se aplicar à mesma a *Função de Discernibilidade*. A aplicação dessa função resulta em uma expressão booleana cujo processo de minimização gera os *redutos* do *SI*. A *Função de Discernibilidade* é definida pelas equações (4.8) e (4.9).

$$f(B) = \wedge \{ \vee \bar{m}_D(i, j) \mid i, j = 1, 2, \dots, n, m_D(i, j) \neq 0 \} \quad (4.8)$$

$$\bar{m}_D(i, j) = \{ \bar{a}_k \mid a_k \in m_D(i, j) \} \quad (4.9)$$

De uma forma geral, a saída da *Função de Discernibilidade* é a conjunção das disjunções dos elementos de cada célula da *Matriz de Discernibilidade*.

A minimização da expressão booleana resultante da aplicação da equação (4.8) da origem a um dos principais conceitos da teoria de *conjuntos aproximados*, o de *redutos*. Um *reduto* é definido como um conjunto de atributos mínimos necessários para manter as mesmas propriedades de um *SD* que utiliza todos os atributos originais do sistema (PAWLAK; SKOWRON, 2007). De uma maneira mais formal, define-se um *reduto* de B como um conjunto de atributos $B' \subseteq B$ tal que todos os atributos $a \in B - B'$ são dispensáveis ou $IND(B') = IND(B)$. O termo $RED(B)$ é utilizado para denotar determinada família de redutos de B .

Uma vez encontrado os *redutos*, pode-se com base nestes determinar o conjunto de regras que define o modelo a ser aplicado na resolução do problema em questão. A transformação de um *reduto* em regras é conseguida unindo-se os valores dos *atributos condicionais* da classe de *objetos* da qual o *reduto* foi originado com os atributos correspondentes do *reduto*. Por fim, completa-se a regra a partir da construção de seu conseqüente por meio da adição da decisão no fim da mesma.

4.3 Aplicações de Conjuntos Aproximados

São muitas as áreas do conhecimento que se utilizam das vantagens da teoria de *conjuntos aproximados* para resolução de problemas. A fundamentação matemática dessa teoria permite que padrões ocultos nas bases de dados sejam encontrados (TAY; SHEN, 2002). Essa característica tem grande utilidade em problemas inseridos nos campos relacionados a reconhecimento de padrões, classificação da informação, mineração de dados, sistemas de decisão, sistemas especialista, processamento de sinais de som, processamento de imagens, *clusterização*, dentre outros (PAWLAK; SKOWRON, 2007).

Muitas pesquisas nas áreas de bioinformática, negócios, finanças, química, engenharia da computação, economia, engenharia elétrica, informática, medicina, biologia molecular, neurologia, robótica, ciências sociais, engenharia *web* e engenharia de *software* também podem ser encontradas. Uma revisão mais aprofundada de trabalhos relacionados pode ser encontrada em Pawlak e Skowron (2007).

Contudo, não é comum encontrar trabalhos que consideram a aplicação da teoria em processos de previsão, principalmente quando a previsão consiste em inferir valores futuros de uma série temporal.

Como exemplo de aplicação dos conceitos de *conjuntos aproximados* em previsões, pode-se citar o trabalho de Shen e Loh (2003). Estes autores analisaram o mercado de ações e determinaram se para uma determinada ação, esta deveria ser vendida, comprada ou retida. A fim de realizar tal procedimento, um sistema híbrido composto por uma rede neural *SOM* juntamente com conceitos de *conjuntos aproximados* foi proposto. O *SI* do trabalho era composto por sete *atributos de decisão* relacionados a indicadores financeiros relevantes ao problema. O procedimento foi dividido em três etapas. Na primeira, o sistema de informação foi reconstruído por meio de uma categorização dos dados realizado pela *SOM*. Na segunda etapa, os dados gerados na primeira etapa sofreram uma discretização. Na terceira e última etapa, aplicou-se conceitos da teoria de *conjuntos aproximados* para construção dos *redutos*. Dos *redutos* encontrados nessa etapa, escolheu-se para construção das regras aquele que apresentava menor número de *atributos*. No caso de empate no número de *atributos*, escolheu-se aquele que apresentou melhor qualidade de aproximação. O melhor modelo gerado era composto por 1044 regras e conseguiu classificar corretamente 58% das informações.

Ahn, Cho e Kim (2000) propuseram um sistema híbrido composto por um modelo gerado via *conjuntos aproximados* e redes neurais *MLP* para previsão da saúde financeira de empresas. Para determinação do modelo de previsão, foram considerados dados de 1200 empresas coreanas saudáveis e 1200 não saudáveis num total de 2400. Para representar cada empresa, considerou-se oito informações relevantes ao problema. A teoria de *conjuntos aproximados* foi empregada a fim de que uma redução na dimensionalidade fosse conseguida. Dos oito *atributos* considerados, verificou-se que somente quatro deles eram relevantes ao problema. Com base nesses *atributos*, uma rede neural artificial do tipo *MLP* foi treinada. A previsão da saúde financeira de uma determinada empresa é realizada submetendo-se os dados da mesma ao conjunto de regras gerados via *conjuntos aproximados*. No caso de não haver nenhuma regra capaz de gerar um resultado, os dados

são submetidos à rede neural artificial treinada. Os resultados do modelo final gerado pelo sistema híbrido foi capaz de classificar corretamente 94% das empresas enquanto que os modelos neurais utilizados para fins de comparação classificaram corretamente cerca de 90% das empresas.

Um outro sistema híbrido também composto por uma rede neural artificial e *conjuntos aproximados* foi proposto por Xiao et al. (2009). Contudo, diferentemente do modelo anterior, o componente principal desse sistema é o modelo neural final obtido que foi aplicado na previsão da demanda de carga elétrica cinco dias a frente. A teoria de *conjuntos aproximados* foi aplicada com o objetivo de realizar uma diminuição da dimensionalidade do problema que, originalmente, era composto por nove variáveis: duas observações de dias anteriores, temperaturas máximas e mínimas do dia, temperatura média no dia, maior e menor umidade relativa do ar e média da umidade relativa do ar no dia. Uma vez aplicada a teoria de *conjuntos aproximados*, escolheu-se o *reduto* com menor quantidade de *atributos* para projetar um conjunto de treinamento para uma rede neural *MLP*. Os resultados desse modelo neural foram comparados com modelos neurais obtidos a partir de um treinamento baseado no conjunto de dados composto por todos os atributos disponíveis. Como pôde ser verificado, o modelo obtido via sistema híbrido obteve melhores resultados.

A previsão sobre a quantidade de turistas foi o tema abordado em Goh e Law (2003). Neste trabalho, foram considerados seis atributos para determinar se o volume de turistas em Hong Kong, vindos de países como China, Estados Unidos, Inglaterra, Japão, Coreia, dentre outros, aumentaria ou diminuiria. Os atributos considerados foram: Produto Interno Bruto, Índice de Preços do Consumidor, população, volume de negócios, taxa de câmbio e o país de origem. Para determinar o modelo aproximado, considerou-se seis conjuntos de testes compostos por 131 amostras de turistas escolhidas aleatoriamente. Para fins de validação, 10 amostras foram reservadas. Os testes mostraram que o modelo conseguiu um acerto de 87,2 % na previsão de aumento ou queda do turismo na região.

Herbert e Yao (2009) aplicou a teoria de *conjuntos aproximados* na análise das ações de mercado da Nova Zelândia. A série temporal usada contemplava observações que começava em 31 de julho de 1991 e terminava em 27 de abril de 2007. Os dados disponíveis eram referentes ao preço de abertura e fechamento e ao maior e menor preços alcançados pela ação ao longo do dia. Para cada uma das observações, adicionou-se um *atributo de decisão* cuja a função era verificar se a ação deveria ser vendida, comprada ou se nada deveria ser feito. No total, dispunha-se de 1665 *objetos* para construir um modelo e 555 *objetos* para validação. Após aplicação da técnica de *conjuntos aproximados*, um total de

18 *redutos* foram encontrados. Desses *redutos*, selecionou-se aquele com menor número de *atributos*. Com um total de 10 regras, o modelo foi capaz de atingir uma precisão de aproximadamente 67%.

Uma revisão da literatura relacionada a aplicação da teoria de *conjuntos aproximados* em previsões na área de economia e finanças foi realizada por Tay e Shen (2002). Segundo os autores, *Previsão de Falhas em Negócios*, *Investimento Financeiro* e *Database Marketing* são três subáreas onde a teoria vem sendo frequentemente aplicada.

O objetivo da subárea relacionada a *Previsão de Falhas em Negócios*, como o próprio nome diz, é prever a possibilidade de uma empresa ir a falência ou não. Slowinski e Zopounidis (1995) citam, dentre outros, os trabalhos de Slowinski e Zopounidis (1995) e Dimitras et al. (1999). Slowinski e Zopounidis (1995) aplicou técnicas de *conjuntos aproximados* num *SD* composto de 12 *atributos* e 39 *objetos* referentes a registros de empresas para determinar a possibilidade ou não de falência. Já Dimitras et al. (1999) utilizaram informações de 40 *objetos* referentes a empresas com boa saúde financeira e 40 *objetos* referentes a empresas com saúde financeira debilitada. A análise dos 80 *objetos* e dos 12 *atributos* permitiu a construção de um modelo capaz de prever as possibilidades de falência das empresas. O trabalho de Ahn, Cho e Kim (2000) descrito anteriormente nessa seção é um outro exemplo de pesquisa que se enquadra nessa subárea.

O *Database Marketing* é a subárea cujo objetivo é analisar os dados de clientes a fim de descobrir padrões que permitam que estratégias relacionadas a esses clientes possam ser mais eficientemente traçadas e mais precisamente direcionadas. Tay e Shen (2002) citam, dentre outros, os trabalhos de Poel (2009) e Mrozek e Skabek (1998). Neste sentido, Poel (2009) considerou dados como comportamento, características pessoais e transações financeiras passadas de clientes de empresas para criar um modelo capaz de prever se esses seriam bons ou maus clientes no futuro. O modelo gerado obteve uma precisão de 74,35% contra 74,62% de um modelo estatístico aplicado para fins de comparação. Mrozek e Skabek (1998) usaram modelos baseados em *conjuntos aproximados* para analisar dados de clientes atuais a fim de criar padrões que descrevessem clientes típicos. Dessa forma, pôde-se prever se uma pessoa seria ou não um cliente em potencial caso esta enquadrasse-se nos perfis determinados.

Por fim, a subárea de *Investimento Financeiro* avalia as possibilidades de negociação de uma determinada ação. Exemplos de trabalhos relacionados são os de Shen e Loh (2003) e Herbert e Yao (2009), descritos anteriormente nessa seção.

4.4 Uma Proposta de Metodologia de Previsão Baseada em Conjuntos Aproximados

Conforme discutido nas seções anteriores, é comum encontrar aplicações diversas da teoria em problemas reais mas, em contrapartida, uma abordagem que considera a teoria em processos de previsão de séries temporais dificilmente é encontrada.

Dessa forma, o objetivo dessa seção é apresentar a abordagem proposta por Pinheiro et al. (2010a, 2010b) e propor uma extensão dessa abordagem a fim de gerar mecanismos através dos quais a previsão de uma série temporal possa ser realizada por meio de modelos originados a partir da teoria de *conjuntos aproximados*. Esses modelos serão denominados de *modelos aproximados*.

4.4.1 Conjuntos Aproximados como Aproximadores de Funções

Conforme discutido na seção 4.2, um *SD* pode ser representado em formato tabular conforme ilustrado na Tabela 1.

Os *atributos condicionais* (x_i) são variáveis cujos dados ($x_N^{(k)}$) estão relacionados a funções gerais definidas pelos valores ($y_{(k)}$) do *atributo de decisão* (y). Desta forma, do *SD* representado pela Tabela 1 derivam-se as sentenças representadas em (4.10). Por exemplo, para

$x_1 = x_1^{(k)}$, $x_2 = x_2^{(k)}$, $x_3 = x_3^{(k)}$, e $x_N = x_N^{(k)}$ tem-se $y = y^{(k)}$ definido por s_k . E para $x_1 = x_1^{(m)}$, $x_2 = x_2^{(m)}$, $x_3 = x_3^{(m)}$, e $x_N = x_N^{(m)}$ tem-se $y = y^{(m)}$ definido por s_m .

s_1 : SE $x_1=x_1^{(1)}$ E $x_2=x_2^{(1)}$ E ... E $x_n=x_n^{(1)}$ ENTÃO $y=y^{(1)}$

s_2 : SE $x_1=x_1^{(2)}$ E $x_2=x_2^{(2)}$ E ... E $x_n=x_n^{(2)}$ ENTÃO $y=y^{(2)}$

⋮

s_k : SE $x_1=x_1^{(k)}$ E $x_2=x_2^{(k)}$ E ... E $x_n=x_n^{(k)}$ ENTÃO $y=y^{(k)}$

⋮

s_m : SE $x_1=x_1^{(m)}$ E $x_2=x_2^{(m)}$ E ... E $x_n=x_n^{(m)}$ ENTÃO $y=y^{(m)}$

s_v : SE $x_1=x_1^{(v)}$ E $x_2=x_2^{(v)}$ E ... E $x_n=x_n^{(v)}$ ENTÃO $y=y^{(v)}$

(4.10)

Considerando-se valores intermediários aos dados representados na Tabela 1, ou seja, $x_1^{(k)} \leq x_1 \leq x_1^{(m)}$, $x_2^{(k)} \leq x_2 \leq x_2^{(m)}$, $x_3^{(k)} \leq x_3 \leq x_3^{(m)}$ e $x_N^{(k)} \leq x_N \leq x_N^{(m)}$, as regras

definidas em (4.10) podem ser redefinidas conforme apresentado em (4.11) ou na forma de (4.12).

$$\begin{aligned} IF x_1^{(k)} \leq x_1 \leq x_1^{(m)} \text{ AND } x_2^{(k)} \leq x_2 \leq x_2^{(m)} \text{ AND } \dots \text{ AND} \\ x_N^{(k)} \leq x_N \leq x_N^{(m)} \text{ THEN } \min\{y^{(k)}, \dots, y^{(m)}\} \leq y \leq \max\{y^{(k)}, \dots, y^{(m)}\} \end{aligned} \quad (4.11)$$

$$\begin{aligned} IF x_1 = [x_1^{(k)}, x_1^{(m)}] \text{ AND } x_2 = [x_2^{(k)}, x_2^{(m)}] \text{ AND } \dots \text{ AND} \\ x_N = [x_N^{(k)}, x_N^{(m)}] \text{ THEN } y = [y^{(k)}, y^{(m)}] \end{aligned} \quad (4.12)$$

Dessa forma, caso o *SD* considerado for composto por um único *atributo condicional* (x_1) de valores numéricos, a equação (4.13) pode ser aplicada para estimar os valores intermediários correspondentes.

$$y = y^{(k)} + (y^{(m)} - y^{(k)}) \cdot \frac{(x_1 - x_1^{(k)})}{(x_1^{(m)} - x_1^{(k)})} \quad (4.13)$$

No caso em que o *SD* considerado for composto por dois atributos condicionais (x_1 e x_2) de valores numéricos, a equação (4.14), uma extensão de (4.13), pode ser aplicada para determinar os valores intermediários correspondentes.

$$y = y^{(k)} + \frac{(y^{(m)} - y^{(k)})}{2} \cdot \left(\frac{(x_1 - x_1^{(k)})}{(x_1^{(m)} - x_1^{(k)})} + \frac{(x_2 - x_2^{(k)})}{(x_2^{(m)} - x_2^{(k)})} \right) \quad (4.14)$$

Para um *SD* composto por um número n de *atributos condicionais* numéricos, a forma (4.15) constitui uma extensão válida que produz valores iguais à fórmula de interpolação linear de Lagrange, sendo que

$n = 1, 2, \dots, N$ e $i = k, m$.

$$y = (x_n, x_n^{(i)}, y^{(i)}) = y^{(k)} + \frac{(y^{(m)} - y^{(k)})}{N} \cdot \sum_{n=1}^N \frac{(x_n - x_n^{(k)})}{(x_n^{(m)} - x_n^{(k)})} \quad (4.15)$$

É justamente esse tipo de modelagem e utilização de interpolação uma das contribuições de Pinheiro et al. (2010a, 2010b) para a abordagem que considera a aplicação dos *conjuntos aproximados* em procedimentos de modelagem de sistemas dinâmicos e

aproximações de funções. Esta abordagem será adaptada nesta dissertação visando a sua aplicação para a previsão de séries temporais.

4.4.2 Aplicação do Método em Processos de Previsão

A primeira etapa da aplicação da técnica descrita na seção 4.4 em problemas de previsão consiste em definir um *SD* que represente os dados da série temporal na forma ilustrada na Tabela 2.

Tabela 2: Representação de um *SD* para previsão de um passo à frente.

\mathbf{x}_1	\mathbf{x}_2	\mathbf{x}_3	\dots	\mathbf{x}_n	\mathbf{y}
a_1	a_2	a_3	\dots	a_n	a_{n+1}
a_2	a_3	a_4	\dots	a_{n+1}	a_{n+2}
a_3	a_4	a_5	\dots	a_{n+2}	a_{n+3}
\dots	\dots	\dots	\dots	\dots	\dots
a_{N-n-2}	a_{N-n-1}	a_{N-n}	\dots	a_{N-1}	a_N

Como pode ser verificado a partir da Tabela 2, os *atributos condicionais* correspondem às observações do passado a partir de um instante n . Uma amostra dessas observações quando apresentadas ao modelo deveria gerar como saída a previsão no instante $n+1$ que, no *SD* em questão, corresponde ao *atributo de decisão*. Caso haja a necessidade de se representar uma variável exógena, a mesma poderia ser inserida no *SD* como um *atributo condicional* adicional.

É fácil perceber a semelhança do *SD* representado pela Tabela 2 com os conjuntos de dados utilizados para o treinamento das redes neurais artificiais e dos sistemas *neuro-fuzzy* frequentemente utilizados em processos de previsão.

Conforme discutido na seção 4.4, a proposta deve ser aplicado em *SDs* compostos por valores numéricos, já que trabalhar com observações de séries temporais envolve manipulação de dados numéricos. Contudo, a aplicação dos conceitos de *conjuntos aproximados* é realizada em *SDs* cujos dados já encontram-se discretizados. Sendo assim, a segunda etapa da proposta consiste em discretizar os dados do *SD* gerado na primeira etapa via um método adequado de discretização. A partir desse ponto, as regras de decisão do *SD* podem ser obtidas via aplicação da teoria de *conjuntos aproximados* caracterizando a terceira etapa do processo.

Uma vez determinadas as regras, o *modelo aproximado* pode ser obtido. Esse modelo será utilizado como base de conhecimento que tentará representar o comportamento de

uma série temporal e assim inferir valores futuros da mesma.

O processo de previsão consiste em apresentar para o *modelo aproximado* obtido, um padrão de entrada semelhante ao padrão formado pelos *atributos de decisão* que constituem o *SD* inicial (Tabela 2). Para uma regra ativada, aplica-se a equação (4.15) a fim de determinar um valor de saída que é resultado da interpolação entre os valores mínimos e máximos do conseqüente dessa regra. Esse valor de saída é o resultado da previsão em um determinado instante t .

Caso mais de uma regra seja ativada, o valor de saída é dado pela média aritmética de todos os valores obtidos pela equação (4.15) para cada uma das regras ativadas.

O valor inferido no instante t , juntamente com as $t - n$ observações anteriores, servirá como entrada para a previsão no passo seguinte ($t + 1$). Tal interação é repetida até que o número de passos de previsão desejados seja processado.

Os procedimentos da metodologia estão representados na Figura (9).

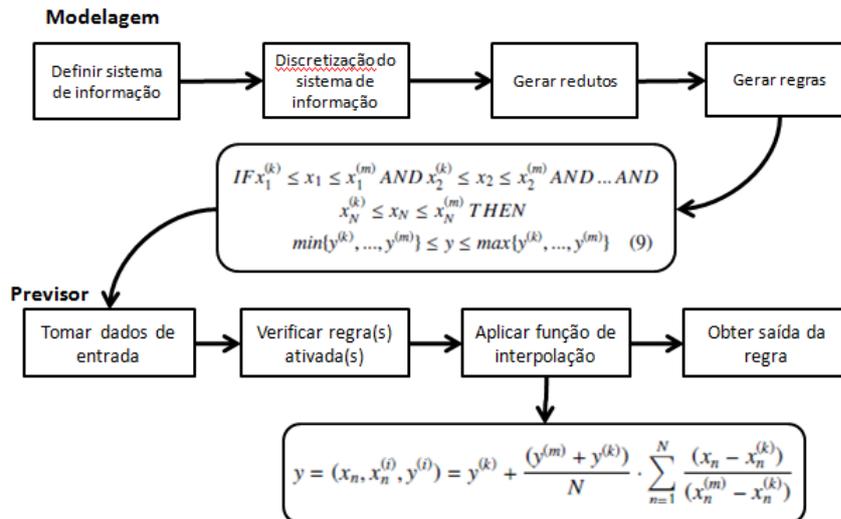


Figura 9: Esquema geral da metodologia proposta.

4.5 Considerações Sobre a Aplicação da Metodologia

É comum a comunidade científica da área recorrer à ferramentas de apoio capazes de processar os conceitos de *conjuntos aproximados* visando uma maior eficiência no processo de obtenção dos redutos e das regras que caracterizam os modelos.

Como exemplo de tais ferramentas, pode-se citar a *Grobian* desenvolvida por Ivo

Düntsch da Universidade de Osnabruek e Günther Gediga da Universidade de Ultesr; a *Rough Enough* desenvolvida por Anders Torvill Bjorvand; a *Rough Sets Library* desenvolvida por M. Gawrys e J. Sienkiewicz da Universidade Warsaw de Tecnologia, Polônia e a biblioteca *MLC++* cujo projeto foi iniciado na Universidade de Stanford, Estado Unidos.

A fim de aplicar a metodologia proposta neste capítulo, optou-se em utilizar a ferramenta *Rosetta* (OHRN; KOMOROWSKI, 1997). O *Rosetta* é um aplicativo de fácil utilização e que fornece variados recursos para processar conjuntos aproximados como meios para criação dos *SD*'s, discretização de dados numéricos e construção de *redutos* e regras, atendendo adequadamente às necessidades da proposta.

5 Metodologia de Processos de Previsão

De uma forma geral, os processos de previsão passam por certas etapas que, independente do método empregado, devem ser consideradas e analisadas.

Neste trabalho, qualquer processo de previsão é composto pelas seguintes etapas:

- Definição dos conjuntos de treinamento;
- Normalização dos dados;
- Pré processamento dos dados;
- Configuração do sistema previsor a ser empregado;
- Previsão e análise dos erros de previsão.

5.1 Definição dos conjuntos de treinamento e teste

Um conjunto de treinamento é composto por vários elementos de dados. Cada elemento do conjunto de treinamento pode ser dividido em duas partes. A primeira parte, denominada de vetor de entrada, é referente ao padrão de entrada para o previsor. A segunda, denominada de vetor de saída, está relacionada ao padrão de saída esperado a partir da apresentação do padrão de entrada correspondente.

O vetor de entrada de cada elemento de treinamento consiste em um vetor cujos elementos são observações do passado e, caso necessário, variáveis exógenas. Esses componentes serão os responsáveis pela previsão do valor seguinte.

O número de componentes do vetor de saída de cada elemento de treinamento é determinado de acordo com o tipo de previsão a ser considerado. Segundo Zhang, Patuwo e Hu (1998) existem dois tipos de previsão: *um passo a frente* ou *múltiplos passos a frente*.

Na abordagem um passo a frente, o vetor de saída é composto por uma única componente. Nesta abordagem, os valores previstos no instante t servem como entrada para previsão no passo $t + 1$ seguinte. Na abordagem *múltiplos passos a frente*, também conhecido como método direto, o número de componentes do vetor de saída é igual ao número de valores que se deseja prever.

Desta forma, um exemplo de vetor de entrada seria um composto pelas observações $x(t), x(t-1), x(t-2), \dots, x(t-n)$ juntamente com as variáveis exógenas consideradas, onde n corresponde ao número de observações do passado utilizados. Para o vetor de saída, teria-se um conjunto B de valores previstos, sendo que $B = \{x(t+1), x(t+2), \dots, x(t+k)\}$, onde k é o número de valores que se deseja prever caso a previsão seja de *múltiplos passos a frente*. No caso da previsão *um passo a frente*, tem-se $B = \{x(t+1)\}$

O tipo de previsão empregado em todos os experimentos desse trabalho foi o de um *passo a frente* sendo que algumas variáveis exógenas foram consideradas em alguns estudos.

Para avaliação dos modelos obtidos adotou-se um conjunto de testes que consiste em algumas observações, geralmente as últimas da série, que são reservadas exclusivamente para fins de testes. Vale ressaltar que essas informações não são fornecidas ao sistema predictor durante o processo de treinamento.

5.2 Normalização dos dados

A segunda etapa da maioria dos processos de previsão, principalmente daqueles que envolvem a aplicação de técnicas de inteligência artificial é a análise da necessidade ou não de uma transformação inicial nos dados da série temporal de modo que a mesma enquadre-se na arquitetura do método de previsão escolhido. Por exemplo, os modelos estatísticos têm dificuldades em operar em séries cuja componente de tendência é acentuada. Dessa forma, uma normalização que reduza a tendência, como a *diferenciada* ou a *logarítmica*, pode ser considerada (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998). Um outro exemplo são os modelos neurais cuja função de ativação na camada de saída não seja a linear. Nesses casos, uma normalização linear para enquadrar os dados em um intervalo definido pode ser considerada (ZHANG; PATUWO; HU, 1998).

Os principais tipos de normalização de uma série temporal, que são comumente empregados em trabalhos relacionados e que inclusive foram considerados nesse trabalho são a normalização linear, normalização linear para um intervalo $[a,b]$, normalização estatís-

tica, normalização logarítmica e normalização diferenciada, dadas pelas equações (5.1), (5.2), (5.3), (5.4) e (5.5) respectivamente.

$$y_n = \frac{(x_n - x_{min})}{(x_{max} - x_{min})} \quad (5.1)$$

$$y_n = (b - a) \cdot \frac{(x_n - x_{min})}{(x_{max} - x_{min})} + a \quad (5.2)$$

$$y_n = \frac{(x_n - \bar{x})}{s} \quad (5.3)$$

$$y_n = \ln\left(\frac{x_n}{x_{n-1}}\right) \quad (5.4)$$

$$y_n = x_n - x_{n-1} \quad (5.5)$$

A aplicação da normalização linear resulta em uma série cujas observações estão inseridas no intervalo que vai de zero a um. Já a normalização linear para intervalos enquadra as observações da série em um intervalo [a,b] definido. A normalização estatística utiliza-se da média e do desvio padrão das observações e é capaz de suavizar a componente de tendência da série. A normalização logarítmica, também conhecida como *retornos*, é frequentemente empregada em séries financeiras cujo comportamento é, na maioria das vezes, aleatório. Por fim, a normalização da diferença, inspirada no método *ARIMA* descrito na seção 2.2.3.4, é aplicada com a finalidade de suavizar a aleatoriedade presente nas séries. Vale ressaltar que não foi encontrado nenhum trabalho que se utilizasse do método em questão para normalização de dados de treinamento, porém, como pode ser visto no capítulo 6 bons resultados foram obtidos.

Os três sistemas baseados em inteligência artificial considerados nesse trabalho foram executados, na maioria das vezes, sobre dados normalizados segundo os procedimentos anteriormente descritos.

5.3 Pré Processamento dos Dados

O pré processamento dos dados foi empregado, nesse trabalho, no sistema *neuro-fuzzy* de previsão e no sistema baseado em *conjuntos aproximados*.

Conforme discutido na seção 3.4.1, o *ANFIS* pode sofrer de problemas de dimensionalidade resultando em modelos compostos por um número de regras tão grande que o uso do modelo obtido pode se tornar inviável. A fim de evitar esse problema, um *agrupamento* por meio do algoritmo *fuzzy c-means* foi considerado.

Para os modelos aproximados, o pré processamento consiste em discretizar o *SD* de forma a tornar os dados que o compõe nominais, e assim possibilitar a aplicação das técnicas de *conjuntos aproximados*.

5.4 Configuração do sistema predictor a ser empregado

A quarta etapa consiste da configuração dos parâmetros de treinamento do sistema predictor aplicado no processo de previsão.

Independente do sistema empregado, o conjunto de dados utilizado para construção do modelo final de previsão é composto pelos mesmos dados definidos na primeira etapa. Para os sistemas neurais e *neuro-fuzzy*, a quantidade de neurônios na camada de entrada é dada pelo número de observações do passado agregados com possíveis variáveis exógenas consideradas. O número de observações do passado juntamente com as variáveis exógenas definirá também a quantidade de *atributos condicionais* do *SD* utilizado para determinação dos modelos aproximados. Nesse trabalho, todas as redes neurais e sistemas *neuro-fuzzy* empregados possuem um único neurônio na camada de saída. O valor desse neurônio corresponde à previsão no passo $t + 1$. Esse valor nos *SD*'s é dado pelo *atributo de decisão*.

Quando a previsão é realizada por meio de um sistema neural *MLP*, parâmetros específicos como a quantidade de camadas intermediárias, número de neurônios em cada camada, função de ativação em cada camada, taxa de aprendizagem e termo de momento devem ser definidos.

Nesse trabalho, as redes neurais *MLP* são todas compostas por duas camadas intermediárias sendo que o número de neurônios de cada camada varia de acordo com a série temporal em estudo. Os demais parâmetros também são ajustados experimentalmente de forma a minimizar o erro o máximo possível.

Para o sistema *neuro-fuzzy*, os parâmetros essenciais são o número de *agrupamentos* para o algoritmo *fuzzy c-means*, a taxa de aprendizagem para o algoritmo de treinamento na fase *forward* e o tipo de função de pertinência encapsulada nos neurônios adaptativos

da primeira camada.

Todos os modelos *fuzzy* empregados nos processos de previsão foram gerados por um *ANFIS* cujos neurônios na primeira camada implementam a função gaussiana de pertinência.

Os *modelos aproximados* foram obtidos com o auxílio da ferramenta *Rosetta* (OHRN; KOMOROWSKI, 1997). A ferramenta, dentre outros recursos, implementa uma série de algoritmos de discretização. Em todos os testes realizados, o algoritmo de discretização aplicado foi o *Equal Frequency Bining*. Tal algoritmo divide um conjunto de dados em intervalos iguais e insere cada dado em seu intervalo correspondente. O número de intervalos é o parâmetro a ser definido no algoritmo.

5.5 Previsão e análise dos erros de previsão

Uma vez que um modelo neural, *fuzzy* ou aproximado foi gerado, a etapa de previsão pode ser iniciada. O processo de previsão é realizado conforme descrito na seção 5.1.

Realizada a previsão, o resultado obtido é comparado ao conjunto de teste a fim de que a precisão do modelo predictor seja avaliada.

A precisão de cada modelo de previsão é avaliada por meio de índices de desempenho. Os índices mais frequentemente empregados são: *Mean Absolute Deviation (MAD)*, *Sum of Squared Error (SSE)*, *Mean Squared Error (MSE)*, *Mean Percentage Absolute Error (MAPE)* e desvio padrão, definidos pelas equações (5.6), (5.7), (5.8), (5.9) e (5.10) respectivamente.

$$E = \frac{\sum |x_n - y_n|}{N} \quad (5.6)$$

$$E = \sum (x_n - y_n)^2 \quad (5.7)$$

$$E = \frac{\sum (x_n - y_n)^2}{N} \quad (5.8)$$

$$E = \frac{1}{N} \sum \left| \frac{x_n - y_n}{x_n} \right| \quad (5.9)$$

$$E = \sqrt{\frac{(x - y)^2}{n - 2}} \quad (5.10)$$

Onde x_n corresponde ao n-ésimo elemento do conjunto de teste e y_n à n-ésima observação prevista.

Nos experimentos realizados, todos os índices de erros foram empregados quando não era objetivo a comparação entre os resultados de um estudo com os resultados de outros trabalhos.

6 Experimentos

A fim de testar a precisão dos modelos gerados a partir das técnicas de inteligência artificial discutidas nos capítulos anteriores, cinco séries temporais práticas foram consideradas.

A escolha de cada uma das séries em questão foi motivada por características peculiares inerentes a cada uma delas. Pretende-se com isso, analisar o desempenho de cada modelo na modelagem e previsão de séries com diferentes formas e comportamentos. Além disso, muitas das séries consideradas nesse trabalho foram analisadas em outros trabalhos e sendo assim, a comparação entre os diversos resultados obtidos também é de interesse.

A primeira série temporal a ser considerada consiste em uma série referente ao consumo de café no Brasil. Esta série foi analisada por Bacci (2007) onde diversos métodos estatísticos de previsão foram utilizados. Tal série apresenta uma forte componente de tendência como característica marcante. Portanto, a motivação para se trabalhar com esta série consiste em, além de comparar os modelos propostos nesse trabalho com os métodos estatísticos aplicados em Bacci (2007), em afirmar se os modelos baseados em inteligência artificial são capazes de modelar e prever satisfatoriamente séries com componente de tendência acentuada.

A segunda série temporal considerada refere-se ao preço da arroba do boi gordo. Freiman (2006) analisou esta série por meio de modelos baseados em redes neurais e modelos de auto regressão múltipla. Foi constatado que Freiman (2006), de certa forma, falhou na tentativa de definir uma configuração ótima para sua rede neural no sentido de não ter explorado todas as possibilidades de definição de parâmetros para a construção da mesma.

Uma série referente à demanda de carga elétrica é a terceira série temporal considerada nesse trabalho. Carpinteiro et al. (2007) realizaram diversas análises e previsões por meio de experimentos neste conjunto de dados que, além do histórico de observações, contava com a presença de integradores temporais. Essa série temporal apresenta uma

forte componente sazonal, sendo assim, a relevância de se estudar esta série consiste em verificar a precisão dos modelos deste trabalho quando aplicados neste tipo de séries temporais e em comparar os resultados obtidos por meio da aplicação dos modelos aqui propostos com os modelos neurais propostos em Carpinteiro et al. (2007).

Um outro objeto de estudo considerado refere-se a uma série temporal que representa a atividade pluviométrica na cidade de Itajubá, localizada no sul do estado das Minas Gerais. Esta série temporal, semelhante à de demanda de carga elétrica, apresenta uma forte componente sazonal. A possibilidade de integração dos modelos gerados para análise desta série com sistemas de alerta de enchentes foi uma das motivações para a construção de modelos para este conjunto de dados.

Por fim, a quinta série temporal considerada consiste em uma série temporal financeira. Como é comum em séries dessa categoria, esta série temporal apresenta um comportamento bastante aleatório caracterizado pela presença de fortes componentes sazonal e de tendência. A motivação em se trabalhar com esta série, portanto, consiste em verificar se existem modelos baseados nas técnicas propostas neste trabalho capazes de modelar e prever séries muito mal comportadas.

6.1 Experimento 1 - Série de Consumo de Sacas de Café no Brasil

Resultados da aplicação de métodos estatísticos de previsão para a série de consumo de sacas de café no Brasil podem ser encontrados em Bacci (2007). Neste trabalho, dentre outros métodos, um *ARIMA* (2,2,5) foi aplicado para previsão dois passos a frente. O trabalho também propõe um método cujo o resultado da previsão de uma determinada observação da série no instante t é dado pela média aritmética dos resultados obtidos a partir da aplicação de três métodos estatísticos de previsão (*ARIMA* + *Análise de Tendência Quadrática* (ATQ) + *Dupla Suavização Exponencial* (DSE)). A equação (6.1) ilustra o método citado que foi denominado “*COMB*”.

$$Y_{t(COMB)} = \frac{Y_{t(ARIMA(2,2,5))} + Y_{t(DSE)} + Y_{t(ATQ)}}{3} \quad (6.1)$$

A série temporal desse experimento está representada pela Figura 10 e pela Tabela 3, ou seja, observações anuais de consumo de sacas de café de 1975 até 2004. Deseja-se prever o consumo para 2005 e 2006.

Para efetuar previsões, um modelo baseado em regras *fuzzy* foi proposto. A fim de comparar a eficiência do modelo *fuzzy*, um modelo baseado em redes neurais também foi desenvolvido. Por fim, a eficiência dos modelos baseados em inteligência artificial será confrontada comparando-se os resultados obtidos com os resultados de Bacci (2007).

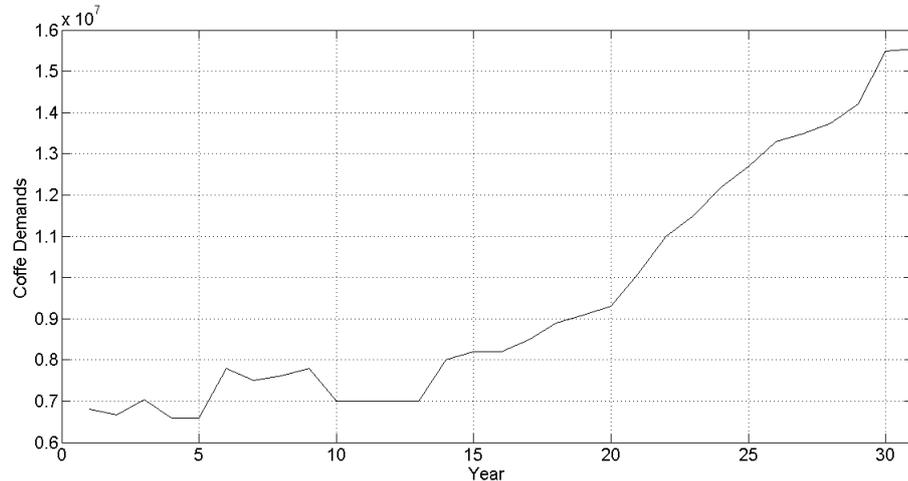


Figura 10: Gráfico da série de consumo de café.

Tabela 3: Tabela com as observações da série de consumo de sacas de café.

Ano	Sacas	Ano	Sacas	Ano	Sacas	Ano	Sacas
1975	6.795.000	1983	7.800.000	1991	8.500.000	1999	12.700.000
1976	6.661.000	1984	7.000.000	1992	8.900.000	2000	13.289.000
1977	7.036.000	1985	7.000.000	1993	9.100.000	2001	13.490.000
1978	6.600.000	1986	7.000.000	1994	9.300.000	2002	13.750.000
1979	6.600.000	1987	7.000.000	1995	10.100.000	2003	14.200.000
1980	7.800.000	1988	8.000.000	1996	11.000.000	2004	15.490.000
1981	7.500.000	1989	8.200.000	1997	11.500.000		
1982	7.624.000	1990	8.200.000	1998	12.200.000		

6.1.1 Modelo Fuzzy Aplicado à Previsão da Série de Demanda de Café

Para determinar um modelo baseado em regras *fuzzy*, realizou-se primeiro, uma normalização nas observações da série temporal. O processo de normalização empregado consistiu em duas etapas. Na primeira, realizou-se uma diminuição na ordem das observações dividindo-se todas por 10.0000.0000. Na segunda aplicou-se a normalização da diferença por meio da equação (6.2) conforme descrito na seção 5.2.

$$Z_t = Y_t - Y_{t-1} \quad (6.2)$$

Ao fim do processo de normalização, obteve-se os dados que compõem os elementos do conjunto de treinamento. O conjunto de treinamento é composto por 25 elementos onde cada elemento é constituído por quatro dados de entrada ($a_t, a_{t-1}, a_{t-2}, a_{t-3}$) para previsão um passo a frente (a_{t+1}).

Para utilização do *ANFIS* nessa série, aplicou-se o algoritmo *Fuzzy C-Means* nos dados de treinamento normalizados a fim de agrupá-los em 15 *clusters*. Iniciou-se o processo de treinamento com 100 épocas a uma taxa de aprendizado de 0,01. As funções de pertinência para os neurônios da primeira camada do *ANFIS* foram todas do tipo gaussiana. Obteve-se no final, um modelo composto por 15 regras *fuzzy*. A Tabela 4 mostra os parâmetros das funções gaussianas para cada variável de entrada do modelo. A Tabela 5 mostra os parâmetros das funções lineares que representam os consequentes de cada regra.

Tabela 4: Antecedentes das regras do modelo fuzzy para série de consumo de café.

	t		$t - 1$		$t - 2$		$t - 3$	
	a	b	a	b	a	b	a	b
R1	0,014945	-0,002214	0,012002	0,000131	0,023476	0,099961	0,011103	0,016818
R2	0,010662	0,050569	0,008324	0,058030	0,008942	0,020194	0,009870	0,026944
R3	0,012396	0,001382	0,025444	0,119226	0,018881	-0,029339	0,007713	0,014258
R4	0,012328	0,018708	0,032209	-0,077859	0,012200	-0,000035	0,012324	0,000984
R5	0,011349	0,030094	0,011726	0,039570	0,011289	0,020060	0,007465	0,019798
R6	0,014446	-0,027492	0,008510	0,013870	0,008077	0,019901	0,031312	-0,079234
R7	0,014024	0,000571	0,023930	0,098011	0,011189	0,020095	0,012448	0,000598
R8	0,015184	0,023229	0,010890	0,020451	0,027604	0,075430	0,019647	0,087840
R9	0,026431	0,051053	0,017004	0,014737	0,020543	0,023046	0,024023	0,067598
R10	0,025340	0,088062	0,015668	0,052340	0,012110	0,072853	0,002866	0,048921
R11	0,018127	0,056406	0,015119	0,070040	0,008349	0,051766	0,019802	0,057485
R12	0,044460	0,015995	0,023897	0,008524	0,019509	0,004939	0,035086	0,090992
R13	0,020960	-0,016615	0,015843	0,034994	0,022725	-0,042009	0,011444	0,000014
R14	0,008513	0,019990	0,011674	-0,000568	0,012299	0,029075	0,011441	0,039690
R15	0,014629	0,000823	0,011825	0,029246	0,013118	0,039687	0,010202	0,019854

A Figura 11 ilustra a adequação do modelo *fuzzy* aos dados de treinamento normalizados. Conforme mostrado pelo gráfico da figura, pode-se verificar que a modelagem da série via o modelo empregado é bastante satisfatória. Um erro médio quadrático de $2.42 \cdot 10^{-10}$ foi conseguido.

Os resultados da previsão obtidos pelo modelo *fuzzy* foram desnormalizados para que as comparações com os resultados de Bacci (2007) pudessem ser realizadas.

A discussão dos resultados da previsão pode ser encontrada na seção 6.1.3.

Tabela 5: Consequentes das regras do modelo fuzzy para série de consumo de café.

	a	b	c	d	e
R1	-0,112644	0,000001	0,051414	-0,129230	-0,002561
R2	0,002128	0,002506	0,000860	0,001103	0,042889
R3	-0,000279	0,000579	-0,000732	-0,000016	0,017508
R4	0,000000	0,000001	0,000000	0,000000	0,000001
R5	0,002495	0,003315	0,001645	0,001651	0,082089
R6	-0,000038	-0,000009	-0,000013	0,000060	0,000004
R7	0,000004	0,002965	0,000595	0,000003	0,029746
R8	0,002834	0,000886	-0,002389	0,003523	0,048996
R9	-0,046980	0,011222	0,008043	0,062579	0,164533
R10	-0,123409	0,058549	0,041431	0,003183	0,063942
R11	0,647415	0,416208	0,000881	0,218138	-0,054563
R12	-0,879583	0,276088	-0,036025	-0,059054	0,021688
R13	-1,306613	0,996235	1,825604	0,000001	0,144765
R14	-0,000039	0,000001	-0,000060	-0,000079	-0,001993
R15	0,000071	0,000564	0,000673	0,000360	0,018007

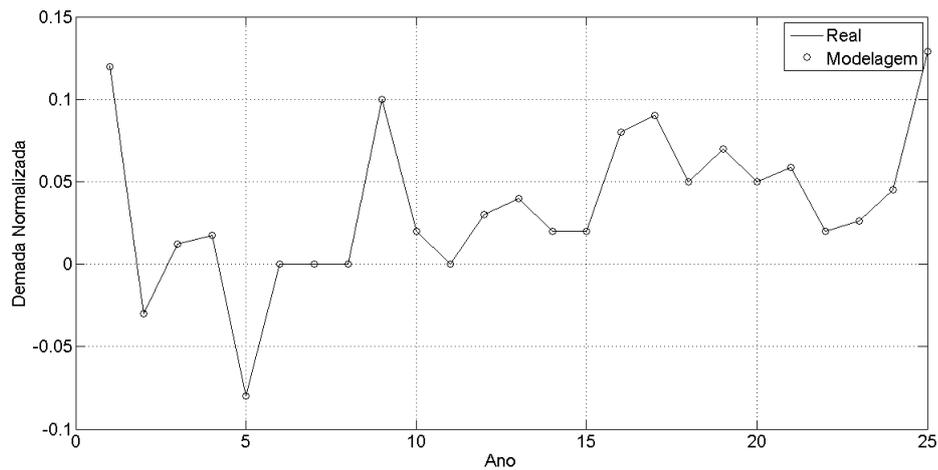


Figura 11: Gráfico da adequação do modelo fuzzy aos dados de treinamento da série de consumo de café.

6.1.2 Modelo Neural Aplicado à Previsão da Série de Consumo de Café

O modelo neural empregado na previsão da série de consumo de café é composto por quatro neurônios na camada de entrada, duas camadas intermediárias com 15 e 25 neurônios cada uma e um neurônio na camada de saída. A função de ativação utilizada nas duas camadas intermediárias foi a tangente hiperbólica. Para a camada de saída, aplicou-se uma função linear. O treinamento do modelo foi realizado em 2400 épocas a uma taxa de aprendizado de 0,02, e termo de momento de 0,9.

O conjunto de dados de treinamento, composto por 25 elementos, foi gerado aplicando-se uma normalização linear nas observações da série (Tabela 3). Cada elemento do conjunto de treinamento é composto por quatro dados de entrada ($a_t, a_{t-1}, a_{t-2}, a_{t-3}$) para previsão um passo a frente (a_{t+1}).

Os resultados obtidos pelo modelo neural foram desnormalizados para que as comparações com os resultados de Bacci (2007) pudessem ser realizadas. A tabela 6 mostra os resultados dos modelos considerados.

A discussão dos resultados da previsão pode ser encontrada na seção 6.1.3 juntamente com os resultados obtidos a partir da aplicação do modelo *fuzzy* e dos métodos de Bacci (2007).

6.1.3 Discussão dos Resultados (Série Consumo de Café)

Os resultados da aplicação dos modelos *fuzzy* e neural desse trabalho bem como o resultado da aplicação dos métodos estatísticos de Bacci (2007) estão sumarizados na Tabela (6).

Tabela 6: Tabela de erros da previsão da série de consumo de sacas de café.

Valores da Previsão (milhões de sacas)		
Método de previsão	Previsão para 2005	Previsão para 2006
Valor Real (Milhões)	15,540	16,330
Arima(2,2,5)	16,400	16,890
Comb	16,150	16,750
Fuzzy	15,862	16,336
RNA	15,107	16,470
Erros da Previsão (%)		
Método de previsão	Previsão para 2005	Previsão para 2006
Arima(2,2,5)	5,534	3,429
Comb	3,925	2,572
Fuzzy	2,073	0,035
RNA	2,787	0,857

A partir da análise da Tabela 6, pode-se verificar facilmente que os modelos gerados a partir da aplicação de técnicas da inteligência artificial resultaram em erros menores que os estatísticos. Verifica-se também que o modelo baseado em regras *fuzzy* apresentou resultados bastante satisfatórios sendo, dentre os modelos empregados nesse experimento, aquele que obteve melhores resultados. Dessa forma, pode-se concluir que modelos *fuzzy* podem ser aplicados em processos de previsão de séries temporais cuja componente de tendência se faz presente de forma bastante acentuada.

6.2 Experimento 2 - Série de Preço da Arroba de Boi

Freiman (2006) utilizou-se de modelos de auto regressão múltipla e de uma rede neural artificial para fazer previsões do preço da arroba de boi gordo em seu trabalho.

Neste trabalho considerou-se observações mensais de preços a partir de abril de 1998 até fevereiro de 2004 para compor o conjunto de treinamento utilizado nos experimentos do trabalho em questão. Para fins de validação do modelo, as observações de março, abril e maio de 2004 foram reservadas a fim de compor um conjunto de testes para os modelos obtidos. O gráfico da Figura 12 ilustra a série deste estudo.

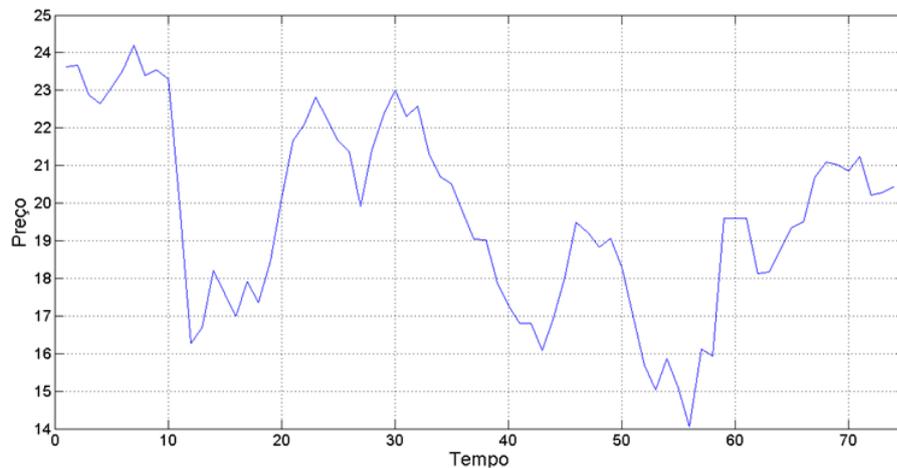


Figura 12: Gráfico da série de preços da arroba do boi gordo.

Além das observações referentes ao preço, outras informações relevantes ao mercado estavam disponíveis. O índice *IGPM* e a taxa *Selic* foram variáveis exógenas consideradas.

Dessa forma, a massa de dados que compõe os elementos do conjunto de treinamento para construção dos modelos tomavam duas observações de preço referentes aos dois meses anteriores ao mês t (a_{t-1} e a_{t-2}), a taxa *Selic* do mês anterior ($Selic_{t-1}$), os índices *IGPM* referentes aos dois meses anteriores ($IGPM_{t-1}$ e $IGPM_{t-2}$), e uma informação que identifica o mês t cujo valor do preço desejava-se prever.

A partir de uma análise dos dados considerados, Freiman (2006) constatou a existência de multicolinearidade entre as variáveis a_{t-1} e a_{t-2} e considerou utilizar, além do conjunto original, um conjunto onde a variável a_{t-2} não se fizesse presente a fim de que um outro modelo de regressão múltipla fosse determinado.

Ambos os conjuntos de treinamento são aplicados na construção dos modelos que compõem os testes desse experimento.

Esta seção mostra que a rede neural artificial proposta em Freiman (2006) não é a ótima para o processo de previsão da série em estudo; que modelos baseados em regras *fuzzy* cujos erros de previsão são compatíveis com os erros dos outros modelos considerados podem ser conseguidos; e, finalmente, que modelos baseados em regras *fuzzy* são eficientes quando aplicados a processos de previsão onde variáveis exógenas fazem-se presentes.

6.2.1 Modelo Fuzzy Aplicado à Previsão da Série de Preço da Arroba de Boi

Os dados de treinamento para obtenção do modelo *fuzzy* desse teste não sofreram qualquer tipo de normalização. Tanto para a construção do modelo onde as duas observações sobre o preço (a_{t-1} e a_{t-2}) foram consideradas quanto para aquele onde foi considerada somente uma, nenhuma normalização foi empregada.

Para o modelo que considera ambas observações de preço (aqui denominado de *Fuzzy 2T*), agrupou-se o conjunto de treinamento por meio da aplicação do algoritmo *Fuzzy C-Means*, em 50 *clusters* distintos. Ao *FIS* inicial obtido, empregou-se um processo de treinamento com 100 épocas a uma taxa de aprendizado de 0,02. Todos os neurônios da primeira camada tiveram como funções de pertinência correspondentes, funções do tipo gaussiana. Obteve-se no final, um *Sistema de Inferência Fuzzy* composto por 50 regras.

Para o modelo que considera apenas uma informação do preço, a_{t-1} (aqui denominado de *Fuzzy 1T*), agrupou-se o conjunto de treinamento por meio da aplicação do algoritmo *Fuzzy C-Means*, em 65 *clusters* distintos. Ao *FIS* inicial obtido, empregou-se um processo de treinamento com 100 épocas a uma taxa de aprendizado de 0,01. Todos os neurônios da primeira camada tiveram como funções de pertinência correspondentes, funções do tipo gaussiana. Obteve-se no final, um *Sistema de Inferência Fuzzy* composto por 65 regras.

Os resultados da previsão via aplicação dos modelos gerados nesse teste estão disponíveis na seção 6.2.3 onde discussões e comparações desses resultados juntamente com os resultados da aplicação dos outros modelos considerados também podem ser encontrados.

6.2.2 Modelo Neural Aplicado à Previsão da Série de Preço da Arroba de Boi

A fim de gerar uma rede neural artificial capaz de adequar-se aos dados de treinamento, uma normalização linear foi aplicada a todos os atributos que constituem os elementos do conjunto.

Para as variáveis a_{t-1} e a_{t-2} empregou-se uma normalização linear para o intervalo de -0,5 a 0,5. As variáveis $Selic_{t-1}$, $IGPM_{t-1}$ e $IGPM_{t-2}$ foram normalizadas para enquadrar-se no intervalo de 0 a 1.

De forma análoga ao experimento com modelos *fuzzy*, dois conjuntos de treinamento foram considerados. Um com duas informações referentes ao preço e outro com apenas uma informação referente ao preço.

Para o modelo que considera ambas observações de preço (aqui denominado de *RNA 2T*), uma *MLP* composta por quatro neurônios na camada de entrada, duas camadas intermediárias com 15 neurônios cada uma e um neurônio na camada de saída foi projetado. A função de ativação utilizada nas duas camadas intermediárias foi a tangente hiperbólica. Para a camada de saída aplicou-se uma função linear. O treinamento do modelo foi realizado em 2000 épocas a uma taxa de aprendizado de 0,15 e termo de momento de 0,9.

Para o modelo que considera apenas uma informação do preço, a_{t-1} , (aqui denominado de *RNA 1T*), uma *MLP* composta por quatro neurônios na camada de entrada, duas camadas intermediárias sendo a primeira com 12 e a segunda com 20 neurônios e um neurônio na camada de saída foi projetada. A função de ativação utilizada nas duas camadas intermediárias foi a tangente hiperbólica. Para a camada de saída aplicou-se uma função linear. O treinamento do modelo foi realizado em 2000 épocas a uma taxa de aprendizado de 0,1 e termo de momento de 0,9.

Discussões e comparações dos resultados obtidos nos testes dessa seção estão disponíveis na seção 6.2.3.

6.2.3 Discussão dos Resultados (Série de Preço da Arroba do Boi Gordo)

Os resultados da aplicação dos modelos *fuzzy* e neurais propostos nesse trabalho para a série de preços bem como o resultado da aplicação dos métodos considerados em Freiman (2006) estão sumarizados na Tabela 7.

A Tabela 7 mostra que a rede neural obtida por Freiman (2006) não se mostrou eficiente na previsão da série. A inserção de uma camada intermediária a mais na estrutura da *MLP* resultou na diminuição do erro em praticamente cinco vezes quando comparado a “*RNA 1T*” (1,4965% contra 6,2482%).

Para os testes considerados, pode-se notar que a presença de uma variável a mais para

Tabela 7: Tabela de erros da previsão da série de preços da arroba de boi.

	Mês / Ano (Valores)			Mês / Ano (Erros)%			Erro Total %
	03/abr	04/abr	05/abr	03/abr	04/abr	05/abr	
Cotação Real	20,27	20,43	19,63	-	-	-	
RNA	19,47	18,00	20,20	-3,9467	-11,8943	2,9037	6,2482
ARM 1T	19,84	20,06	20,22	-2,1214	-1,8111	3,0056	2,3127
ARM 2T	19,57	19,83	20,14	-3,4534	-2,9369	2,5981	2,9954
Fuzzy 1T	19,96	196,46	19,65	-1,4991	-3,8369	0,1066	1,8142
Fuzzy 2T	20,30	19,95	19,59	-0,1764	-2,3293	-0,208	0,9046
RNA 1T	20,16	20,11	20,10	-0,5359	-15,433	24,103	1,4965
RNA 2T	20,18	19,45	20,04	-0,448	-4,7873	2,1009	2,4454

treinamento não proporcionou uma melhoria no resultado obtido pelos modelos baseados em regras *fuzzy*. O contrário pode ser afirmado para os modelos neurais onde a rede “*RNA 1T*” obteve resultado melhor que a “*RNA 2T*”.

Dentre os modelos gerados via uso dos paradigmas da inteligência artificial, o modelo baseado em regras *fuzzy* “*Fuzzy 2T*” foi o que apresentou menor erro percentual geral (0,9046%). Isso confirma a hipótese de que a presença de variáveis exógenas podem auxiliar o processo de previsão resultando em modelos baseados em regras *fuzzy* capazes de obter bons resultados.

Com relação ao método de auto regressão múltipla considerado, pode-se verificar pela Tabela 7 que todos os modelos baseados em regras *fuzzy* propostos nesse trabalho para esta série, obtiveram erros menores.

6.3 Experimento 3 - Série de Demanda de Carga Elétrica

O trabalho de Carpinteiro et al. (2007) utilizou observações de demanda da carga elétrica de uma determinada região dos Estados Unidos para realizar a previsão de demanda dois anos a frente.

No total, oito experimentos foram considerados sendo que quatro deles levaram em consideração observações semanais e quatro observações mensais. Maiores detalhes sobre os experimentos podem ser encontrados na referência citada.

O objetivo desse experimento é comparar os resultados dos modelos baseados em regras gerados via *fuzzy* e *conjuntos aproximados* com um dos experimentos realizados no artigo citado.

Os dados da série analisada nesse estudo consistem em observações mensais que vão do intervalo 1985 a 1986 e 1988 a 1989. O objetivo é prever a demanda de carga dois anos a frente, ou seja, 1990 e 1991. As observações disponíveis para estudo encontravam-se normalizadas no intervalo de -0,5 a 0,5 e variavam de 1265,917 a 3974,333.

Os elementos do conjunto de dados para treinamento considerados é composto por três valores de entrada e um de saída. O primeiro valor corresponde ao valor da carga no mês t . O segundo e terceiro valores, denominados de integradores temporais, são referentes às informações do mês em questão e são expressas pelas expressões $\sin(2\pi month/12)/4$ e $\cos(2\pi month/12)/4$. O dado de saída é o valor da carga em $t + 1$. Ao todo, dispõe-se de quarenta e oito pares de informação para os dados de treinamento.

O mesmo conjunto de treinamento aplicado nos modelos neurais da referência foi usado para o treinamento do *ANFIS* e para a criação do sistema de informação usado na obtenção do modelo via *conjuntos aproximados*. Por fim, a fim de comparar o desempenho dos modelos baseados em inteligência artificial com os estatísticos, foi considerado nesse experimento a previsão da série via método de *Holt Winters* na série em estudo.

6.3.1 Modelo Neural Aplicado à Previsão da Série de Demanda de Carga Elétrica

Carpinteiro et al. (2007) construíram dois modelos neurais híbridos hierárquicos compostos por duas redes neurais *SOM* (*Self Organizing Map*) e uma *SLP* (*Single Layer Perceptron*) para prever dois anos a frente a série considerada nesse experimento. O primeiro modelo utiliza uma função gaussiana para as redes *SOM* e o segundo considera uma função discreta. A Figura 13 ilustra a estrutura destes modelos neurais. Além destes, uma *MLP* foi proposta para fins de comparação de resultados.

O gráfico da Figura 14 ilustra os resultados obtidos na referência em questão. A discussão desses resultados juntamente com a dos outros modelos considerados pode ser encontrada na seção 6.3.5.

6.3.2 Modelo Fuzzy Aplicado à Previsão da Série de Demanda de Carga Elétrica

Para utilização do *ANFIS* neste exemplo, aplicou-se o algoritmo *Fuzzy C-Means* nos dados de treinamento a fim de agrupá-los em 20 *clusters*. À estrutura inicial, conforme ilustrado pela na Figura 15, aplicou-se um processo de treinamento de 100 épocas a uma

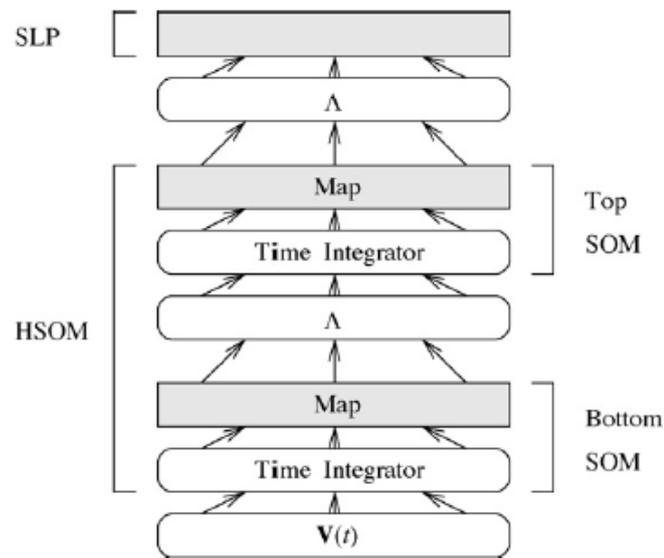


Figura 13: Estrutura dos modelos neurais hierárquicos empregados. Fonte: Carpinteiro et al. (2007)

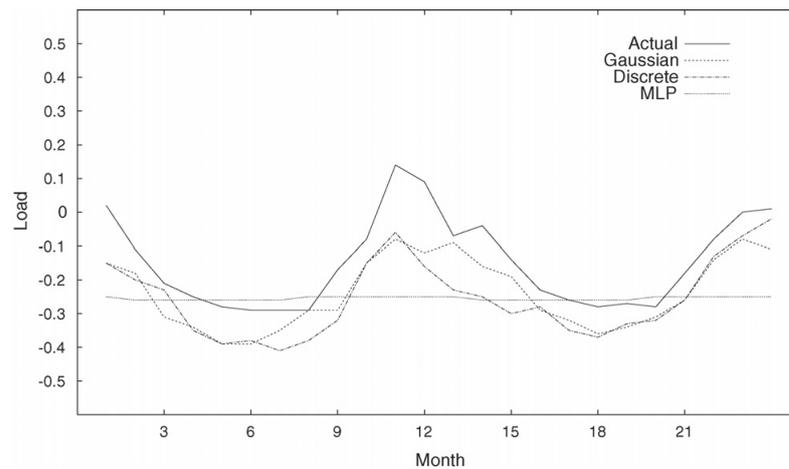


Figura 14: Gráfico da previsão da série de demanda de carga elétrica via redes neurais. Fonte: Carpinteiro et al. (2007)

taxa de aprendizado de 0,01. As funções de pertinência para os neurônios da primeira camada do *ANFIS* foram todas do tipo gaussiana. Obteve-se no final, um modelo composto por 20 regras *fuzzy*.

O gráfico da Figura 16 apresenta a adequação do modelo proposto, aos dados de treinamento. Pode-se verificar que o modelo *fuzzy* aplicado conseguiu bons resultados na modelagem da série temporal original resultando num erro médio quadrático de $5,6184 \cdot 10^{-5}$.

O gráfico da Figura 17 mostra os resultados da previsão. A análise desses resultados

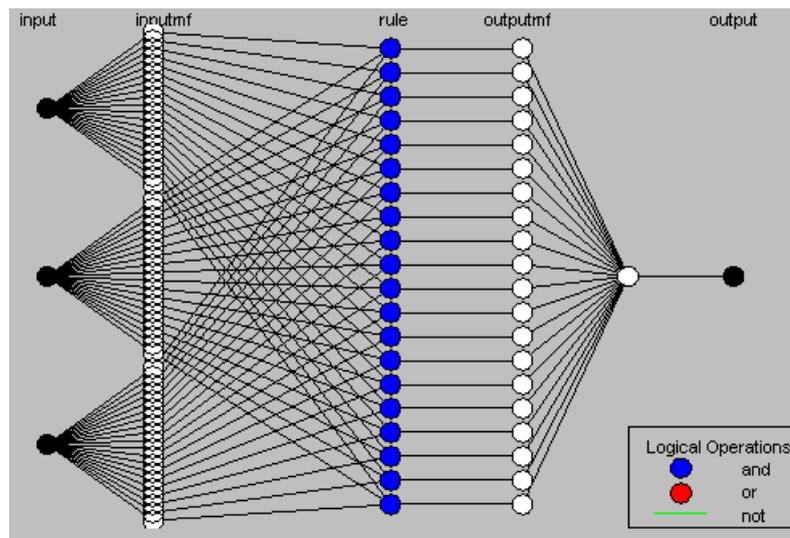


Figura 15: Estrutura do *ANFIS* aplicado no processo de previsão da série de demanda de carga elétrica.

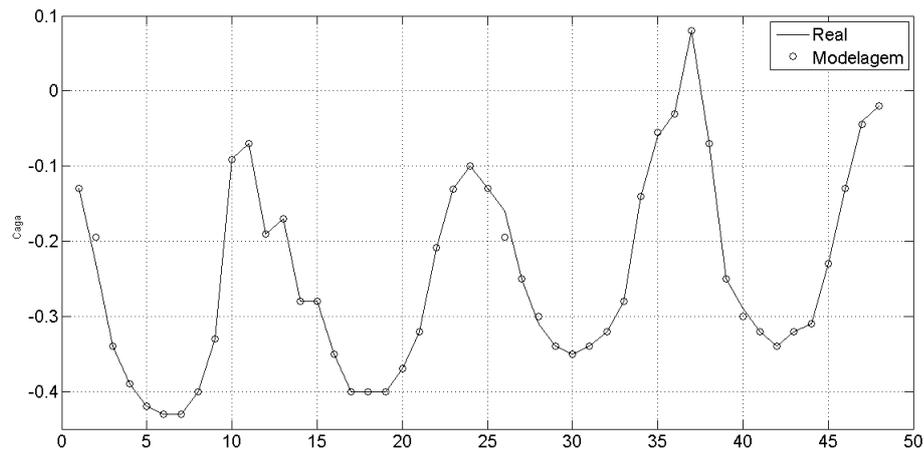


Figura 16: Gráfico da adequação do modelo *fuzzy* aos dados de treinamento da série de demanda de carga elétrica.

encontra-se na seção 6.3.5 onde uma comparação com os demais modelos aplicados é realizada.

6.3.3 Modelagem via Conjuntos Aproximados Aplicada à Previsão da Série de Demanda de Carga Elétrica

O *SI* deste exemplo é composto por três atributos condicionais e um de decisão conforme discutido na seção 6.3.

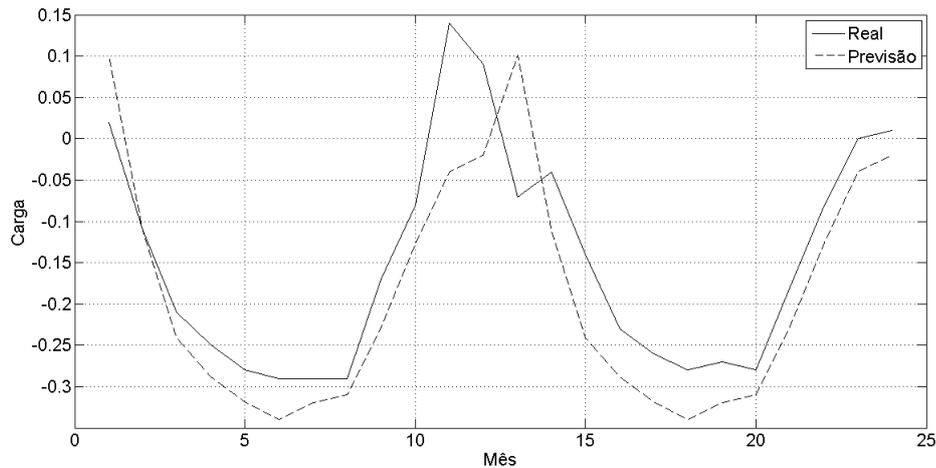


Figura 17: Gráfico da previsão da série de demanda de carga elétrica realizada pelo modelo baseado em regras *fuzzy*.

Os dados da série foram processados pelo software *Rosetta*. A opção de discretização empregada foi o *Equal Frequency Binning* com nível 4 de discretização. Após a discretização, a técnica de construção de redutos via *conjuntos aproximados* foi empregada e o conjunto de regras foi gerado.

A estrutura das regras dos modelo em questão fica então representada por (6.3) conforme mostrado na seção 4.4.2.

$$IF x_1 = [\alpha_j, \alpha_m] \text{ AND } x_2 = [\beta_j, \beta_m] \text{ AND } x_3 = [\gamma_j, \gamma_m] \text{ THEN } y = [\omega_j, \omega_m] \quad (6.3)$$

A Tabela 8 ilustra os parâmetros do modelo baseado em regras correspondente.

Resultados de simulações do modelo estão ilustrados na Figura 18. Os valores foram comparados com os valores da série original obtendo-se um erro quadrático médio de 0,0021, indicando uma modelagem adequada do processo.

Resultados de previsões da série estão indicados na Figura 19. A análise desses resultados encontra-se na seção 6.3.5.

Tabela 8: Regras do modelo aproximado para previsão de demanda de carga elétrica.

Regras	α_j	α_m	β_j	β_m	γ_j	γ_m	ω_j	ω_m
r_1	-0,1350	0,0800	0,0625	0,1710	0,1710	0,2500	-0,1300	0,0800
r_2	-0,1350	0,0800	0,1710	0,2500	0,0625	0,1710	-0,2300	-0,0700
r_3	-0,2850	-0,1350	0,1710	0,2500	-0,1710	0,0625	-0,3500	-0,2500
r_4	-0,3450	-0,2850	0,1710	0,2500	-0,1710	0,0625	-0,3900	-0,3900
r_5	-0,4300	-0,3450	0,0625	0,1710	-0,2500	-0,1710	-0,4200	-0,4000
r_6	-0,4300	-0,3450	-0,1710	0,0625	-0,2500	-0,1710	-0,4300	-0,3400
r_7	-0,4300	-0,3450	-0,2500	-0,1710	-0,1710	0,0625	-0,4000	-0,3200
r_8	-0,3450	-0,2850	-0,2500	-0,1710	0,0625	0,1710	-0,2100	-0,0900
r_9	-0,1350	0,0800	-0,1710	0,0625	0,1710	0,2500	-0,1900	-0,0200
r_{10}	-0,2850	-0,1350	0,0625	0,1710	0,1710	0,2500	-0,1700	-0,1700
r_{11}	-0,2850	-0,1350	0,1710	0,2500	0,0625	0,1710	-0,2800	-0,2800
r_{12}	-0,2850	-0,1350	-0,1710	0,0625	0,1710	0,2500	-0,1300	-0,0600
r_{13}	-0,4300	-0,3450	0,0625	0,1710	0,1710	0,2500	-0,1300	-0,1300
r_{14}	-0,3450	-0,2850	0,0625	0,1710	-0,2500	-0,1710	-0,3400	-0,3200
r_{15}	-0,3450	-0,2850	-0,1710	0,0625	-0,2500	-0,1710	-0,3500	-0,3200
r_{16}	-0,3450	-0,2850	-0,2500	-0,1710	-0,1710	0,0625	-0,3200	-0,2300
r_{17}	-0,2850	-0,1350	-0,2500	-0,1710	0,0625	0,1710	-0,1400	-0,1300
r_{18}	-0,1350	0,0800	0,1710	0,2500	-0,1710	0,0625	-0,2500	-0,2500

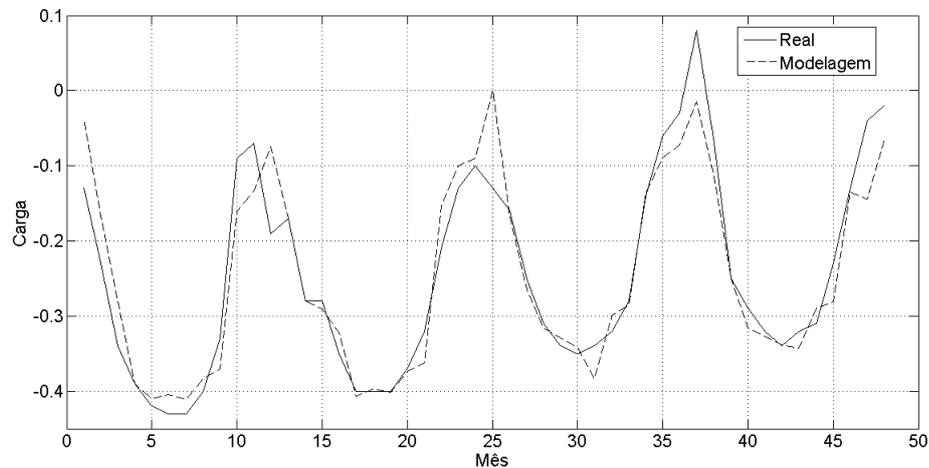


Figura 18: Gráfico da adequação do modelo aproximado aos dados de treinamento da série de demanda de carga elétrica.

6.3.4 Previsão da Série de Demanda de Carga Elétrica via Método de Holt Winters

Para aplicação do método de *Holt Winters*, considerou-se somente as observações referentes à carga do mês em questão desconsiderando-se os integradores temporais.

Resultados da previsão da série via o método em questão estão indicados na Figura 20.

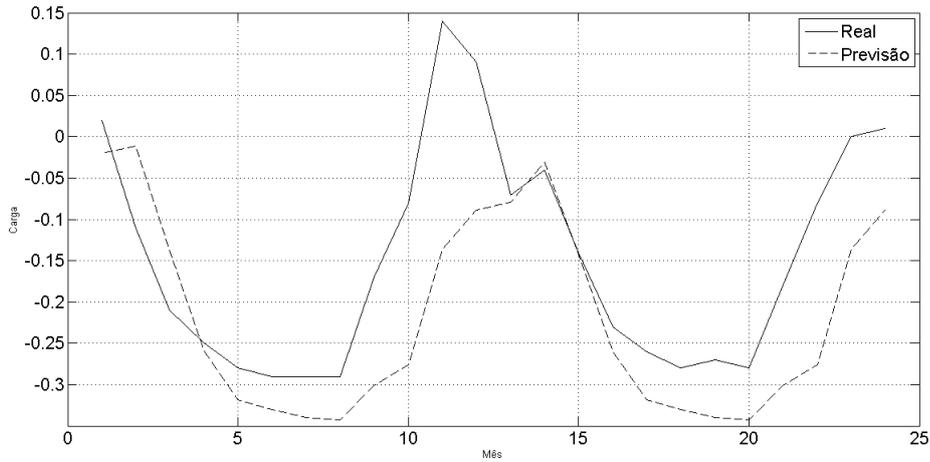


Figura 19: Gráfico da previsão da série de demanda de carga elétrica realizada pelo modelo aproximado.

Os parâmetros utilizados na modelagem foram α , δ e β em 0.2, 0.2, 0.2 respectivamente. A seção 6.3.5 discute os resultados.

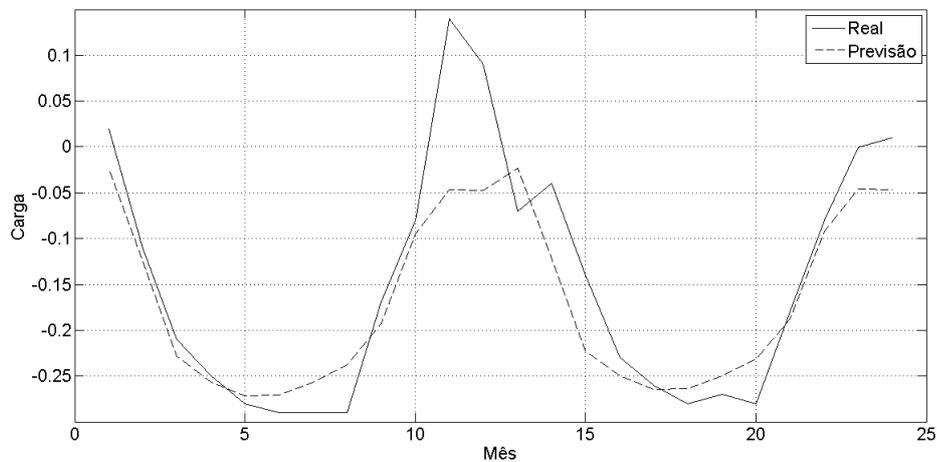


Figura 20: Gráfico da previsão da série de demanda de carga elétrica realizada pelo método de Holt Winters.

6.3.5 Discussão dos Resultados (Série Demanda de Carga Elétrica)

A Tabela 9 indica os erros da previsão dos modelos considerados nesse experimento.

A análise da Tabela 9 permite concluir que a modelagem baseada em regras via *fuzzy* e *conjuntos aproximados* considerados nesse trabalho apresentaram resultados compatíveis com as arquiteturas neurais híbridas propostas.

Tabela 9: Tabela de erros da previsão da série de demanda de carga elétrica.

Método	MAD	MSE	MAPE
MLP	0.123	0.028	13.276
Híbrido (discreta)	0.106	0.015	12.746
Híbrido (gaussiana)	0.089	0.011	10.615
ANFIS	0.060	0.0054	7.119
Rough Sets	0.085	0.012	9.854
Holt Winters	0.042	0.0036	4.711

O modelo gerado via *conjuntos aproximados* obteve, com um *MAPE* de 9,854, um erro sensivelmente menor que os modelos neurais considerados para comparação. Dentre os modelos obtidos via aplicação de técnicas da inteligência artificial, o modelo baseado em regras *fuzzy* apresentou menor erro em todos os indicadores de erros considerados.

O método estatístico de *Holt Winters*, com um *MAPE* de 4,711, obteve um erro praticamente duas vezes menor que os outros métodos considerados. A explicação para o melhor desempenho pode ser atribuída ao fato do método ter sido projetado para trabalhar de forma otimizada em séries cuja componente sazonal se faz presente de forma bastante acentuada, como é o caso da série considerada nesse exemplo.

6.4 Experimento 4 - Série de Atividade Pluviométrica

A cidade de Itajubá sofre frequentemente com enchentes devido aos altos índices pluviométricos no primeiro trimestre de cada ano.

O Instituto de Recursos Naturais (IRN) da Universidade Federal de Itajubá desenvolveu um sistema de monitoramento baseado em sensores que mede o nível de água do rio *Sapucai*. Uma vez a cada hora, um banco de dados é atualizado com a medida da altura do rio em relação ao nível do mar. Essas medições servem como parâmetros para alertas de enchentes. A Figura 21 esquematiza o funcionamento do sistema citado.

O banco de dados do sistema estava disponível para consulta via uma aplicação web que podia ser acessada através do endereço "<http://www.enchentes.unifei.edu.br/>".

De posse do citado banco de dados, uma série temporal considerando informações de janeiro de 1999 até junho de 2006 foi gerada. Cada observação da série temporal é resultante da média aritmética de todas as medições realizadas no mês em questão. O gráfico da Figura 26 ilustra a série desse estudo. A proposta de previsão consiste em estimar a altura do rio em relação ao nível do mar doze passos a frente, ou seja, 1 ano.

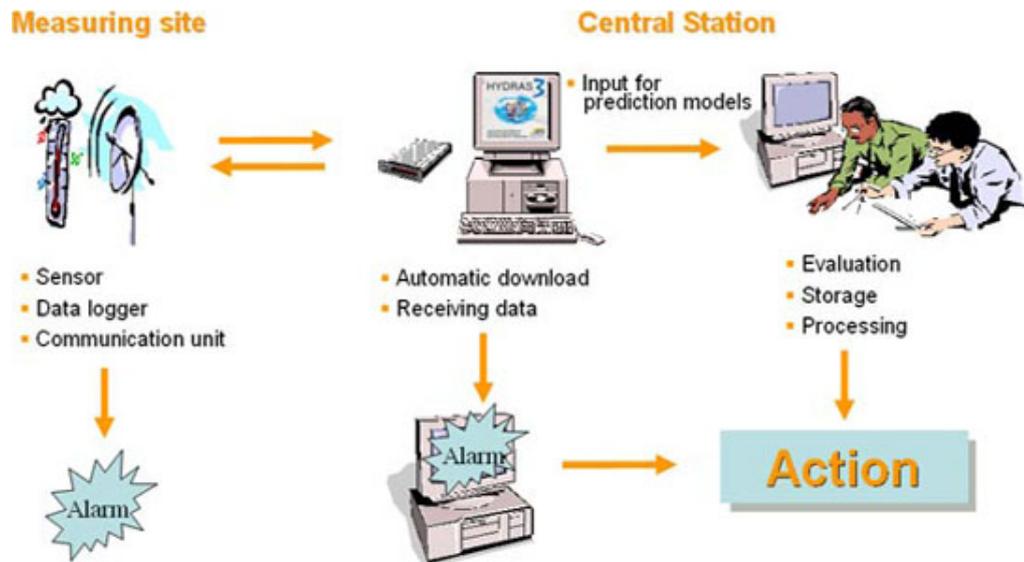


Figura 21: Esquemática do sistema de monitoramento de enchentes. Fonte: “<http://www.enchentes.unifei.edu.br/>”.

Para realização dos testes, dois conjuntos de treinamento foram construídos para determinação dos modelos *fuzzy* e neural. Os elementos do primeiro tomam quatro observações do passado para previsão um passo a frente. Os do segundo, tomam três observações do passado e uma informação sobre o mês em questão para previsão um passo à frente. O sistema de informação para o modelo via *conjuntos aproximados* tem quatro observações do passado como atributos condicionais para previsão um passo a frente, que corresponde ao atributo de decisão.

Em todos os casos o conjunto de treinamento é composto por oitenta e oito elementos.

As seções seguintes descrevem os modelos gerados bem como seus resultados. A seção 6.4.7 traz uma discussão dos resultados obtidos.

6.4.1 Modelo Neural Aplicado à Previsão da Série de Atividade Pluviométrica

Duas abordagens de previsão via redes neurais foram propostas. Cada uma delas, leva em consideração os conjuntos de treinamento descritos na seção (6.4).

6.4.1.1 Modelo Neural Aplicado à Previsão da Série de Atividade Pluviométrica com Informação Temporal

Os elementos do conjunto de treinamento desse exemplo são compostos por três observações da série e uma informação sobre o mês em questão. As observações que compõem

os dados de treinamento receberam a normalização da diferença conforme descrito na seção 5.2.

O conjunto de treinamento foi submetido a um modelo neural que conta com quatro neurônios na camada de entrada, duas camadas intermediárias com 18 e 30 neurônios respectivamente e um neurônio na camada de saída. A função de ativação utilizada nas duas camadas intermediárias foi a tangente hiperbólica. Para a camada de saída, uma função linear foi empregada. O treinamento do modelo foi realizado em 1200 épocas com taxa de aprendizado de 0,01 e termo de momento de 0,9.

O resultado da previsão está ilustrado pelo gráfico da Figura 22. A análise do erro obtido pode ser encontrada na seção 6.4.7

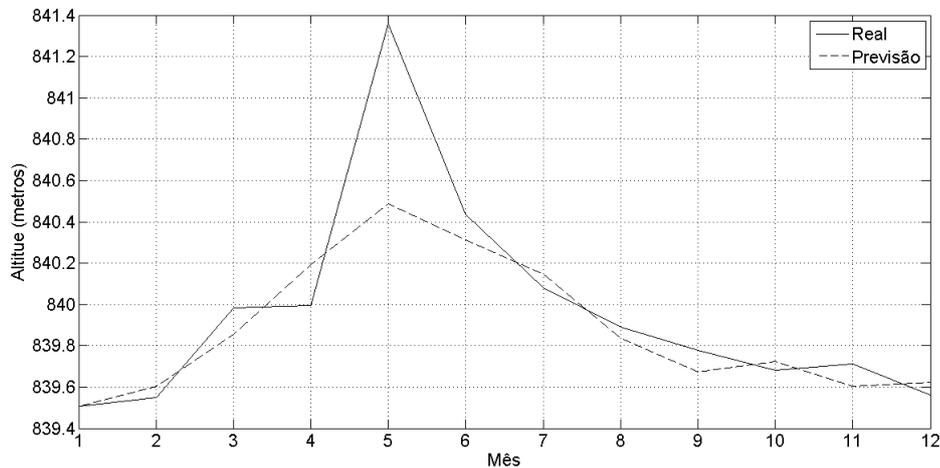


Figura 22: Gráfico da previsão da série de atividade pluviométrica realizada pelo modelo neural com informação temporal.

6.4.1.2 Modelo Neural Aplicado à Previsão da Série de Atividade Pluviométrica sem Informação Temporal

Para este exemplo, os elementos que compõem o conjunto de treinamento são compostos exclusivamente por observações históricas da série. As observações que compõem os dados de treinamento sofreram a normalização da diferença conforme descrito na seção 5.2.

O conjunto de treinamento foi submetido a um modelo neural que conta com quatro neurônios na camada de entrada, duas camadas intermediárias com 15 e 25 neurônios respectivamente e um neurônio na camada de saída. A função de ativação utilizada nas duas camadas intermediárias foi a tangente hiperbólica. Para a camada de saída, uma

função linear foi empregada. O treinamento do modelo foi realizado em 2000 épocas com taxa de aprendizado de 0,01, e termo de momento de 0,9. O resultado da previsão está ilustrado pelo gráfico da Figura 23.

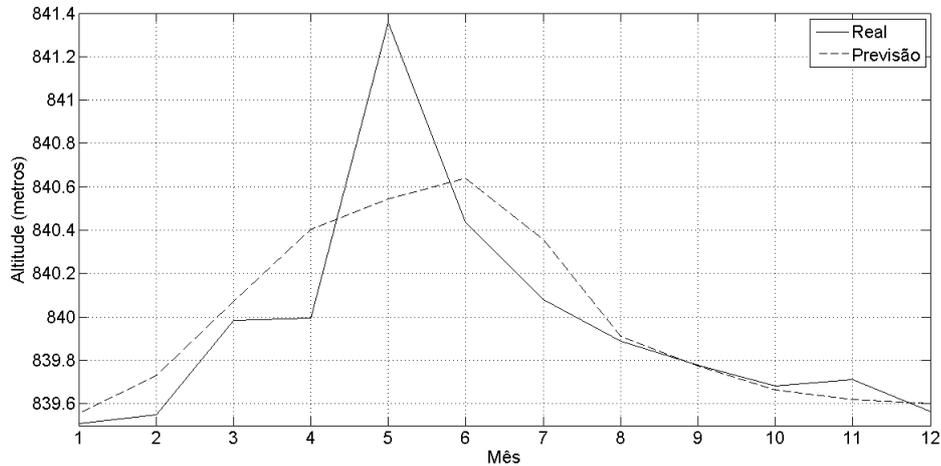


Figura 23: Gráfico da previsão da série de atividade pluviométrica realizada pelo modelo neural sem informação temporal

6.4.2 Modelo Fuzzy Aplicado à Previsão da Série de Atividade Pluviométrica

De forma análoga aos testes realizados com as redes neurais, a aplicação de modelos baseados em regras *fuzzy* considerou o uso de conjuntos de treinamento com e sem informações temporais.

6.4.3 Modelo Fuzzy Aplicado à Previsão da Série de Atividade Pluviométrica com Informação Temporal

Os elementos do conjunto de treinamento desse exemplo são compostos por três observações da série e uma informação sobre o mês em questão. As observações que compõem os dados de treinamento sofreram a normalização da diferença conforme descrito na seção 5.2.

O algoritmo *Fuzzy C-Means* foi aplicado nos dados de treinamento gerando um total de 60 *clusters*. À estrutura inicial, aplicou-se um processo de treinamento com 100 épocas a uma taxa de aprendizado de 0,01. As funções de pertinência para os neurônios da primeira camada do *ANFIS* foram todas do tipo gaussiana. Obteve-se no final um modelo composto por 60 regras *fuzzy*.

O resultado da previsão está ilustrado pelo gráfico da Figura 24. A seção 6.4.7 analisa o erro obtido.

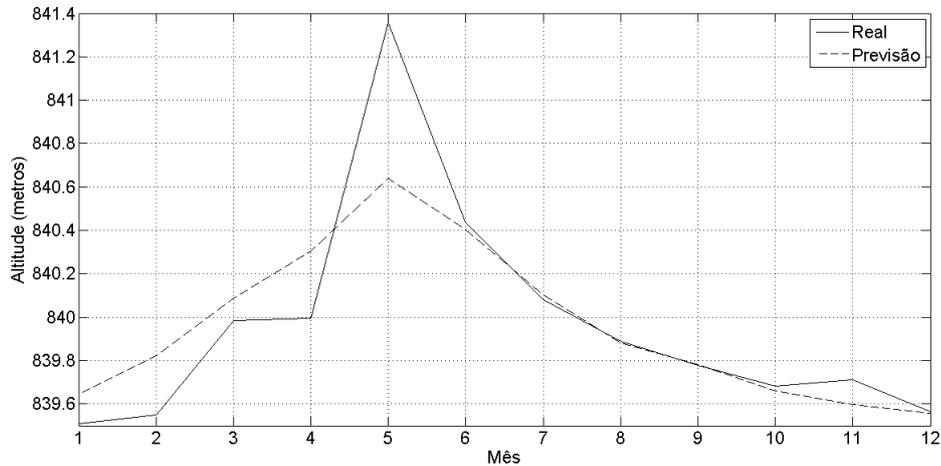


Figura 24: Gráfico da previsão da série de atividade pluviométrica realizada pelo modelo fuzzy com informação temporal

6.4.4 Modelo Fuzzy Aplicado à Previsão da Série de Atividade Pluviométrica sem Informação Temporal

Os elementos do conjunto de treinamento desse exemplo são compostos exclusivamente por observações históricas da série. Para esse teste, nenhuma normalização nos dados do conjunto de treinamento foi empregada.

O algoritmo *Fuzzy C-Means* foi aplicado nos dados de treinamento gerando um total de 60 *clusters*. À estrutura inicial, aplicou-se um processo de treinamento com 100 épocas a uma taxa de aprendizado de 0,01. As funções de pertinência para os neurônios da primeira camada do *ANFIS* foram todas do tipo gaussiana. Obteve-se no final um modelo composto por 60 regras *fuzzy*.

O resultado da previsão é ilustrado pelo gráfico da Figura 25.

6.4.5 Modelagem via Conjuntos Aproximados Aplicada à Previsão da Série de Atividade Pluviométrica

Para a determinação do modelo baseado em regras via *conjuntos aproximados*, um *SI* foi gerado com base nas medições normalizadas no intervalo de 0 a 1. O *SI* contém

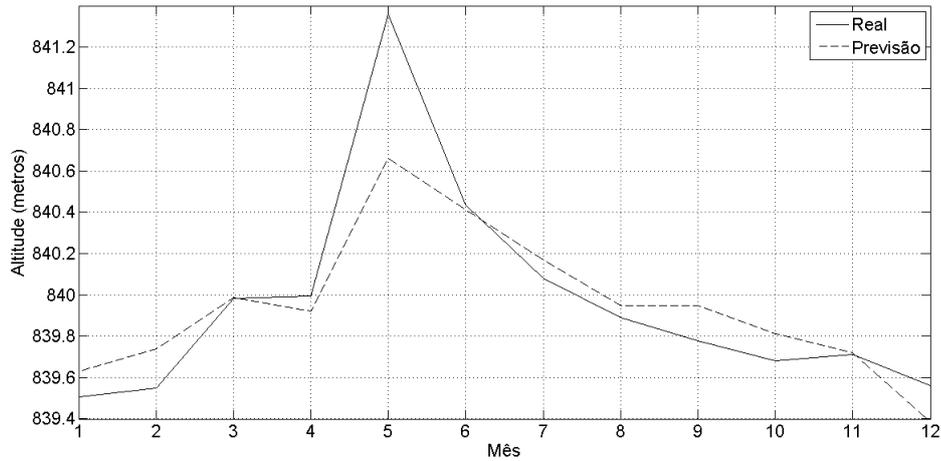


Figura 25: Gráfico da previsão da série de atividade pluviométrica realizada pelo modelo fuzzy sem informação temporal

quatro atributos condicionais ($a_t, a_{t-1}, a_{t-2}, a_{t-3}$) para previsão um passo a frente (a_{t+1}) e oitenta e oito observações.

Os dados da série foram processados no software *Rosetta*. O método de discretização empregado foi o *Equal Frequency Binning* com nível 5 de discretização. O modelo gerado é constituído por um conjunto de 57 regras. O resultado da simulação do modelo está ilustrado na Figura 26. Os valores foram comparados com os dados da série original normalizados. Obteve-se um erro médio quadrático de 0,0015. O resultado da previsão é ilustrado pelo gráfico da Figura (27).

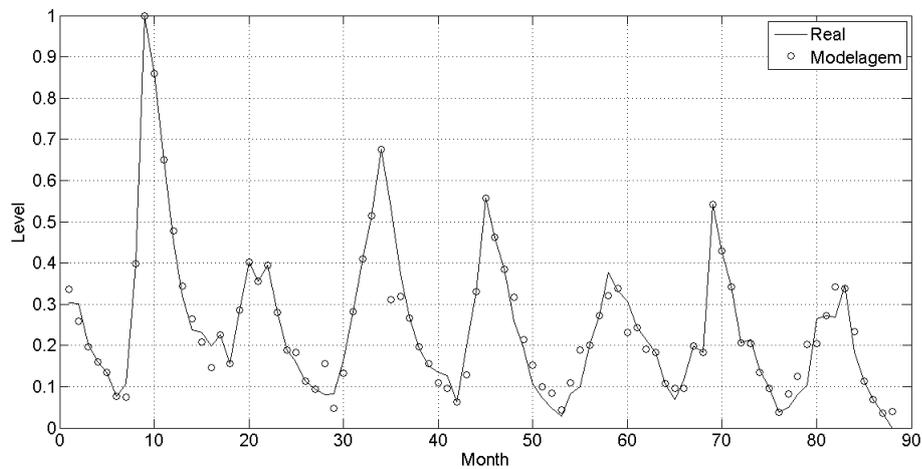


Figura 26: Gráfico da modelagem da série de atividade pluviométrica realizada pelo modelo baseado em regras gerado via conjuntos aproximados

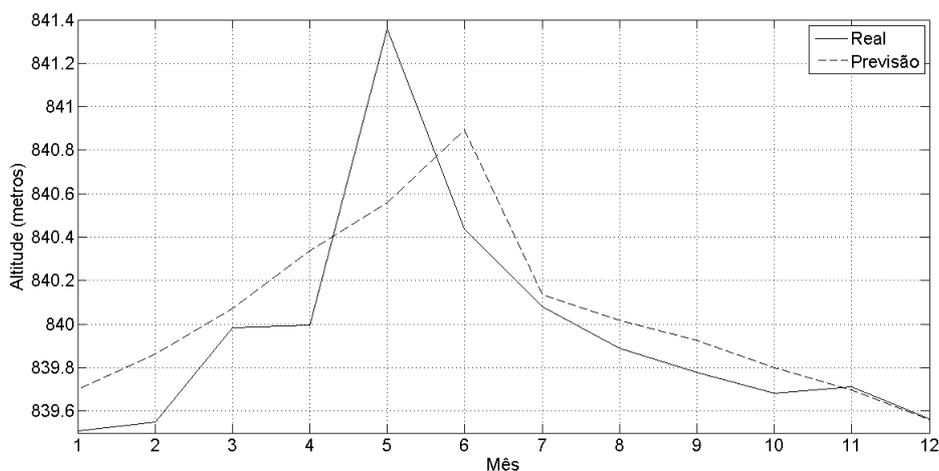


Figura 27: Gráfico da previsão da série de atividade pluviométrica realizada pelo modelo baseado em regras gerado via conjuntos aproximados

A seção 6.4.7 fornece os erros da previsão e compara o modelo desse teste com os outros modelos considerados no processo de previsão da série temporal desse experimento.

6.4.6 Previsão da Série de Atividade Pluviométrica via Método de Holt Winters

Resultados da previsão da série via o método de *Holt Winter* estão indicados na Figura 27. Os parâmetros utilizados na modelagem foram α , δ e β iguais a 0.9, 0.1, 0.1 respectivamente.

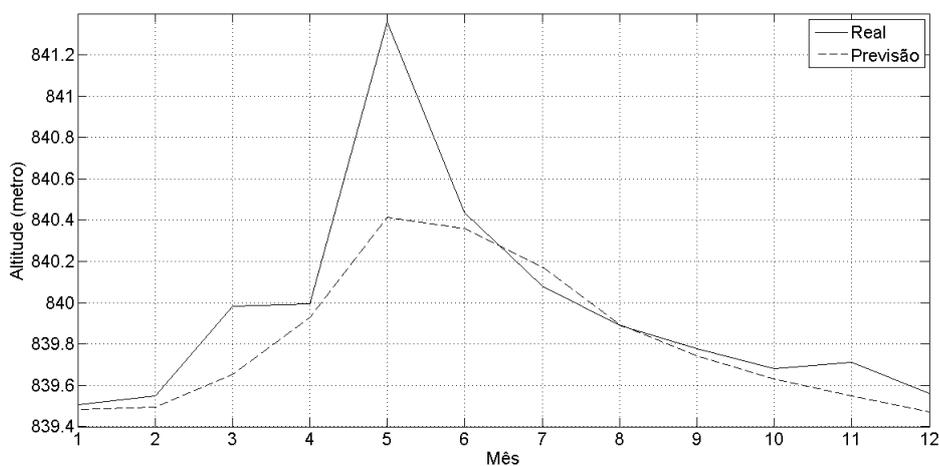


Figura 28: Gráfico da previsão da série de atividade pluviométrica realizada pelo método de Holt Winters.

6.4.7 Discussão dos Resultados (Série Atividade Pluviométrica)

A Tabela 10 trás os erros de previsão dos modelos considerados nesse experimento.

Tabela 10: Tabela de erros da previsão da série de atividade pluviométrica.

Método	Mean	MSE	MAPE	Desvio padrão
ANFIS	0.145	0.053	0.017	0.186
ANFIS Temporal	0.146	0.061	0.017	0.208
MLP	0.183	0.083	0.022	0.234
MLP Temporal	0.151	0.072	0.018	0.232
Rough Sets	0.220	0.097	0.026	0.229
Holt Winters	0.161	0.089	0.019	0.261

Dentre os modelos baseados em técnicas de inteligência artificial empregados, aqueles baseados em regras *fuzzy* foram os que obtiveram uma faixa de erro menor. A presença de uma variável com informação do mês em questão não resultou em ganhos de desempenho visto que o erro médio (*MAD*) para os dois modelos, manteve-se na casa dos 14 centímetros.

Em segundo lugar, tem-se as redes neurais artificiais que, diferentemente dos modelos *fuzzy*, apresentaram melhores resultados quando a presença da variável temporal no conjunto de treinamento estava presente.

O modelo baseado nas regras obtidas via aplicação da técnica de *conjuntos aproximados* foi aquele que apresentou maior erro (*MAPE* igual a 0,026). Contudo, um erro médio de 22 centímetros ainda pode ser considerado um resultado de previsão bastante razoável.

Quando comparado ao método de *Holt Winters*, os modelos *fuzzy* gerados obtiveram erros menores. As redes neurais apresentaram resultados semelhantes e o modelo via *conjuntos aproximados* resultados sensivelmente maiores.

6.5 Experimento 5

Essa seção tem como objetivo apresentar os experimentos realizados em uma série temporal financeira. A série em questão, é referente ao fundo de ações do Banco do Brasil e é representada pelo gráfico da Figura 29.

Séries temporais financeiras tem como característica, a presença de uma aleatoriedade grande nas observações que as compõem. Essa aleatoriedade confere a esse tipo de série um comportamento difícil de ser modelado e previsto.

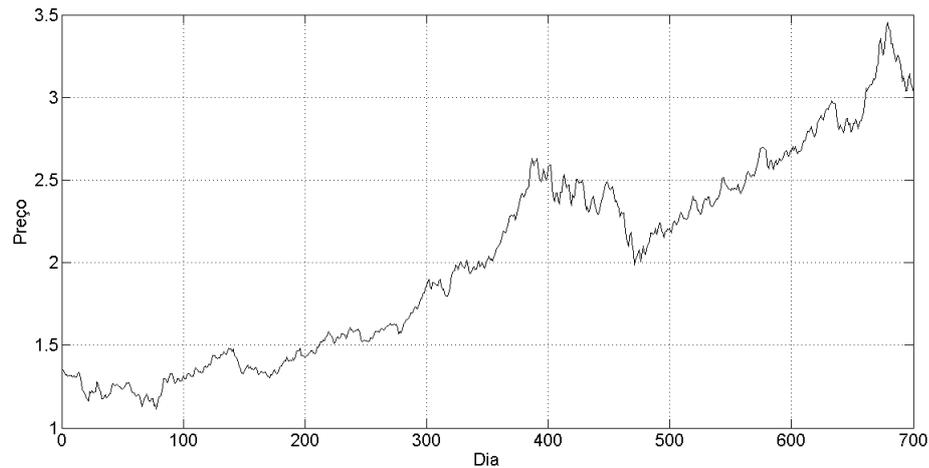


Figura 29: Gráfico da série de fundo de ações do Banco do Brasil.

A série desse estudo pode ser caracterizada como uma série mal comportada, e sendo assim, a obtenção de modelos baseados em regras capazes de atuar eficientemente no processo de previsão é o desafio desse experimento.

O conjunto de regras que compõem os modelos empregados no processo de previsão da série desse estudo são todos do tipo *fuzzy*. A fim de testar a eficiência dessas regras, redes neurais *MLP* foram projetadas.

Para os testes desse experimento, duas normalizações nos dados que compõem os elementos do conjunto de treinamento foram consideradas. A primeira é a normalização da diferença, dada pela aplicação da equação (6.2) e amplamente utilizada nos demais experimentos desse trabalho. A segunda normalização empregada foi a de retornos, frequentemente utilizada em processos de previsão que envolvem séries financeiras. Ambas as técnicas são descritas na seção 5.2.

No total, dispõe-se de uma série composta por 700 observações tomadas diariamente. Essas observações dão origem aos elementos dos conjuntos de treinamento que, por sua vez, são compostos por quatro observações do passado para previsão um passo à frente. Deseja-se realizar uma previsão para duas semanas, ou seja, quatorze passos a frente.

6.5.1 Modelos Fuzzy Aplicados na Previsão da Série de Fundo de Ações do Banco do Brasil

Para avaliação da eficiência dos modelos *fuzzy* em processos de previsão que envolvam séries mal comportadas, como a série desse estudo, dois conjuntos para treinamento,

conforme descrito na seção anterior, foram considerados.

Para o teste cujo conjunto de treinamento foi obtido via a normalização da diferença das observações da série, aplicou-se o algoritmo *Fuzzy C-Means* para agrupamento dos elementos do conjunto em 43 *clusters* distintos. O *ANFIS* utilizado passou por um processo de treinamento de 250 épocas a uma taxa de aprendizado de 0,01. As funções de pertinência para os neurônios da primeira camada do *ANFIS* foram todas do tipo gaussiana. Obteve-se no final, um modelo composto por 43 regras *fuzzy*.

O gráfico da Figura 30 ilustra o resultado da previsão.

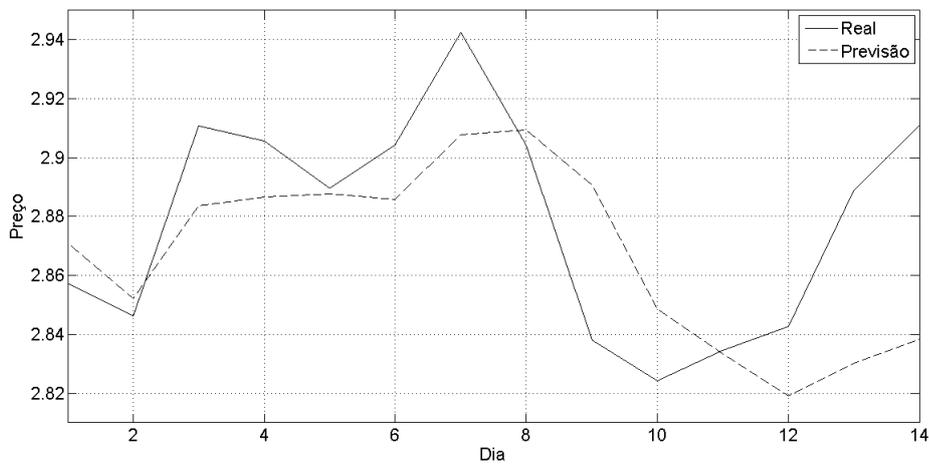


Figura 30: Gráfico da previsão da série de ações do Banco do Brasil (Normalização da diferença).

O teste que levou em consideração o conjunto de treinamento composto pelos retornos dos preços da série, empregou o algoritmo *Fuzzy C-Means* para agrupamento dos elementos do conjunto em 60 *clusters* distintos. O *ANFIS* utilizado passou por um processo de treinamento de 250 épocas a uma taxa de aprendizado de 0,01. As funções de pertinência para os neurônios da primeira camada do *ANFIS* foram todas do tipo gaussiana. Obteve-se no final, um modelo composto por 60 regras *fuzzy*.

Os resultados estão apresentados no gráfico da Figura 31.

Para fins de comparação de resultados, todos os valores previstos via modelos *fuzzy* foram desnormalizados para preços reais. A seção 6.5.3 traz os erros dos testes juntamente com os erros dos experimentos com modelos neurais de previsão.

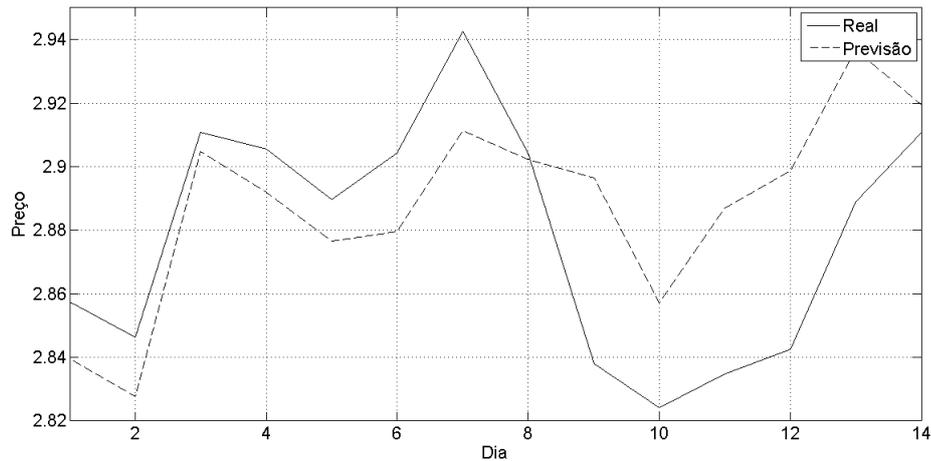


Figura 31: Gráfico da previsão da série de ações do Banco do Brasil (Retornos).

6.5.2 Modelos Neurais Aplicados na Previsão da Série de Fundos de Ações do Banco do Brasil

De forma análoga aos testes realizados com modelos *fuzzy*, duas redes neurais *MLP* foram propostas para esse experimento. Cada uma delas recebe para treinamento os conjuntos definidos na seção ??.

Para o teste cujo conjunto de treinamento foi obtido via a normalização da diferença das observações da série, foi projetada uma rede neural artificial com quatro neurônios na camada de entrada, duas camadas intermediárias com 15 e 20 neurônios respectivamente, e um neurônio na camada de saída. A função de ativação utilizada nas duas camadas intermediárias foi a tangente hiperbólica. Para a camada de saída, uma função linear foi empregada. O treinamento do modelo foi realizado em 1200 épocas com taxa de aprendizado de 0,02 e termo de momento de 0,9.

O gráfico da Figura 32 ilustra o resultado da previsão.

O teste que levou em consideração o conjunto de treinamento composto pelos retornos dos preços da série considerou uma *MLP* com quatro neurônios na camada de entrada e duas camadas intermediárias com 9 neurônios cada uma delas. A camada de saída contava com um neurônio. A função de ativação utilizada nas duas camadas intermediárias foi a tangente hiperbólica. Para a camada de saída, uma função linear foi empregada. O treinamento do modelo foi realizado em 1600 épocas com taxa de aprendizado de 0,02 e termo de momento de 0,9.

Os resultados estão apresentados no gráfico da Figura 33.

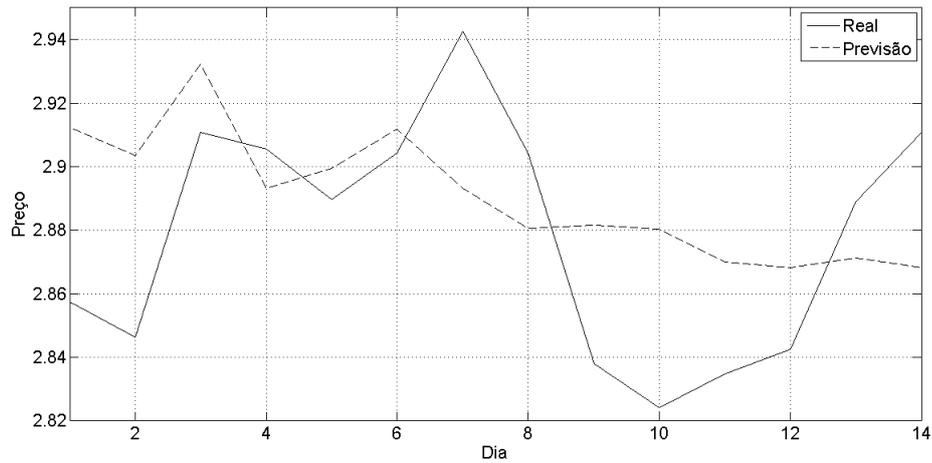


Figura 32: Gráfico da previsão da série de ações do Banco do Brasil via Modelo Neural (Normalização da diferença).

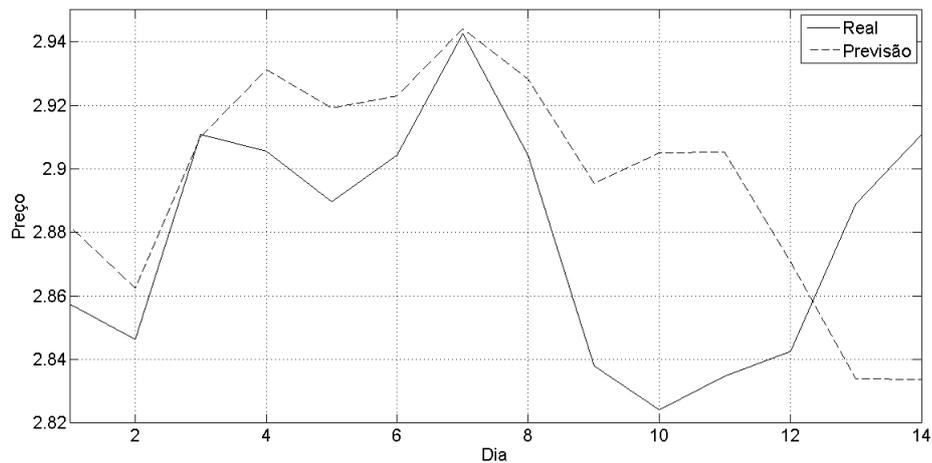


Figura 33: Gráfico da previsão da série de ações do Banco do Brasil via Modelos Neurais (Retornos).

A seção 6.5.3 traz os erros desse experimento calculados a partir dos resultados desnormalizados juntamente com os erros dos outros modelos considerados nesse experimento.

6.5.3 Discussão dos Resultados (Série Financeira)

A Tabela 11 trás os resultados dos testes realizados na série desse experimento.

Pode-se verificar que o modelo *fuzzy* obtido a partir da série normalizada via método das diferenças obteve o melhor resultado gerando um erro médio de 0,0149 centavos. Os outros indicadores de erro também mostram superioridade, exceto com relação ao desvio

Tabela 11: Tabela de erros da previsão da série financeira.

Método	Mean	MSE	MAPE	Desvio padrão
ANFIS (Diferença)	0.0149	0.00065	0.3882	0.022
ANFIS (Retorno)	0.0160	0.00065	0.4174	0.019
MLP (Diferença)	0.0200	0.00080	0.049	0.021
MLP (Retorno)	0.0212	0.00116	0.553	0.027

padrão, onde o outro modelo *fuzzy* considerado obteve resultado melhor. A explicação para o desvio padrão relativamente alto pode ser atribuída a elevada faixa de erro observada nas previsões das duas últimas observações. Se o experimento tivesse considerado somente doze dias de previsão, provavelmente o erro geral seria ainda menor e o desvio padrão o menor dentre os modelos considerados.

Considerando os meios de normalização de dados empregados, pode-se verificar que ambos modelos *fuzzy* conseguiram bons resultados. Contudo, o modelo que usou os dados normalizados pelo método da diferença levou vantagem pois é composto por 43, contra 60 regras do outro modelo *fuzzy*.

Os modelos neurais não apresentaram resultados ruins em termos de erros mas como pode ser verificado nos gráficos de previsão, o resultado da previsão dos modelos, principalmente naquele que considerou o conjunto de treinamento normalizado pelo método da diferença, resultou numa curva que se aproxima da média geral das observações.

Por fim, pode-se concluir comparando os erros e os gráficos dos processos de previsão desse experimento, que ambos modelos *fuzzy* gerados tiveram desempenho melhor que ambas as redes neurais projetadas e que, independente do método de normalização aplicado nas observações da série, o resultado final da previsão dos modelos não varia de forma considerável.

7 Conclusão

7.1 Considerações finais

Com base nas análises empreendidas nas cinco séries temporais escolhidas para esse trabalho, as questões propostas no Capítulo 1 podem agora ser respondidas.

A primeira questão cogitava se para uma determinada série temporal existia um conjunto de regras que, por meio de aplicação de uma metodologia de previsão, fosse capaz de modelar o comportamento da série e inferir resultados futuros de suas observações.

A resposta para essa pergunta é “*SIM para todas as séries consideradas nesse estudo*”. Como visto no Capítulo 6, foram encontrados modelos *fuzzy* para todas as séries estudadas. Obteve-se ainda, modelos baseados em regras gerados via técnicas de *conjuntos aproximados* com desempenhos satisfatórios em dois dos experimentos.

Uma vez respondida a primeira questão, a segunda questão, que indagava a eficiência de tais modelos, pode ser realizada. Para responder a essa pergunta, considerou-se o estudo e aplicação de outras técnicas amplamente empregadas em processos de previsão. Para fins de comparação, foram considerados modelos estáticos de previsão, objetos de estudo de diversas referências desse trabalho e a rede neural artificial *MLP*.

Como demonstrado, dentre as previsões que se basearam em técnicas apoiadas em paradigmas da inteligência artificial, aquelas que foram realizadas via modelos *fuzzy* obtiveram sempre os melhores resultados dentre os demais modelos considerados. Considerando os métodos estatísticos, os modelos *fuzzy* só ficaram atrás no experimento que analisou a série de demanda de carga elétrica. Portanto, a resposta da segunda pergunta para modelos baseados em regras fuzzy é “*Os modelos baseados em regras obtidos mostraram-se bastante eficientes para todas as séries consideradas neste trabalho*”.

Considerando os modelos obtidos via técnicas de *conjuntos aproximados*, pode-se concluir que a metodologia de previsão proposta nesse trabalho, baseada na pesquisa de Pinheiro et al. (2010a), apresenta resultados razoáveis a custo de um menor esforço com-

putacional pois não são necessárias etapas de treinamento e também procedimentos de fuzificação e defuzificação durante a aplicação dos modelos. Os experimentos que consideraram o uso dessa abordagem resultaram em erros bem similares para a série de demanda de carga elétrica e em erros maiores para a série de atividade pluviométrica, porém, ainda compatíveis em termos de tolerâncias aceitáveis na prática. Sendo assim, a resposta da segunda pergunta para esses modelos é “*Compatíveis com os demais modelos empregados*”.

Além da análise do desempenho, é válido ressaltar a simplicidade dos modelos baseados em regras quando comparados aos demais modelos, principalmente os neurais. É relevante um modelo composto por apenas 18 regras, como é o caso do *modelo aproximado* para previsão da demanda de carga elétrica, apresentar desempenho similar a consagradas metodologias de previsão. Além disso, os modelos baseados em regras ainda tem a vantagem de disponibilizar o conhecimento necessário para a realização de seus processos de forma clara por meio das regras *IF...THEN* que os compõem. Essas regras podem ser facilmente implementadas em softwares de propósito geral para as áreas afins. Já nos modelos neurais, o conhecimento está abstraído por meio de pesos cuja interpretação é, na grande maioria das vezes, mais complicada.

7.2 Contribuições e Sugestões para trabalhos futuros

A maior contribuição desse trabalho foi apresentar meios consistentes para que modelos baseados em regras fossem criados e aplicados em processos de previsão a partir de ferramentas da área da inteligência artificial.

Em especial, foi apresentada uma proposta de previsão baseada em *conjuntos aproximados*, ferramenta raramente empregada em processos de previsão de séries temporais, capaz de gerar *modelos aproximados* que podem atuar em processos de previsão. Para facilitar a determinação das regras de modelagem, foi desenvolvido um programa que compila os parâmetros de saída do software *Rosetta* em uma formato adequado a montagem das regras. O código fonte deste utilitário e do programa de computação dos modelos aproximados, estão presentes nos Apêndices A e B da dissertação.

Este trabalho gerou um artigo aceito para publicação em março de 2011 (FAUSTINO et al., 2011).

Como sugestão para trabalhos futuros propõe-se:

- Considerar aplicação de técnicas de algoritmos genéticos para determinação de pa-

râmetros dos sistemas de previsões propostos nesse trabalho como por exemplo, a quantidade ótima de *cluster* que devem ser gerados para aplicação do *ANFIS*.

- Testar outros algoritmos de discretização de dados a fim de avaliar o impacto que estes algoritmos trazem no desempenho final dos modelos aproximados.
- Estudar a possibilidade de se criar sistemas híbridos baseados em conjuntos aproximados e redes neurais artificiais ou conjuntos aproximados e *ANFIS* onde a teoria de conjuntos aproximados poderia ser aplicado para se fazer um estudo de correlação nos dados com o objetivo de diminuir a dimensionalidade do conjunto de treinamento da *RNA* e do *ANFIS* principalmente em situações onde variáveis exógenas fizerem-se presentes.

APÊNDICE A – Programa para tratamento das regras geradas pelo Rosetta

A fim de facilitar o uso do método de previsão baseado em conjuntos aproximados apresentado neste trabalho, tornando-o o mais automatizado possível, foi desenvolvido um programa cuja finalidade é processar as regras geradas pelo *Rosetta* deixando as no formato representado em (4.12).

A saída do programa consiste em um arquivo de texto cujos dados define uma matriz onde cada linha representa uma regra do modelo aproximado. Considerando n o número total de dados em cada linha da matriz, os $n-2$ primeiros termos representam os antecedentes das regras. Cada par desses n primeiros especifica o intervalo em que uma variável está inserida. Os dois últimos dados de cada linha definem o intervalo o qual o atributo de decisão está inserido.

Este arquivo de saída possui um formato compatível com os arquivos reconhecidos pelo MatLab, sendo assim, no processo de simulação do modelo aproximado carrega-se o arquivo e aplica-se o processo discutido na seção 4.4.2.

O código desse programa encontra nas listagens apresentadas a seguir.

Listagem A.1: Classe principal do programa para tratamento das regras

```
package rulesprocessor;
```

```
public class Main {
```

```
5     public static void main(String [] args) {
```

```
        String arquivo;
```

```
        //Carrega arquivo gerado pelo rosetta
```

```
10     arquivo = "D:\\UNIFEI\\Mestrado\\Dissertacao2\\Experimentos\\carga  
        \\rough\\RulesProcessor\\rulesFiles2";
```

```

        //Deve-se definir 2 vetores com os maiores e menores valores de
        //variavel
        double [] min = new double [3];
        min [0] = -0.43;
15      min [1] = -0.25;
        min [2] = -0.25;

        double [] max = new double [3];
        max [0] = 0.08;
20      max [1] = 0.25;
        max [2] = 0.25;

        //Este é a classe responsavel pelo processamento
        RulesConverter conversor = new RulesConverter(18, 3, min, max,
        arquivo);
25      conversor.generateRules();
        conversor.showRules();
        conversor.generateFile();

    }
30 }

```

Listagem A.2: Classe principal do programa para tratamento das regras

```

package rulesprocessor;

import java.io.File;
import java.io.FileNotFoundException;
5  import java.util.Formatter;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

10 public class RulesConverter {

    //define-se o numero de antecedentes das regras
    private int numberOfTerms = 2;
    //vetores de valores minimos e maximo de cada variável
15  private double [] min;
    private double [] max;
    //quatro vetores auxiliares para o processamento
    private double regraTermo1 [] = new double [32];
    private double regraTermo2 [] = new double [32];

```

```

20     private double regrasConsequente1 [];
    private double regrasConsequente2 [];
    //numero de regras
    private int numberOfRules = 16;
    private String arquivo;
25
    //construtor da classe, seta o valor das variaveis definidas
    public RulesConverter(int nor, int nof, double min[], double max[],
        String arquivo) {
        numberOfRules = nor;
        numberOfTerms = nof;
30     this.min = min;
        this.max = max;
        regraTermo1 = new double[numberOfTerms * numberOfRules];
        regraTermo2 = new double[numberOfTerms * numberOfRules];
        regrasConsequente1 = new double[numberOfRules];
35     regrasConsequente2 = new double[numberOfRules];
        this.arquivo = arquivo;
    }

    //método para gerar as regras
40     public void generateRules() {

        try {

            int cont = 0;
45     int cont1 = 0;

            //lê o arquivo do rosetta
            Scanner s = new Scanner(new File(arquivo));
            int contgeral= 1;
50

            //inicia leitura
            while (s.hasNext()) {

                //lê a linha atual
55     String linha = s.nextLine();
                //substitue caracteres não compatíveis para o
                //processamento
                linha = linha.replace("[", "u");
                linha = linha.replace(">", "j");

60     //obtem cada valor da linha atual

```

```

String [] linhasep = linha.split("u");

        //obtem os consequentes das regras
String last = linhasep[linhasep.length - 1];
65 String [] lasts = last.split("="j");
linhasep[linhasep.length - 1] = lasts[0];

        //percorre todos os valores da linha
70 for (int i = 0; i < linhasep.length; i++) {
        //verifica o formato correto
        if (!linhasep[i].contains("x1")) {
                //verifica se é inicio de intervalo
                if (linhasep[i].split(",")[0].equals("*")) {
                        regraTermo1[cont] = min[i - 1];
75 } else {
                        regraTermo1[cont] = Double.parseDouble(linhasep
                                [i].split(",")[0]);
                }
                //preenche vetor que representa os maiores valores
                para os intervalos
                if (linhasep[i].split(",")[1].replace(")", "%").
                        split("%")[0].trim().equals("*")) {
80 regraTermo2[cont] = max[i - 1];
                } else {
                        regraTermo2[cont] = Double.parseDouble(linhasep
                                [i].split(",")[1].replace(")", "%").split("%"
                                ) [0].trim());
                }

                //imprime os termos na tela pra
                verificar se esta certo
85 System.out.println("TERMOS:␣" + regraTermo1[cont] +
                        "␣-␣" + regraTermo2[cont]);
                cont++;
        }

90 }

        //processa os consequentes
String consequente [] = lasts[1].split("OR");
double menor = 10000;
95 double maior = -10000;

```

```

        //varre os consequentes disponiveis
    for (int j = 0; j < consequente.length; j++) {
        //substitue caracteres incompativeis
100         String str = consequente[j].replace("(", "%").
            split("%")[1].trim();
        double n = Double.parseDouble(str.substring(0, str.
            length() - 1));
        //obtem menores e maiores valores dos
            consequentes
        if (n < menor) {
            menor = n;
105     }
        if (n > maior) {
            maior = n;
        }

110     //preenche vetores de maior e menor valores para os
        intervalos dos consequentes
        regrasConsequente1[cont1] = menor;
        regrasConsequente2[cont1] = maior;

    }

115     //imprime na tela o resultado do processamento de
        regra da interaçao atual
    System.out.println(cont1 + "_-" + regrasConsequente1[cont1]
        + "_-" + regrasConsequente2[cont1]);
    cont1++;

    }

120     s.close();

    } catch (FileNotFoundException ex) {
125         Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null,
            ex);
    }
}

//gera arquivo com as regras ja processadas
130 public void generateFile() {

    java.util.Formatter output = null;

```

```

try {
    output = new Formatter("entrada.txt");
135     for (int i = 0; i < regrasConsequente1.length; i++) {
        for (int k = 0; k < numberOfTerms; k++) {
            output.format("%s□%s□", regraTermo1[i * numberOfTerms +
                k], regraTermo2[i * numberOfTerms + k]);
        }
        output.format("%s□%s", regrasConsequente1[i],
            regrasConsequente2[i]);
140     output.format("\n");
    }

    output.close();
} catch (Exception ex) {
145     output.close();
    ex.printStackTrace();
}

}

150 //exibe o arquivo com as regras processadas na tela
public void showRules() {

    for (int i = 0; i < regrasConsequente1.length; i++) {
155     for (int k = 0; k < numberOfTerms; k++) {
        System.out.print("x" + (k + 1) + "□>□" + regraTermo1[i *
            numberOfTerms + k] + "□AND□□x" + (k + 1) + "□<□" +
            regraTermo2[i * numberOfTerms + k] + "□AND□");
        //output.format("%s %s ", regraTermo1[i * numberOfTerms + k
            ], regraTermo2[i * numberOfTerms + k]);
    }
    System.out.println("□entao□y□esta□entre□" + regrasConsequente1[
        i] + "□e□" + regrasConsequente2[i] );
160
    }

}

165 }

```

APÊNDICE B – Programa de processamento de regras para simulação dos modelos aproximados

A simulação dos modelos aproximados é realizada através de um programa desenvolvido em MatLab. Este programa recebe como entrada o arquivo gerado pelo processamento realizado pelo programa descrito na Apêndice A.

A saída desse programa consiste nos valores de simulação dos dados utilizados para geração dos modelos e de valores da previsão n passos a frente. Gráficos da simulação e da previsão também são gerados.

A seguir, tem-se a listagem do programa de simulação dos modelos aproximados.

Listagem B.1: Listagem do programa de simulação de modelos aproximados

```
% Teste regras do modelo do sistema nao-linear.
```

```
clear all;
```

```
5 %carregar regras
```

```
load D:\UNIFEI\Mestrado\Dissertacao2\Experimentos\carga\rough\regras\  
    dadosCarga\regras_d4;
```

```
load D:\UNIFEI\Mestrado\Dissertacao2\Experimentos\carga\cargaTreinamento;
```

```
load D:\UNIFEI\Mestrado\Dissertacao2\Experimentos\carga\cargaTeste;
```

```
load D:\UNIFEI\Mestrado\Dissertacao2\Experimentos\carga\rough\resultados\ye
```

```
    ;
```

```
10
```

```
Np=48; % Parâmetros da serie.
```

```
vn=1:Np;
```

```
minCarga = -0.6;
```

```
15
```

```
maxCarga = 0.08;
```

```

x = cargaTreinamento(1:48,1);
x1 = cargaTreinamento(1:48,2);
x2 = cargaTreinamento(1:48,3);
20 y = cargaTreinamento(1:48,4);

plot(1:48,y);
xlabel('Month');
ylabel('Load');
25 grid
figure

nregras = 18;

30 for n=1:Np

    for i = 1:nregras
        za(i) = -10000;
    end

35     for i = 1:nregras

        if(x(n) >= regras(i,1) & x(n) <= regras(i,2) & x1(n) >= regras(i,3)
            & x1(n) <= regras(i,4) & x2(n) >= regras(i,5) & x2(n) <= regras(i
            ,6))
            u=0.3333*(regras(i,8)-regras(i,7))*((x(n)-regras(i,1))/(regras(i
            ,2)-regras(i,1))+(x1(n)-regras(i,3))/(regras(i,4)-regras(i,3)
            )+(x2(n)-regras(i,5))/(regras(i,6)-regras(i,5)))+regras(i,7);
40         if u == 0
            u = regras(i,8);
        end
        za(i) = u;
    end

45     end

    zaTotal=0;na=0;

50     for i = 1:nregras

        if(za(i) > -10000)
            zaTotal = zaTotal + za(i);
            na = na + 1;
55         end

```

```

        end

        if(na == 0)
60         na = 1;
        end

        zaTotal = zaTotal / na;
        ye(n) = zaTotal;
65
    end

    erro = ((y' - ye).^2);
    mean(erro)
70
    plot(vn,y', 'b',vn, ye, 'r');
    xlabel('Month');
    ylabel('Load');
    legend('actual', 'Rough Sets')
75
    grid
    figure;

    %agora vamos tentar prever algo...
    x = cargaTeste(1:24,1);
80    x1 = cargaTeste(1:24,2);
    x2 = cargaTeste(1:24,3);
    y = cargaTeste(1:24,4);
    vn = 1:24;

85    ye2(1) = x(1);

    for n=1:24

        for i = 1:nregras
90         za(i) = -10000;
        end

        %verificar alterações nas regras.
        %pra valores menores
95        if(ye2(n) < minCarga)
            for i = 1:nregras
                if minCarga == regras(i, 1)
                    regras(i,1) = ye2(n);
                end
            end
        end
    end
end

```

```

        end
100     end
        minCarga = ye2(n);
    end

    %pra valores maiores
105     if(ye2(n) > maxCarga)
        for i = 1:nregras
            if maxCarga == regras(i, 2)
                regras(i,2) = ye2(n);
            end
110         end
        maxCarga = ye2(n);
    end

    for i = 1:nregras
115
        if(ye2(n) >= regras(i,1) & ye2(n) <= regras(i,2) & x1(n) >= regras(i,3) & x1(n) <= regras(i,4) & x2(n) >= regras(i,5) & x2(n) <=
            regras(i,6))
            u=0.3333*(regras(i,8)-regras(i,7))*((ye2(n)-regras(i,1))/(regras(i,2)-regras(i,1))+(x1(n)-regras(i,3))/(regras(i,4)-regras(i,3))+(x2(n)-regras(i,5))/(regras(i,6)-regras(i,5)))+regras(i,7);
            if u == 0
                u = regras(i,8);
120             end
            za(i) = u;
        end

    end

125
    zaTotal=0;na=0;

    for i = 1:nregras

130         if(za(i) > -10000)
            zaTotal = zaTotal + za(i);
            na = na + 1;
        end

135     end

```

```

    if(na == 0)
        na = 1;
    end
140
    zaTotal = zaTotal / na;
    if n~= 24
        ye2(n+1) = zaTotal;
    end
145
end

plot(vn,y,'b',vn,ye2(1:24),'r');

150 erroquadratico = ((y - ye2') .^2);
    erroquadratico = mean(erroquadratico);
    erroquadratico

    erromad = (y - ye2');
155 erromad = mean(abs(erromad));
    erromad

    soma = 0;
    cont=0;
160 for i = 1:24
        if(y(i) ~= 0)
            soma = soma + abs((ye2(i) - y(i))/y(i));
            cont = cont + 1;
        end
165 end

mape = soma/cont;
mape

```

APÊNDICE C – Programa para construção de modelos fuzzy para previsão

A construção de modelos fuzzy de previsão se dá a partir de aplicação desenvolvida em MatLab conforme listagem seguinte.

É apresentado o código necessário para construção de modelos para previsão de enchentes. O código para previsão das demais séries desse trabalhos são semelhantes.

Listagem C.1: Listagem do programa de para construção de modelos fuzzy para previsão de enchentes

```

load D:\UNIFEI\Mestrado\Dissertacao2\Experimentos\enchente\serie_99; %
    carrega serie temporal.
Np=108; % Parâmetros da serie.

5 % Inicializa variavel com dados da série.
for t = 1:Np
    x(t) = serie_99(t);
end

10 % Contrução do conjunto de treinamento.
x1=x(1:88); x2=x(2:89); x3=x(3:90); x4=x(4:91); y=x(5:92);
vx = [x1' x2' x3' x4']; y = y';
DadosTreina = [vx y];

15 params = [100 0 0.01 0.4 1.5];%parametros para o backpropagation.
params2 = [2 100 1e-5 1];%parametros para o FCM
%executa FCM e gera FIS inicial
in_fuzzy = genfis3(vx,y,'sugeno',45, params2);
%Realiza treinamento e gera FIS final

20 [out_fuzzy, error, stepsize] = anfis(DadosTreina, in_fuzzy, params);
%Avaliação do modelo fuzzy com os dados de treinamento.
ye = evalfis(vx, out_fuzzy);

```

```
%imprime grafico da modelagem
vn = 1:88;
25 plot(vn,y,'b',vn,ye,'r');
figure

%inicia conjunto de teste
x1=x(89); x2=x(90); x3=x(91); x4=x(92);
30 vx = [x1 x2 x3 x4];
saida=x(1:92);
base=92;
%realiza previsao
for n=1:12
35 % Avaliação do modelo de previsao.
    ye = evalfis(vx,out_fuzzy);
    saida(base+n)=ye;
    x1 = x2; x2 = x3; x3 = x4; x4 = ye;
    vx = [x1 x2 x3 x4];
40 end
%imprime grafico da previsao
vye = saida(93:104);
vy = x(93:104);
vn = 1:12;
45 plot(vn,vy,'b',vn,vye,'r');
figure
```

Referências

- AHN, B.; CHO, S.; KIM, C. The integrated methodology of rough set theory and artificial neural network for business failure prediction. *Expert Systems with Applications*, v. 18, p. 65–74, 2000.
- BACCI, L. A. *Combinação de Métodos de Séries Temporais para Previsão da Demanda de Café no Brasil*. 2007. Dissertação de Mestrado (UNIFEI).
- BOWERMAN, B. L.; O'CONNELL, R. T.; KOEHLER, A. B. *Forecasting, Time Series, and Regression*. 4th. ed. [S.l.]: Thomson, 2005.
- CARPINTEIRO, O. A. et al. Long-term load forecasting via a hierarchical neural model with time integrators. *Electric Power Systems Research*, p. 371–378, 2007.
- CASTILHO, O.; MELIN, P. Hibrid intelligent systems for time series prediction using neural networks, fuzzy logic, and fractal theory. *IEEE*, n. 6, p. 1395–1408, 2002.
- CHEN, D.-W.; ZHANG, J.-P. Time series prediction based on ensemble anfis. *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, p. 18–21, 2005.
- CHEN, M.-S.; YING, L.-C.; PAN, M.-C. Forecasting tourist arrivals by using the adaptive network-based fuzzy inference system. *Experts Systems with Applications*, v. 37, p. 1185–1191, 2010.
- CHEN, M.-S.; YING, L.-C.; PAN, M.-C. Forecasting tourist arrivals by using the adaptive network-based fuzzy inference system. *Expert Systems with Applications*, v. 37, p. 1185–1191, 2010.
- DENAİ, M. A.; PALIS, F.; ZEGHBIB, A. Anfis based automatic voltage regulator with hybrid learning algorithm. *IEEE Transaction on Systems, Man and Cybernetics*, p. 397–401, 2007.
- DIMITRAS, A. I. et al. Business failure prediction using rough sets. *European Journal of Operational Research*, v. 114, p. 263–280, 1999.
- DUBOIS, D.; PRADE, H.; YANGER, R. R. *Fuzzy Information Engineering: A Guided Tour of Applications*. 1st. ed. [S.l.: s.n.], 1996.
- FAHIMIFARD, S. et al. Application of anfis to agricultural economic variables forecasting. case study: Poultry retail price. *Journal of Artificial Intelligence*, v. 2, n. 2, p. 65–72, 2009.
- FAUSTINO, C. P. et al. Time series forecasting through rule-based models obtained via rough sets. *Artificial Intelligence Review*, 2011.

- FEI, S. wei; LIU, C. liang; MIAO, Y. bin. Support vector machine with genetic algorithm for forecasting of key-gas ratios in oil-immersed transformer. *Expert Systems with Applications*, v. 36, 2009.
- FIRAT, M.; GÜNGÖR, M. River flow estimation using adaptive neuro fuzzy inference system. *Mathematics and Computers in Simulation*, v. 75, p. 87–96, 2007.
- FREIMAN, J. P. *Utilização de Redes neurais Artificiais na Previsão de Indicadores Financeiros para avaliação econômica de Negócios em Situação de Risco*. 2006. Dissertação de Mestrado (UNIFEI).
- GOH, C.; LAW, R. Incorporating the rough sets theory into travel demand analysis. *Tourism Management*, v. 24, p. 511–517, 2003.
- HAYKIN, S. *Neural Networks, a Comprehensive Foundation*. 1st. ed. [S.l.]: Macmillan Publishing Company, 1994.
- HAYKIN, S. *Neural Networks: Principles and Practices*. 1st. ed. [S.l.]: Bookman, 2000.
- HENTSCHEL, L. All in the family. nesting symmetric and asymmetric garch models. *Journal of Financial Economics*, v. 39, 1998.
- HERBERT, J.; YAO, J. Time series data analysis with rough sets. *Applied Soft Computing*, v. 9, p. 1000–1007, 2009.
- JANG, J.-S. R. Anfis: Adaptive-network-based fuzzy inference system. *IEEE Transaction on Systems, Man and Cybernetics*, v. 23, n. 3, p. 665–685, 1993.
- JANG, J.-S. R.; SUN, C.-T. Neuro-fuzzy modeling and control. *Proceedings of the IEEE*, v. 83, p. 378–406, 1995.
- JANG, J.-S. R.; SUN, C.-T.; MIZUTANI, E. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. 1st. ed. [S.l.]: Prentice Hall, 1997.
- KASABOV, N.; SONG, Q. Denfis: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 2001.
- KLIR, G. J.; YUAN, B. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. 1st. ed. [S.l.]: Prentice Hall, 1995.
- KURIAN, C. P. et al. Anfis model for the time series prediction of interior daylight illuminance. *AIML Journal*, v. 6, 2006.
- LI, R.-J.; XIONG, Z.-B. Forecasting stock market with fuzzy neural networks. *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, p. 18–21, 2005.
- LIAO, G.-C.; TSAO, T.-P. Application of fuzzy neural networks and artificial intelligence for load forecasting. *Electric Power Systems Research*, v. 70, p. 237–244, 2007.
- MAKRIDAKIS, S.; WHEELWRIGHT, S. C.; HYNDMAN, R. J. *Forecasting, Methods and Applications*. 3th. ed. [S.l.]: John Willey & Sons, 1998.

- MCCULLOCH, R. E.; TSAY, R. S. Statistical analysis of economic time series via markov switching models. *Journal of Time Series Analyses*, p. 523–539, 1994.
- MILLS, T. C. *The Econometric Modelling of Financial Time Series*. 2nd. ed. [S.l.]: Cambridge University Press, 1999.
- MING-BAO, P.; XIN-PING, Z. Traffic flow prediction of chaos time series by using subtractive clustering for fuzzy neural network modeling. *Second International Symposium on Intelligent Information Technology Application*, 2008.
- MITRA, P. et al. Anfis based modelling and control of non-linear systems: A tutorial. *Universities Power Engineering Conference, 2007. UPEC 2007. 42nd International*, 2004.
- MITRA, S.; PAL, S. K. Self-organizing neural network as a fuzzv classifier. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 24, n. 3, p. 385–399, 1994.
- MOREIRA, D. A. *Administração da Produção e Operações*. 1st. ed. [S.l.]: Pioneira Thomson Learning, 2004.
- MROZEK, A.; SKABEK, K. *Rough sets in economic applications. Em: Polkowski, L., Skowron, A. (Eds.), Rough Sets in Knowledge Discovery*. [S.l.]: Physica-Verlag, Wurzburg, 1998. 238-271 p.
- NAUCK, D.; KRUSE, R. Nefclass - a neuro-fuzzy approach for the classification of data. *Proceedings of the 1995 ACM symposium on Applied computing*, p. 461–465, 1995.
- NAYAK, P. et al. A neuro-fuzzy computing technique for modeling hydrological time series. *Journal of Hydrology*, v. 291, p. 52–66, 2004.
- OHRN, A.; KOMOROWSKI, J. Rosetta - a roug sets toolkit for analyses of data. *Third International Joint Conference on Information Sciences*, p. 403–407, 1997.
- PALIT, A. K.; POPOVIC, D. *Computational Intelligence In Time Series Forecasting*. 1st. ed. [S.l.]: Springer, 2005.
- PAWLAK, Z. Rough sets. *International Journal of Information and Computer Sciences*, v. 11, p. 341–356, 1982.
- PAWLAK, Z. Rough set theory and its applications. *Journal of Communications and Information Tecnhnology*, p. 7–10, 2003.
- PAWLAK, Z. et al. Rough sets. *Communications of ACM*, v. 38, p. 89–95, 1995.
- PAWLAK, Z.; SKOWRON, A. Rudiments of rough sets. *Information Sciences*, v. 117, p. 3–27, 2007.
- PINHEIRO, C. et al. *Granular Synthesis of Rule-Based Models and Function Approximation using Rough Sets. Chapter in the book Novel Developments in Granular Computing*. 1st. ed. [S.l.]: JingTao Yao, IGI Global, 2010a. 408-425 p.
- PINHEIRO, C. et al. Modelos baseados em conjuntos aproximados. *XVIII Congresso Brasileiro de Automática*, 2010b.

- POEL, D. *Rough Sets for Database Marketing. Em: Polkowski, L., Skowron, A. (Eds.), Rough Sets in Knowledge Discovery*. [S.l.]: Physica-Verlag, Wurzburg, 2009. 324-335 p.
- SHEN, L.; LOH, H. T. Applying rough sets to market timing decisions. *Decision Support Systems*, 2003.
- SLOWINSKI, R.; ZOPOUNIDIS, C. Application of the rough set approach to evaluation of bankruptcy risk. *International Journal of Intelligent Systems in Accounting, Finance & Management*, v. 4, p. 27–41, 1995.
- SYED-AHMAD, M. N. et al. Short-term load forecasting using adaptive neuro-fuzzy inference system (anfis). application to aleppo load demand. 2007.
- TAY, F. E.; CAO, L. Application of support vector machines in financial time series forecasting. *Omega*, v. 29, 2001.
- TAY, F. E.; SHEN, L. Economic and financial prediction using rough sets model. *European Journal of Operational Research*, v. 141, p. 641–659, 2002.
- TSAY, R. S. Testing and modeling threshold autoregressive processes. *Journal of the American Statistical Association*, v. 84, p. 231–240, 1989.
- WANG, L.-X.; MENDEL, J. M. Generating fuzzy rules by learning from example. *IEEE*, v. 22, n. 6, Novembro 1992.
- XIAO, Z. et al. Bp neural network with rough set for short term load forecasting. *Expert Systems with Applications*, v. 36, p. 273–279, 2009.
- YING, L.-C.; PAN, M.-C. Using adaptive network based fuzzy inference system to forecast regional electricity loads. *Energy Conversion and Management*, v. 49, p. 205–211, 2008.
- ZADEH, L. A. Fuzzy sets. *Information and Control*, v. 8, p. 338–353, 1965.
- ZADEH, L. A. Fuzzy logic = computing with words. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 4, n. 2, p. 103–111, 1996.
- ZHANG, G.; PATUWO, B. E.; HU, M. Y. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, p. 35–62, 1998.
- ZIMMERMANN, H. J. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. 4st. ed. [S.l.]: Prentice Hall, 2001.