



**Universidade Federal de Minas Gerais**  
Programa de Pós-Graduação em Engenharia Elétrica

---

## **Modelagem Nebulosa Evolutiva: Novas Topologias e Algoritmos de Aprendizagem**

Tese de Doutorado submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica.

por

**André Paim Lemos**

Bacharel em Ciência da Computação - UFMG

Mestre em Engenharia Elétrica - UFMG

**Orientador:** Prof. Dr. Walmir Matos Caminhas

**Co-orientador:** Prof. Dr. Fernando Antônio Campos Gomide

Fevereiro – 2011

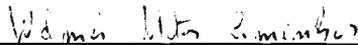
**"Modelagem Nebulosa Evolutiva: Novas Topologias  
e Algoritmos de Aprendizagem"**

**André Paim Lemos**

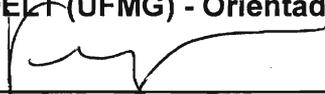
Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 02 de fevereiro de 2011.

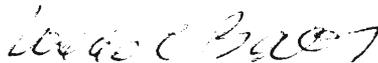
Por:



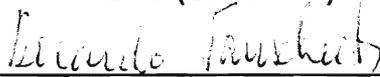
Prof. Dr. Walmir Matos Caminhas  
DELT (UFMG) - Orientador



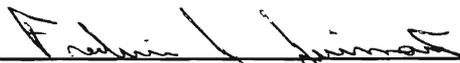
Prof. Dr. Fernando Antônio Campos Gomide  
DCA (UNICAMP) - Co-Orientador



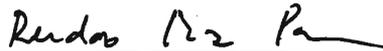
Prof. Dr. Laécio Carvalho de Barros  
IMECC (UNICAMP)



Prof. Dr. Ricardo Tanscheit  
Engenharia Elétrica (PUC-Rio)



Prof. Dr. Frederico Gadelha Guimarães  
DEE (UFMG)



Prof. Dr. Reinaldo Martinez Palhares  
DELT (UFMG)

# Resumo

Esse trabalho tem como objetivo propor novas topologias e algoritmos de aprendizagem para sistemas nebulosos evolutivos. Sistemas nebulosos evolutivos são definidos como sistemas nebulosos inteligentes, com alto grau de flexibilidade e autonomia. Esses sistemas são capazes de realizar tarefas, tais como identificação de sistemas, controle e classificação de padrões, em um ambiente dinâmico não estacionário, ajustando seus parâmetros e estrutura a partir de um fluxo de dados. Esse trabalho propõe duas técnicas de modelagem nebulosa evolutiva. A primeira técnica é baseada em funções de pertinência Gaussianas multivariadas definidas a partir de um algoritmo de agrupamento recursivo não supervisionado, baseado no conceito do aprendizado participativo. O aprendizado participativo é um modelo de aprendizagem baseado na forma de aprendizagem humana, que tem como principal característica a robustez, uma vez que o impacto que uma nova amostra causa na revisão dos conhecimentos sobre um sistema depende do grau de compatibilidade dessa amostra com os conceitos já adquiridos. Em seguida, uma nova abordagem para modelagem nebulosa evolutiva é proposta, baseada em árvores de regressão nebulosas evolutivas, construídas a partir de um fluxo de dados. O algoritmo incremental proposto nesse trabalho atualiza a topologia da árvore a partir de um fluxo de dados, utilizando testes estatísticos de seleção de modelos. O teste é computado recursivamente e constrói a árvore substituindo folhas por subárvores, que melhorem a qualidade do modelo resultante. Para avaliar a qualidade do modelo, o teste utilizado leva em consideração a precisão e o número de parâmetros livres, gerando modelos eficientes e parcimoniosos. Os modelos nebulosos evolutivos propostos nesse trabalho são avaliados em problemas de previsão de séries temporais, identificação de sistemas dinâmicos não lineares, seleção de características e detecção e diagnóstico de falhas. Os resultados obtidos sugerem os modelos como alternativas promissoras para utilização em ambientes não estacionários, ou para processamento de um grande volume de dados, em que a utilização de algoritmos tradicionais acarretaria em um elevado custo computacional.

**Palavras-chave:** Sistemas Nebulosos Evolutivos, Modelagem Adaptativa .

# Abstract

This work aims to introduce new evolving fuzzy topologies and learning algorithms. Evolving fuzzy systems are defined as a new class of intelligent fuzzy systems, with a high flexibility and autonomy. These systems are able to address problems like nonlinear system identification, control and pattern classification in a dynamic changing environment, adapting its parameters and structure based on a data stream. This work proposes two alternative evolving fuzzy modeling techniques. The first technique uses multivariate Gaussian membership functions defined by a non supervised recursive clustering algorithm based on participatory learning. Participatory learning is a learning model based on the human learning with an essential characteristic of robustness, since the impact of a new observation in causing learning or belief revision depends on its compatibility with the current system belief. Next, a novel approach for evolving fuzzy modeling is proposed, using fuzzy linear regression trees built from a stream of data in an incremental manner. The learning algorithm proposed grows the trees from a stream of data using a statistical model selection test. This model selection test is done recursively and grows the tree replacing leaves with subtrees that improves the model quality. To evaluate the model quality, the statistical model selection test takes into account the accuracy and number of the parameters of the resulting model, generating highly efficient models and avoiding over fitting. Experiments considering nonlinear system identification, time series forecasting, feature selection and fault detection and diagnosis are performed to evaluate the evolving systems proposed in this work. The results suggests these models as a promising approach for adaptive system modeling or to be used on large datasets, where the application of traditional approaches would lead to an excessive computational cost.

**Keywords:** Evolving Fuzzy Systems, Adaptive Modeling.

# Agradecimentos

Ao Prof. Walmir Caminhas, meu orientador, pelo apoio e amizade. Por sempre me incentivar a seguir a carreira acadêmica, desde os tempos de especialização (CEAI). Por me mostrar que a pesquisa que realizamos pode e deve ser aplicada na indústria, o que acredito ser de extrema importância para o desenvolvimento do país.

Ao Prof. Fernando Gomide, meu co-orientador, por seu exemplo de carreira acadêmica, simplicidade e grande apoio em todas as etapas do desenvolvimento desse trabalho.

Aos meus pais e irmãs, pelo amor incondicional, carinho e por sempre apoiarem meus estudos.

Aos Professores Benjamim Menezes, Carlos Júlio e Reinaldo Palhares, pela amizade e incentivo durante esses anos de pós-graduação.

Aos colegas do LabIComp, pelos momentos de descontração e pelas contribuições científicas.

Ao CNPq, pelo apoio financeiro, N<sup>o</sup> 141323/2009-4.

Aos professores do CPDEE que contribuíram para minha formação acadêmica.

Aos meus amigos e primos, pela amizade e momentos de descontração, que tanto me ajudaram a concluir esse trabalho.

À Tati, pelo amor e pela compreensão que me dedicou durante todos esses anos. Por suportar minhas ausências e por me ajudar em todas as dificuldades que passei.

# Sumário

<b>Lista de Figuras</b>	<b>vi</b>
<b>Lista de Tabelas</b>	<b>viii</b>
<b>Lista de Algoritmos</b>	<b>ix</b>
<b>Lista de Símbolos</b>	<b>x</b>
<b>Lista de Acrônimos</b>	<b>xi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação e Relevância . . . . .	1
1.2 Objetivos . . . . .	3
1.3 Organização do Trabalho . . . . .	4
1.4 Contribuições da Tese . . . . .	5
1.4.1 Trabalhos em Periódicos . . . . .	5
1.4.2 Capítulos de Livro . . . . .	5
1.4.3 Trabalhos em Congressos . . . . .	6
1.4.4 Trabalhos Submetidos . . . . .	7
<b>2 Modelagem Nebulosa Evolutiva</b>	<b>8</b>
2.1 Introdução . . . . .	8
2.2 Modelos de Regressão Nebulosos Evolutivos . . . . .	8
2.2.1 Takagi-Sugeno Evolutivo (eTS) . . . . .	9
2.2.2 Outros Modelos Funcionais Evolutivos . . . . .	13
2.3 Classificadores Nebulosos Evolutivos . . . . .	17
<b>3 Modelagem Baseada no Agrupamento Evolutivo Participativo Gaussiano</b>	<b>19</b>
3.1 Introdução . . . . .	19

3.2	Agrupamento Evolutivo Participativo Gaussiano . . . . .	20
3.3	Funções de Pertinência Gaussianas Multivariáveis . . . . .	30
3.4	Modelo Funcional Evolutivo Gaussiano Multivariável . . . . .	36
3.5	Modelo Linguístico Evolutivo para Detecção e Diagnóstico Adaptativo de Falhas em Sistemas Dinâmicos . . . . .	39
3.5.1	Classificador Nebuloso Evolutivo . . . . .	41
3.5.2	Metodologia de Detecção de Diagnóstico Adaptativo de Falhas . . . . .	41
<b>4</b>	<b>Árvores de Regressão Linear Nebulosas Evolutivas</b>	<b>46</b>
4.1	Introdução . . . . .	46
4.2	Árvores de Regressão Linear Nebulosas Evolutivas . . . . .	47
4.2.1	Estrutura do Modelo . . . . .	47
4.2.2	Algoritmo de Aprendizado Incremental . . . . .	53
4.3	Árvores de Regressão Linear Nebulosas com Seleção de Características . . . . .	60
4.3.1	Estrutura do Modelo . . . . .	61
4.3.2	Algoritmo de Aprendizado Incremental . . . . .	63
<b>5</b>	<b>Resultados Numéricos</b>	<b>67</b>
5.1	Introdução . . . . .	67
5.2	Problemas de Regressão . . . . .	67
5.2.1	Previsão de Carga . . . . .	68
5.2.2	Identificação de Sistemas Não Lineares . . . . .	74
5.2.3	Problema de Alta Dimensão . . . . .	78
5.3	Seleção de Características . . . . .	80
5.3.1	Dados Artificiais . . . . .	80
5.3.2	Previsão de Séries Temporais . . . . .	82
5.4	Detecção e Diagnóstico Adaptativo de Falhas . . . . .	86
5.4.1	Detecção e Diagnóstico de Falhas em um Atuador . . . . .	87
5.4.2	Detecção e Diagnóstico de Falhas em um Motor de Indução . . . . .	91
<b>6</b>	<b>Conclusões e Perspectivas</b>	<b>98</b>
6.1	Propostas de Continuidade . . . . .	99
	<b>Referências bibliográficas</b>	<b>101</b>

# Lista de Figuras

2.1	Mecanismo de atualização da base de regras . . . . .	9
2.2	Exemplo de geração de funções de pertinência a partir de grupos . . . . .	11
2.3	Ilustração da ideia do potencial . . . . .	12
2.4	Topologia do modelo SOFMLS . . . . .	16
3.1	Aprendizado participativo . . . . .	23
3.2	Exemplo unidimensional da medida de compatibilidade . . . . .	24
3.3	Limiar de compatibilidade em função da dimensão do espaço de entrada . . . . .	25
3.4	Probabilidade de se observar $v$ violações do limiar de compatibilidade em uma sequência de 50 observações dado $\alpha = 0.1$ . . . . .	27
3.5	Índice de alerta, definido como a probabilidade de ser observar menos de $v$ violações do limiar em uma sequência de $w = 50$ observações dado $\alpha = 0.1$ . . . . .	28
3.6	Estrutura de grupos do espaço de entrada . . . . .	32
3.7	Funções de pertinência Gaussianas univariáveis (projeção dos grupos da figura 3.6 nos eixos do espaço de entrada) . . . . .	33
3.8	Relações nebulosas induzidas pelo antecedente das regras utilizando a $t$ -norma produto como operador lógico $E$ . . . . .	34
3.9	Funções de pertinência Gaussianas multivariáveis definidas a partir do centro e matriz de dispersão de cada grupo . . . . .	35
3.10	Estrutura do classificador nebuloso evolutivo proposto para diagnóstico de falhas . . . . .	42
3.11	Metodologia de detecção e diagnóstico adaptativo de falhas proposta . . . . .	43
4.1	Exemplo de uma árvore de regressão linear . . . . .	48
4.2	Partição do espaço de entrada induzida pela árvore de regressão linear ilustrada na figura 4.1 . . . . .	49
4.3	Funções de pertinência sigmoidais descrevendo <i>menor que</i> and <i>maior que</i> 5 . . . . .	50
4.4	Partição do espaço de entrada induzida por uma árvore de regressão linear nebulosa . . . . .	51
4.5	Funções de pertinência extraídas da árvore de regressão linear nebulosa . . . . .	53
4.6	Ponto de corte candidato genérico . . . . .	54
4.7	Caminhos entre o nó raiz e as folhas que passam pela função de pertinência $\mu_{x_2 < 2}$ . . . . .	56
4.8	Árvore de regressão linear nebulosa com seleção de características . . . . .	62
4.9	Exemplos de pontos de corte candidatos . . . . .	64

5.1	Valores normalizados de carga (Kw/Hora) correspondentes aos 3 primeiros dias de Agosto de 2000 . . . . .	69
5.2	Resultado da previsão de carga para o eMG . . . . .	70
5.3	Resultado da previsão de carga para o eFT . . . . .	71
5.4	Estrutura de grupos do espaço de entrada gerados pelo modelo eMG para o problema de previsão de carga . . . . .	71
5.5	Árvore de regressão linear nebulosa evolutiva para o problema de previsão de carga	72
5.6	Resultado da identificação para o eMG . . . . .	75
5.7	Resultado da identificação para o eFT . . . . .	75
5.8	Árvore de regressão linear nebulosa evolutiva para o problema de identificação .	76
5.9	Resultado do modelo eMG para o problema de alta dimensão eFT . . . . .	79
5.10	Resultado do modelo eFT para o problema de alta dimensão . . . . .	79
5.11	Árvore de regressão linear nebulosa evolutiva para o problema de alta dimensão	80
5.12	Resultado da identificação para o xFT . . . . .	82
5.13	Árvore de regressão linear nebulosa evolutiva com seleção de características para o problema de identificação . . . . .	83
5.14	Resultado do modelo xFT para o problema de alta dimensão . . . . .	85
5.15	Árvore de regressão linear nebulosa evolutiva com seleção de características para o problema de alta dimensão . . . . .	85
5.16	Diagrama esquemático do atuador . . . . .	88
5.17	Diagrama do simulador do atuador . . . . .	88
5.18	Cenário de falha 1 . . . . .	89
5.19	Cenário de falha 2 . . . . .	90
5.20	Fluxo de dados simulados para o problema do atuador . . . . .	90
5.21	Saída do classificador para o problema do atuador . . . . .	91
5.22	Diagrama do modelo utilizado . . . . .	93
5.23	Curto circuito na fase A ( $t > 1$ segundo) . . . . .	94
5.24	Sequência de modos de operação simulados para o problema do motor . . . . .	95
5.25	Saída do classificador para o problema do motor . . . . .	96

# Lista de Tabelas

3.1	Número Médio de Violações do Limiar de Compatibilidade em Função do Crescimento da Dimensão dos Dados . . . . .	26
5.1	Desempenho comparativo para a previsão de carga . . . . .	73
5.2	Avaliação do teste <i>MGN</i> para o problema de previsão de carga . . . . .	73
5.3	Desempenho comparativo para o problema de identificação . . . . .	77
5.4	Avaliação do teste <i>MGN</i> para o problema de identificação . . . . .	77
5.5	Desempenho comparativo para o problema de alta dimensão . . . . .	80
5.6	Avaliação do teste <i>MGN</i> para o problema de alta dimensão . . . . .	81
5.7	Desempenho comparativo do modelo xFT para o problema de identificação . . . . .	84
5.8	Avaliação do teste <i>MGN</i> para o problema de seleção de características . . . . .	84
5.9	Desempenho comparativo do modelo xFT para o problema de alta dimensão . . . . .	86
5.10	Avaliação do teste <i>MGN</i> do modelo xFT para o problema de alta dimensão . . . . .	86
5.11	Modos de Operação Simulados . . . . .	89
5.12	Índices de desempenho para o problema do atuador . . . . .	92
5.13	Modos de operação simulados para os experimentos do motor de indução . . . . .	93
5.14	Índices de desempenho para os dados do motor . . . . .	97

# Lista de Algoritmos

1	Algoritmo de atualização da estrutura do modelo eTS . . . . .	13
2	Algoritmo de agrupamento evolutivo participativo Gaussiano . . . . .	30
3	Algoritmo de modelagem funcional evolutiva proposto . . . . .	38
4	Algoritmo proposto para detecção e diagnóstico adaptativo de falhas . . . . .	45
5	Algoritmo para calcular os valores de pertinência e a saída dos modelos lineares de todas as folhas da árvore . . . . .	52
6	Algoritmo para computar os parâmetros dos pontos de corte candidatos e atualizar a árvore . . . . .	57
7	Algoritmo de aprendizagem utilizado para evoluir a árvore de regressão linear nebulosa . . . . .	59
8	Algoritmo de aprendizagem utilizado para evoluir a árvore de regressão linear nebulosa com seleção de características . . . . .	66

# Lista de Símbolos

$n$	-	número de amostras
$m$	-	dimensão do espaço de entrada
$k$	-	índice de uma amostra
$x^k$	-	amostra de dimensão $1 \times m$
$y_i$	-	saída da regra $i$
$c_i^k$	-	centro do grupo $i$ na iteração $k$ , de dimensão $1 \times m$
$\Sigma_i^k$	-	matriz de dispersão do grupo $i$ na iteração $k$ , de dimensão $m \times m$
$g^k$	-	número de grupos existentes na iteração $k$
$\beta$	-	taxa de aprendizado
$\rho_i^k$	-	medida de compatibilidade entre $x^k$ e $c_i^k$
$a_i^k$	-	índice de alerta do grupo $i$
$T_\rho$	-	limiar da medida de compatibilidade
$T_a$	-	limiar do índice de alerta
$M(x^k, c_i^k)$	-	distância M entre $x^k$ e $c_i^k$
$\alpha$	-	nível de significância
$\chi_{m,\alpha}^2$	-	intervalo de confiança superior de uma distribuição <i>Qui-Quadrado</i> com $m$ graus de liberdade
$\sigma^k$	-	ocorrência relacionada à amostra $x^k$
$w$	-	tamanho da janela utilizada para computar $a_i^k$
$v^k$	-	número de violações do limiar $T_\rho$ , computado baseando-se nas últimas $w$ observações
$\Sigma_{init}$	-	matriz de dispersão inicial
$T_{\mu_y}$	-	limiar do grau de pertinência do classificador nebuloso evolutivo
$\sigma$	-	dispersão da função de pertinência sigmoidal
$l^k$	-	número de folhas da árvore de regressão linear nebulosa na iteração $k$
$s^k$	-	número de nós internos da árvore de regressão linear nebulosa na iteração $k$
$w_i$	-	grau de ativação associado a folha $i$
$\gamma$	-	número de pontos de corte candidatos

# Lista de Acrônimos

DENFIS	-	<i>Dynamic Evolving Neural-Fuzzy Inference System</i>
eFT	-	<i>evolving Fuzzy Tree</i>
eMG	-	<i>evolving Multivariable Gaussian</i>
eTS	-	<i>evolving Takagi-Sugeno</i>
NDEI	-	<i>Non Dimensional Error Index</i>
RMSE	-	<i>Root Mean Square Error</i>
xFT	-	<i>eXtended evolving Fuzzy linear regression Tree with feature selection</i>
xTS	-	<i>eXtended Takagi-Sugeno</i>

# Capítulo 1

## Introdução

### 1.1 Motivação e Relevância

Ao longo das últimas décadas, sistemas nebulosos baseados em regras e suas derivações híbridas vêm sendo amplamente utilizados para modelagem de sistemas complexos. Inicialmente, essa modelagem era realizada através de conhecimento adquirido de especialistas (Mamdani e Assilian, 1975). Durante a década de 90, uma nova tendência surgiu: a modelagem baseada em dados (Angelov e Guthke, 1997; Babuska, 1998; Chiu, 1994; Jang, 1993; Jang e Sun, 1997; Kasabov, 1996; Yager e Filev, 1994), em que o modelo era construído baseando-se em informações extraídas a partir de um conjunto de dados e o conhecimento de especialistas era utilizado como informação complementar (Wang e Mendel, 1992a). As principais técnicas utilizadas nesse período eram algoritmos de agrupamento, mínimos quadrados linear e/ou técnicas de otimização não linear, para ajuste fino dos parâmetros do modelo (Jang e Sun, 1997).

O aumento contínuo na disponibilidade de grandes quantidades de informação tem motivado o desenvolvimento de modelos baseados em algoritmos incrementais capazes de processar fluxos de dados. O desafio de processar grandes fluxos de dados variantes no tempo, não estacionários, traz problemas únicos que requerem um esforço considerável no desenvolvimento de novas topologias e algoritmos de aprendizagem. Por exemplo, na indústria, máquinas sofrem desgaste, envelhecimento e falhas; na economia, indicadores de desempenho e índices de bolsas variam ao longo do tempo; em sistemas de comunicação, parâmetros e condições de transmissão de mídias também são sujeitos a variações contínuas. Esse crescente interesse por métodos de modelagem adaptativos motivou o desenvolvimento de sistemas nebulosos altamente adaptativos e inteligentes, denominados sistemas nebulosos evolutivos *eFS* (*evolving Fuzzy Systems*). Esses sistemas devem ser capazes de realizar tarefas, tais como identificação de sistemas, previsão de séries temporais, controle e classificação de padrões, em um ambiente dinâmico não estacionário (Angelov et al., 2008a).

Sistemas nebulosos evolutivos podem ser vistos como uma sinergia entre: sistemas nebulosos, utilizados como um mecanismo de compactação e representação de informação de maneira

evolutiva; e métodos recursivos de aprendizado de máquina (Kasabov e Filev, 2006). A grande maioria desses sistemas utilizam informação sobre a organização espacial das variáveis de entrada, ou entrada e saída, para definir um conjunto de regras de forma adaptativa, a partir de um fluxo de dados. Geralmente, a organização espacial dessas variáveis é estimada através de algoritmos de agrupamento recursivos não supervisionados (Oliveira e Pedrycz, 2007), capazes de processar novas amostras de forma incremental, adaptando os grupos existentes. Os grupos representam uma decomposição evolutiva do espaço de entrada (ou entrada e saída), gerando regras nebulosas, que podem ser adicionadas ou excluídas de um modelo com estrutura flexível, baseada na variação dinâmica dos dados. O restante dos parâmetros do modelo são ajustados por métodos recursivos de aprendizado de máquina.

Esses sistemas têm como principal característica a capacidade de adaptar seus parâmetros e, principalmente, sua estrutura de forma autônoma em tempo real, à medida que variações nos dados de entrada são detectadas. Esses sistemas representam uma mistura nebulosa de modelos (geralmente lineares) locais simples, cuja combinação é não linear e adaptativa, sendo particularmente apropriados para processamento de dados não estacionários em tempo real, ou processamento de uma grande massa de dados, em que a utilização de métodos tradicionais acarreta em um elevado custo computacional.

O termo “evolutivo” foi utilizado para diferir tais sistemas de sistemas adaptativos, pois esses sistemas, além de serem capazes de ajustar seus parâmetros a partir de dados (o que geralmente é atribuído ao termo adaptativo) também realizam uma adaptação de sua estrutura. Dessa maneira, o termo “evolutivo” foi utilizado para definir um nível mais alto de adaptação, em que, diferentemente de sistemas adaptativos, a estrutura do sistema não é fixa (Angelov e Zhou, 2006).

Além disso, o termo “evolutivo” não deve ser confundido com o “evolucionário”. Algoritmos evolucionários, tais como algoritmos genéticos (Goldberg, 1989) e programação genética (Koza, 1992), baseiam-se no processo de evolução que ocorre em populações de indivíduos e utilizam operadores baseados nos conceitos de seleção, cruzamento e mutação de cromossomos como mecanismos de adaptação. Já os sistemas nebulosos evolutivos baseiam-se no processo de evolução de indivíduos ao longo de sua vida, especificamente o processo de aprendizagem humano, baseado na geração e adaptação de conhecimento a partir de experiências (Angelov e Zhou, 2006). Assim, uma analogia trivial para o mecanismo de aprendizagem desses sistemas seria o processo de aprendizagem de um indivíduo ao longo de sua vida. Esse processo inicia-se a partir de um conjunto vazio de regras (conhecimento) e novas regras são aprendidas à medida que o indivíduo depara-se com novas experiências, que não podem ser explicadas pelas regras existentes. O processo de geração de regras é gradual e as regras não são fixas ou pré-determinadas. Além disso, regras podem ser revisadas para melhor se adaptarem às experiências vivenciadas.

O interesse por essa nova abordagem para o problema de modelagem nebulosa vem crescendo nos últimos anos. Diversos simpósios (IEEE IS2002, EFS06, GEFS08, ESDIS2009, EIS2010), tutoriais e seções especiais em congressos (FUZZ-IEEE/IJCNN204, NAFIPS 2005, WCCI2006, FUZZ-IEEE2007, SSCI2007, FUZZ-IEEE2007, IPMU2008, EUSFLAT/IFSA2009, IPMU2010, WCCI2010, SSCI2011) vêm sendo organizados para discutir avanços nessa área, além de um grupo de trabalho (*IEEE Computer Society Task Force on Adaptive and Evolving Fuzzy Systems*) criado para auxiliar o direcionamento das pesquisas.

Diversas aplicações de eFS são reportadas na literatura em áreas como sistemas autônomos (Angelov et al., 2008c; Zhou e Angelov, 2007), sensores e agentes inteligentes (Angelov e Kordon, 2010; Angelov et al., 2008b; Macias-Hernandez et al., 2007) monitoramento e controle de processos (Filev e Tseng, 2006; Lughofer, 2008,b; Wang e Vrbanek, 2008), biomedicina, bioinformática e nanotecnologia (Kasabov, 2007).

## 1.2 Objetivos

O objetivo principal desse trabalho é desenvolver procedimentos sistemáticos para modelagem de sistemas dinâmicos que possuam comportamento não linear, variante no tempo, utilizando o paradigma de sistemas nebulosos. Os procedimentos propostos devem possuir a capacidade de:

- expandir e/ou reduzir a topologia do modelo, assim como atualizar os parâmetros, ou seja, “evoluir”;
- trabalhar de forma incremental, ou seja, processar cada amostra apenas uma vez, e, caso necessário, em tempo real.

Duas metodologias de modelagem são propostas nesse trabalho. A primeira é definida como modelos nebulosos baseados em regras contendo funções de pertinência Gaussianas multivariadas nos antecedentes. A estrutura do modelo é definida a partir de um fluxo de dados através de um algoritmo de agrupamento recursivo, baseado no conceito do aprendizado participativo (Yager, 1990). Essa metodologia difere dos demais modelos nebulosos evolutivos baseados em regras por:

- usar funções de pertinência multivariadas para os conjuntos nebulosos dos antecedentes, com o objetivo de evitar perda de informação sobre a interação entre variáveis de entrada;
- possuir um mecanismo automático capaz de evitar a *maldição da dimensionalidade*<sup>1</sup> (Hastie et al., 2001) do espaço de entrada (mais detalhes nos próximos capítulos);
- ser baseada em uma metodologia de aprendizagem que provê um mecanismo de suavização de dados incompatíveis, ou seja, ruído ou *outliers*.

Conforme descrito anteriormente, a grande maioria das técnicas de modelagem nebulosa evolutiva propostas na literatura utilizam informação sobre a organização espacial das variáveis de entrada (ou entrada e saída), para definir um conjunto de regras nebulosas de forma adaptativa a partir de um fluxo de dados. Assim, esse trabalho também tem como objetivo propor técnicas alternativas de modelagem, que não sejam baseadas na organização espacial das variáveis. Para isso, esse trabalho propõe uma metodologia de construção de modelos nebulosos funcionais evolutivos baseados em árvores de regressão nebulosas construídas a partir de um

---

<sup>1</sup>do inglês *curse of dimensionality*

fluxo de dados. A topologia desses modelos é definida utilizando informação sobre a qualidade do modelo resultante. A qualidade é estimada levando-se em consideração a precisão e também o número de parâmetros livres, resultando em modelos eficientes, e evitando-se o problema de sobreajuste. Assim, essa metodologia de modelagem difere das demais por:

- possuir uma topologia em forma de árvore, que realiza uma partição recursiva do espaço de entrada;
- realizar a evolução do modelo baseado em sua qualidade e não na organização espacial das entradas (ou entradas e saídas).

Uma variação da árvore de regressão nebulosa também é proposta, tendo como objetivo abordar uma deficiência presente na maioria dos modelos nebulosos evolutivos. Geralmente, esses modelos tratam o problema de modelagem de sistemas não estacionários através de uma estrutura flexível, isto é, capaz de se modificar para lidar com variações no sistema modelado ao longo do tempo. Porém, essas abordagens ainda possuem uma limitação na flexibilidade do modelo: a definição das variáveis de entrada, que geralmente são definidas antes do processo de aprendizagem e, portanto, são fixas. Assim, uma variação da árvore de regressão nebulosa evolutiva é proposta capaz de realizar seleção de características, isto é, seleção incremental das variáveis de entrada do modelo, durante o processo de aprendizagem.

Por fim, este trabalho tem por objetivo mostrar que as abordagens propostas são promissoras na modelagem de sistemas dinâmicos não estacionários, variantes no tempo. A qualidade desses modelos é comprovada pela precisão obtida em problemas de regressão; pelo grau de automatismo alcançado para problemas de detecção e diagnóstico falhas e seleção de características; ou pela interpretabilidade dos modelos resultantes.

### 1.3 Organização do Trabalho

Após essa introdução, o capítulo 2 realiza uma breve revisão do presente estado da arte no ramo de modelagem nebulosa evolutiva. Trabalhos pioneiros e atuais são revisados com o objetivo de descrever as principais técnicas comumente utilizadas no desenvolvimento de modelos nebulosos evolutivos. Deficiências de alguns modelos encontrados na literatura, que inspiraram o desenvolvimento dos modelos propostos nesse trabalho, também são detalhadas nesse capítulo.

O capítulo 3 descreve a técnica proposta de modelagem evolutiva baseada em funções de pertinência Gaussianas multivariadas, utilizando um algoritmo de agrupamento recursivo não supervisionado baseado no aprendizado participativo. O algoritmo de agrupamento proposto é descrito, assim como modelos nebulosos evolutivos funcionais e linguísticos, cuja topologia é definida incrementalmente a partir desse algoritmo.

O capítulo 4 descreve a segunda técnica de modelagem nebulosa proposta, definida como árvores de regressão linear nebulosas evolutivas construídas a partir de um algoritmo de aprendizado incremental. Esse algoritmo define a topologia da árvore baseado em testes estatísticos, tendo como objetivo definir modelos precisos e parcimoniosos para problemas de regressão. Esse

capítulo também descreve uma variação da árvore proposta, que é capaz de realizar seleção de características, ou seja, seleção das entradas do modelo durante o processo de aprendizagem.

O capítulo 5 apresenta os resultados de experimentos realizados para avaliação do desempenho dos modelos nebulosos evolutivos propostos. O desempenho dos modelos foi avaliado em problemas de previsão de séries temporais, identificação de sistemas dinâmicos, seleção de características e detecção e diagnóstico adaptativo de falhas.

Finalmente, o capítulo 6 conclui o trabalho, resumindo suas contribuições e propondo os tópicos a serem desenvolvidos como continuidade.

## 1.4 Contribuições da Tese

Alguns trabalhos derivados deste texto já foram publicados, sendo duas publicações em periódicos, seis em congressos e dois capítulos de livros. Além disso, dois artigos foram submetidos para periódicos. A lista dos trabalhos é apresentada a seguir.

### 1.4.1 Trabalhos em Periódicos

1. LEMOS, A. P.; GOMIDE, Fernando ; CAMINHAS, W. M. Multivariable Gaussian Evolving Fuzzy Modeling System. IEEE Transactions on Fuzzy Systems, 2011, v. 19(1), p.91-104.
2. LEMOS, A. P.; CAMINHAS, W. M. ; GOMIDE, Fernando. Fuzzy Evolving Linear Regression Trees. Evolving Systems, 2011 (Aceito).

O primeiro item descreve o modelo funcional evolutivo proposto no capítulo 3 seção 3.4. O segundo item descreve a árvore de regressão linear nebulosa evolutiva proposta no capítulo 4 seção 4.2.

### 1.4.2 Capítulos de Livro

1. LEMOS, A. P. ; CAMINHAS, W. M. ; GOMIDE, Fernando. Fuzzy Multivariable Gaussian Evolving Approach for Fault Detection and Diagnosis. In: Hüllermeier, Eyke; Kruse, Rudolf; Hoffmann, Frank. (Org.). Computational Intelligence for Knowledge-Based Systems Design, Lecture Notes in Artificial Intelligence. 1st ed. : Springer Berlin / Heidelberg, 2010, v. 6178, p. 360-369
2. DANGELO, M. F. S. V. ; PALHARES, R. M. ; CAMINHAS, W. M. ; MAIA, R. D. ; LEMOS, A. P. ; INACIO, M. J. ; TAKAHASHI, R. H. C. Detecção de Falhas: Uma Revisão Com Aplicações. In: A. P. Feltrin; C. R. Minussi;M. C. M. Teixeira; R. A. R. Lazaro. (Org.). Tutoriais do XVIII Congresso Brasileiro de Automática. São Paulo: Cultura Acadêmica, 2010, v. , p. 1-47.

O primeiro item descreve a metodologia de detecção e diagnóstico adaptativo de falhas proposta no capítulo 3 seção 3.5. O segundo item é um tutorial apresentado no Congresso Brasileiro de Automática 2010. Uma das seções desse tutorial descreve uma metodologia de detecção de novidades, que foi utilizada como base no desenvolvimento do algoritmo de agrupamento recursivo não supervisionado descrito no capítulo 3 seção 3.2.

### 1.4.3 Trabalhos em Congressos

1. LEMOS, A. P. ; CAMINHAS, W. M. ; GOMIDE, Fernando. Evolving Fuzzy Linear Regression Trees. In: IEEE World Congress on Computational Intelligence, 2010, Barcelona. Proceedings of IEEE World Congress on Computational Intelligence (WCCI), 2010. p. 1-8.
2. LEMOS, A. P. ; CAMINHAS, W. M. ; GOMIDE, Fernando. Modelo Adaptativo Nebuloso para Detecção e Diagnóstico de Falhas. In: Congresso Brasileiro de Automática, 2010, Bonito, MS. XVIII Congresso Brasileiro de Automática, 2010. p. 1-8.
3. MAIA, R. D. ; INACIO, M. J. ; LEMOS, A. P. ; CAMINHAS, W. M. Aprendizado Participativo Aplicado à Detecção e Diagnóstico On-Line da Falhas em Sistemas Dinâmicos. In: IX Congresso Brasileiro de Redes Neurais / Inteligência Computacional, 2009, Ouro Preto. Anais do IX Congresso Brasileiro de Redes Neurais / Inteligência Computacional, 2009.
4. LEMOS, A. P. ; CAMINHAS, W. M. ; GOMIDE, Fernando. New Uninorm-Based Neuron Model and Fuzzy Neural Networks. In: North American Fuzzy Information Processing Society's 2010, 2010, Toronto. Proceedings of NAFIPS 2010, 2010. p. 1-6.
5. LEMOS, A. P.; KREINOVICH, V.; CAMINHAS, W. M. ; GOMIDE, Fernando. Universal approximation with uninorm-based fuzzy neural networks. In 2011 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), March 2011 (Aceito).
6. LEMOS, A. P. ; CAMINHAS, W. M. ; GOMIDE, Fernando. Evolving Fuzzy Linear Regression Trees with Feature Selection. In IEEE Symposium Series on Computational Intelligence - SSCI 2011, April 2011 (Aceito).

O primeiro item descreve uma versão simplificada da árvore de regressão linear nebulosa evolutiva inicialmente proposta. O segundo item descreve a aplicação da metodologia de detecção e diagnóstico adaptativo de falhas no monitoramento de um motor de indução. O terceiro item descreve uma ideia inicial relativa a aplicação de um algoritmo de agrupamento recursivo não supervisionado no monitoramento de sistemas dinâmicos. O quarto e quinto item descrevem um modelo proposto recentemente, definido como uma rede neural nebulosa (Pedrycz e Gomide, 2007) formada por neurônios lógicos nebulosos que utilizam uninormas (Pedrycz e Hirota, 2007). Uma das propostas de continuidade desse trabalho é o desenvolvimento de uma metodologia evolutiva de aprendizagem para esse modelo (mais detalhes no

capítulo 6). Finalmente, o último item descreve a variação da árvore de regressão linear nebulosa evolutiva capaz de realizar seleção de características, descrita no capítulo 4 seção 4.3.

#### 1.4.4 Trabalhos Submetidos

1. LEMOS, A. P. ; CAMINHAS, W. M. ; GOMIDE, Fernando. Online Fault Detection and Diagnosis Using an Evolving Fuzzy Classifier. Information Sciences.
2. SOUZA, L. ; LEMOS, A. P ; CAMINHAS, W. M. ; BOAVENTURA, W. Thermal Modeling of Power Transformers Using Evolving Fuzzy Systems. IEEE Transactions on Power Delivery.

O primeiro item descreve detalhadamente a metodologia de detecção e diagnóstico adaptativo de falhas proposta, assim como sua aplicação em um *benchmark* de detecção e diagnóstico de falhas. O segundo item descreve a aplicação do modelo funcional evolutivo proposto no capítulo 3 na modelagem térmica de transformadores.

# Capítulo 2

## Modelagem Nebulosa Evolutiva

### 2.1 Introdução

Esse capítulo realiza uma revisão dos principais modelos nebulosos evolutivos propostos na literatura. O objetivo dessa revisão não é detalhar todos modelos já propostos, mas sim descrever os princípios de funcionamento básicos de diferentes modelos, assim como algumas de suas deficiências, que serviram como base para o desenvolvimento dos modelos propostos nesse trabalho.

A ideia de sistemas flexíveis, capazes de adaptar sua estrutura automaticamente, surgiu inicialmente na década de 90 com alguns trabalhos aplicados a redes neurais (Fritzke, 1994; Kwok e Yeung, 1997; Williamson, 1996). Os primeiros sistemas nebulosos evolutivos propostos na literatura datam do início deste século (Angelov, 2002; Kasabov e Song, 2002) e foram desenvolvidos para suprir uma demanda de modelos flexíveis, adaptativos e interpretáveis, necessários para o desenvolvimento de sensores inteligentes, sistemas autônomos, etc. Sistemas nebulosos evolutivos possuem vantagens sobre outros modelos caixa preta, como redes neurais evolutivas, por serem linguisticamente interpretáveis, tornando possível a extração de informação de sua topologia na forma de grânulos de informação (termos linguísticos representados por conjuntos nebulosos) (Angelov, 2010).

A seção 2.2 revê os principais modelos nebulosos evolutivos propostos na literatura para resolver problemas de regressão. A seção 2.3 descreve os modelos propostos para problemas de classificação de padrões.

### 2.2 Modelos de Regressão Nebulosos Evolutivos

Há vários trabalhos na literatura que propõem modelos funcionais nebulosos evolutivos para resolver problemas de regressão, tais como, identificação de sistemas dinâmicos e previsão de séries temporais. A maior parte desses trabalhos (Angelov e Buswell, 2001; Angelov e Filev, 2004,

2005; Angelov, 2002; Kasabov e Song, 2002; Lima et al., 2009; Lughofer, 2008a) propõem modelos funcionais baseados em um conjunto de regras do tipo Takagi-Sugeno (Takagi e Sugeno, 1985) em que a estrutura do modelo (número de regras e parâmetros dos antecedentes) é flexível, sofrendo alterações baseadas em grupos criados por algoritmos de agrupamento recursivos não supervisionados, conforme ilustrado pela figura 2.1. Os parâmetros dos consequentes são atualizados utilizando-se mínimos quadrados recursivos ou suas variações (Ljung, 1999; Young, 1984).

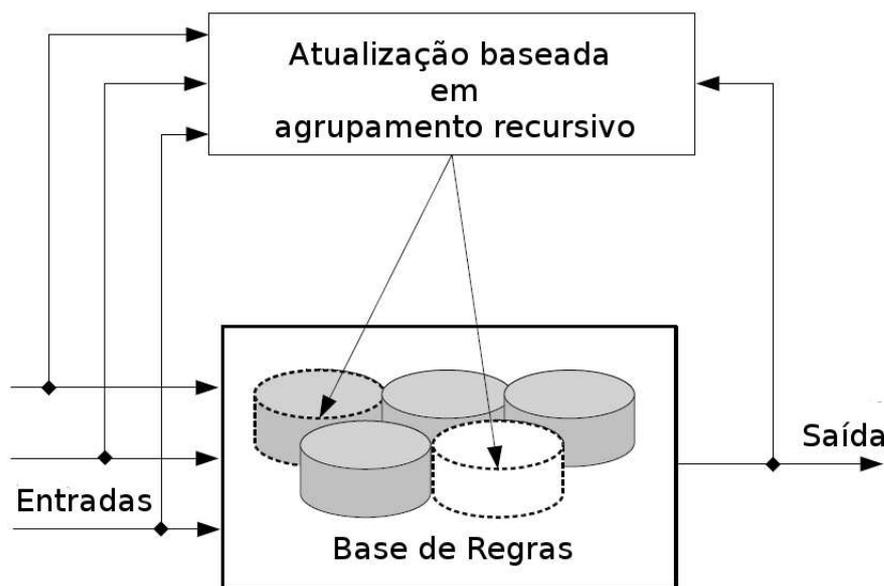


Figura 2.1: Mecanismo de atualização da base de regras

Para ilustrar o funcionamento básico de tais modelos, o modelo eTS (*evolving Takagi-Sugeno*) (Angelov e Filev, 2004) será detalhado. Esse modelo pode ser considerado como um dos trabalhos pioneiros na área de modelagem nebulosa evolutiva, proposto inicialmente em Angelov (2002) e detalhado em Angelov e Filev (2004).

### 2.2.1 Takagi-Sugeno Evolutivo (eTS)

O modelo eTS é formado por um conjunto de regras do tipo:

$$R_i : \text{Se } x_1 \text{ é } A_{i1} \text{ e } \dots \text{ e } x_m \text{ é } A_{im} \text{ então } y_i = a_{i0}x_1 + \dots + a_{im}x_m$$

onde  $R_i$  é a  $i$ -ésima regra nebulosa para  $i = 1, \dots, g^k$ ,  $g^k$  é o número de regras na iteração  $k$ ,  $x_j$  para  $j = 1, \dots, m$  são as  $m$  variáveis de entrada,  $A_{ij}$  são os conjuntos nebulosos do antecedente,

$y_i$  é a saída do modelo linear presente no conseqüente e  $a_{ij}$  os parâmetros do modelo linear da regra  $i$ .

Os conjuntos nebulosos dos antecedentes são definidos por funções de pertinência Gaussianas:

$$\mu_{ij} = \exp\left(-\frac{4}{r^2}\|x_j - x_{ij}^*\|^2\right) \quad (2.1)$$

sendo  $r$  um parâmetro do algoritmo (valor positivo) que define o espalhamento da função de pertinência e a zona de influência do modelo presente na regra  $i$ ,  $\|\cdot\|$  a norma Euclidiana e  $x_{ij}^*$  o ponto focal do conjunto nebuloso.

O grau de ativação de cada regra é definido como a conjunção, pela  $t$ -norma (Pedrycz e Gomide, 2007) produto, dos conjuntos nebulosos de cada regra:

$$\tau_i = \mu_{i1}(x_1) \times \mu_{i2}(x_2) \times \cdots \times \mu_{im}(x_m); \quad (2.2)$$

A saída do modelo é então definida como a média ponderada da saída de cada regra:

$$y = \sum_{i=1}^{g^k} \lambda_i y_i \quad (2.3)$$

sendo  $\lambda_i = \tau_i / \sum_{j=1}^{g^k} \tau_j$  o grau de ativação normalizado de cada regra e  $y_i$  a saída do modelo linear presente no conseqüente da regra  $i$ .

Um algoritmo de agrupamento recursivo não supervisionado é utilizado para atualizar a base de regras do modelo a cada nova iteração, isto é, a cada nova amostra apresentada. O agrupamento é realizado no espaço de entrada e saída  $z = [x^T \ y]^T$ . A cada iteração, os grupos existentes são projetados nos eixos das variáveis de entrada gerando os conjuntos nebulosos dos antecedentes das regras. A figura 2.2<sup>1</sup> ilustra a metodologia de definição dos conjuntos nebulosos através da projeção de grupos para um espaço de entrada de duas dimensões. Cada grupo define o antecedente de uma regra.

O algoritmo atualiza a estrutura de grupos a cada iteração de forma que cada amostra processada pode gerar um novo grupo ou simplesmente atualizar os parâmetros de um grupo existente.

Esse modelo utiliza uma versão incremental do algoritmo de agrupamento subtrativo (Chiu, 1994). O algoritmo de agrupamento utilizado baseia-se na noção de que a representatividade de cada amostra é medida a partir de uma função que estima seu *potencial* (Yager e Filev, 1994).

A função potencial de uma dada amostra  $z^k$  é definida como uma medida da proximidade da amostra a todas as outras (Angelov e Filev, 2004):

$$P(z^k) = \frac{1}{k-1} \sum_{i=1}^{k-1} \exp(-r\|z^k - z^i\|^2) \quad (2.4)$$

---

<sup>1</sup>Figura adaptada de Hell (2008)

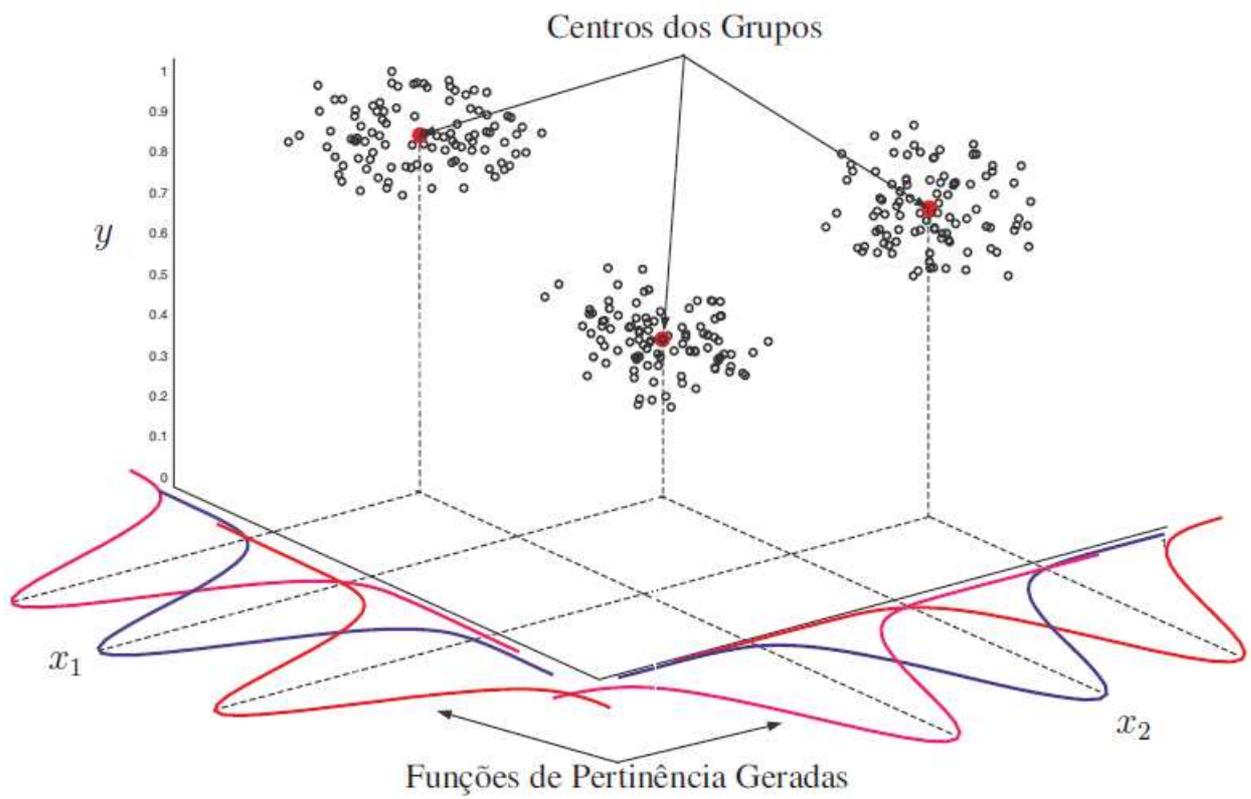


Figura 2.2: Exemplo de geração de funções de pertinência a partir de grupos

sendo  $k = 2, 3, \dots$  o índice das amostras já processadas.

A função potencial tenta encontrar amostras que possam ser definidas como centros de regiões que possuam concentrações de dados, conforme ilustrado pela figura 2.3 em que o potencial da amostra  $A$  é inferior ao potencial da amostra  $B$ . Quanto mais amostras estiverem concentradas em uma determinada região, maior será o potencial das amostras presentes nessa região.

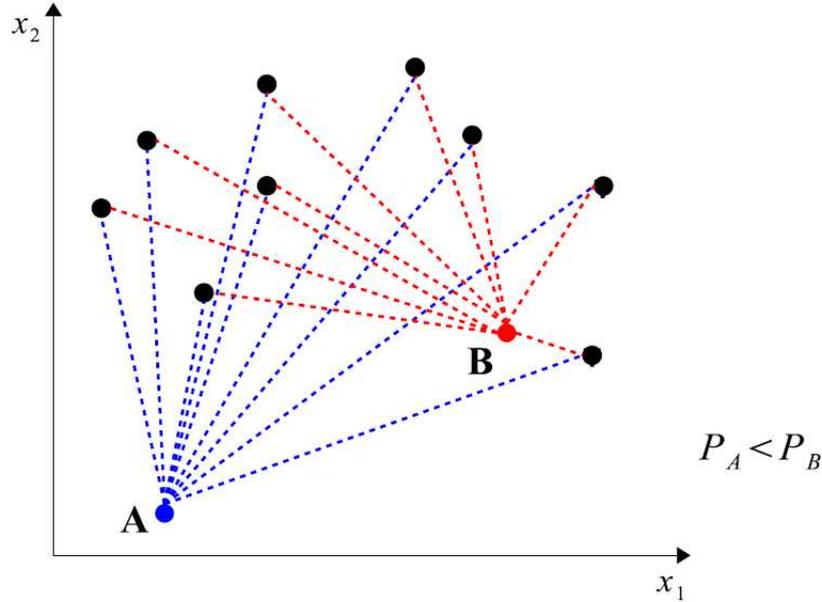


Figura 2.3: Ilustração da ideia do potencial

Angelov e Filev (2004) propõem uma forma de estimar o potencial de cada nova amostra recursivamente:

$$P^k(z^k) = \frac{k-1}{(k-1)(\vartheta^k + 1) + \sigma^k - 2\nu^k} \quad (2.5)$$

onde  $\vartheta^k = \sum_{j=1}^{m+1} (z_j^k)^2$ ,  $\sigma^k = \sum_{l=1}^{k-1} \sum_{j=1}^{m+1} (z_j^l)^2$  e  $\nu^k = \sum_{j=1}^{m+1} z_j^k \beta_j^k$  sendo  $\beta_j^k = \sum_{l=1}^{k-1} z_j^l$ .

Os parâmetros  $\vartheta^k$  e  $\nu^k$  são computados a partir de  $z^k$ , enquanto  $\beta_j^k$  e  $\sigma^k$  podem ser computados recursivamente (Angelov e Filev, 2004).

O potencial dos centros de todos os grupos existentes também são atualizados:

$$P^k(z_l^*) = \frac{(k-1)P^{k-1}(z_l^*)}{k-2 + P^{k-1}(z_l^*) + P^{k-1}(z_l^*) \sum_{j=1}^{m+1} d_j^{k(k-1)}} \quad (2.6)$$

sendo  $z_l^*$  o centro do grupo  $l$  (dimensão  $1 \times m + 1$ ) e  $d_j^{k(k-1)} = z_j^k - z_j^{k-1}$ .

Uma vez estimado o potencial da nova amostra e atualizados os potenciais dos centros dos grupos existentes, caso o potencial da nova amostra seja maior que o potencial de todos os grupos, a nova amostra é definida como o centro de um grupo. Caso a amostra esteja próxima o suficiente do centro de um grupo existente, essa amostra substituirá o centro desse grupo. Caso contrário, a amostra será definida como o centro de um novo grupo.

Caso o potencial da nova amostra não seja maior que o potencial do centro de todos os grupos, os parâmetros do consequente da regra relacionada ao grupo com menor distância Euclidiana são atualizados, utilizando o algoritmo recursivo de mínimos quadrados ou o algoritmo recursivo de mínimos quadrados ponderados (Ljung, 1999; Young, 1984).

O algoritmo 1 sumariza o procedimento de atualização da estrutura do modelo eTS.

---

**Algoritmo 1** Algoritmo de atualização da estrutura do modelo eTS

---

```

1: Computa o potencial da nova amostra  $P(z^k)$ 
2: for  $j = 1, \dots, g^k$  do
3:   Computa o potencial do centro  $c^j$ 
4: end for
5: if  $P(z^k) > P(c^j) \forall j$  then
6:   if  $z^k$  é próximo o suficiente do centro de algum grupo  $j$  then
7:      $z^k$  substitui  $c^j$  como centro do grupo  $j$ 
8:   else
9:     Um novo grupo é criado centrado em  $z^k$ 
10:  end if
11: else
12:   Atualiza os parâmetros do consequente do grupo mais próximo
13: end if

```

---

## 2.2.2 Outros Modelos Funcionais Evolutivos

Diversos outros modelos podem ser encontrados na literatura com topologia e aprendizado similar ao eTS. Esses modelos diferem, principalmente, pelo algoritmo de agrupamento recursivo utilizado para definir a base de regras a cada iteração. Além de apresentarem conceitos novos no processo de inferência e na atualização do conjunto de regras.

O DENFIS (*Dynamic Evolving Neural-Fuzzy Systems*) (Kasabov e Song, 2002) é um modelo similar ao eTS em que a base de regras é definida utilizando um algoritmo de agrupamento recursivo baseado na distância Euclidiana entre os pontos e os centros dos grupos, denominado

ECM (*Evolving Clustering Method*). Um novo grupo é criado caso a distância Euclidiana entre uma nova amostra e o centro de todos os grupos existentes seja maior que um determinado limiar de distância, definido como parâmetro do algoritmo. O agrupamento é realizado apenas no espaço de entrada e o centro de cada grupo é definido como o vetor médio estimado, ao contrário do eTS, em que o centro dos grupos é sempre definido como uma amostra. Esse modelo utiliza funções de pertinência triangulares e, por isso, utiliza uma versão modificada do processo de inferência Takagi-Sugeno. Para cada nova amostra, apenas as  $m$  regras mais próximas (distância estimada utilizando o centro do grupo relacionado) são selecionadas para estimar a saída correspondente.

Uma versão simplificada do eTS foi proposta em Angelov e Filev (2005) para reduzir o custo computacional no cálculo do potencial. O SimpLeTS substitui o potencial pelo conceito do *espalhamento*, que pode ser calculado mais facilmente, resultando em um algoritmo similar e mais efetivo do ponto de vista computacional. Além disso, esse modelo propõe uma metodologia para exclusão de regras inativas. Para isso, as regras existentes são monitoradas, e são excluídas do modelo caso o grupo relacionado à regra represente menos de 1% de todas as amostras já processadas. A representatividade de um grupo é expressa por sua *população*, que representa o número de amostras presentes na zona de influência do grupo. A população dos grupos é atualizada a cada iteração da seguinte forma:

$$POP^j = POP^j + 1; \quad j = \arg \min_j \|z^k - z_j^*\|^2 \quad (2.7)$$

Angelov e Zhou (2006) propõem uma versão estendida do eTS capaz de estimar o raio de atuação de cada grupo (parâmetro  $r$  do modelo eTS) de forma recursiva a partir das amostras. O xTS também utiliza a população para medir a qualidade dos grupos, porém nesse modelo a população de um grupo é denominada como seu *suporte*. Além disso, introduz um novo índice de qualidade dos grupos, denominado *idade*. A idade de um grupo é definida como o número de amostras já processadas menos o valor médio dos índices temporais das amostras representadas pelo grupo:

$$AGE^j = k - \frac{2A^j}{k+1} \quad (2.8)$$

sendo  $A^j = \sum_{i=1}^{POP^j} idx^j$  o tempo acumulado de chegada, e  $idx^j$  o índice temporal da  $j$ -ésima amostra. O tempo acumulado de chegada é calculado de forma similar à população do grupo (2.7), porém utiliza o índice temporal da amostra:

$$idx^j = idx^j + k; \quad j = \arg \min_j \|z^k - z_j^*\|^2 \quad (2.9)$$

Esse índice assume valores no intervalo  $(0, k]$ . Valores próximos de 0 indicam que o grupo é novo, enquanto valores próximos que  $k$  sugerem um grupo antigo, cujo centro deve ser substituído por novas amostras.

O FLEXFIS (*Flexible Fuzzy Inference System*), detalhado em Lughofer (2008a), utiliza um algoritmo de agrupamento derivado de uma modificação da técnica de quantização vetorial (Gray, 1984), denominada eVQ (*evolving Vector Quantization*). Assim como o DENFIS, esse

modelo também utiliza um limiar de distância para criar novos grupos. Porém, esse limiar é definido como função da dimensão do espaço de entrada. Isso é realizado para se evitar a *maldição da dimensionalidade*, ou seja, quanto maior a dimensão, maior a distância entre dois pontos adjacentes (Hastie et al., 2001). Assim, caso o limiar seja definido sem levar em conta a dimensão do espaço de entrada, quanto maior a dimensão, maior a probabilidade do algoritmo de aprendizagem gerar um número excessivo de grupos, o que pode causar sobreajuste. O limiar de distância desse modelo é definido como:

$$\rho = \text{fac} \frac{\sqrt{m+1}}{\sqrt{2}} \quad (2.10)$$

sendo  $m$  a dimensão do espaço de entrada e  $\text{fac}$  um parâmetro que deve ser ajustado para cada problema. Contudo, como ajustar o parâmetro  $\text{fac}$  ainda é uma questão em aberto (Lughofer, 2008a).

Esse método reconhece o problema da *maldição da dimensionalidade* e propõe uma solução. Porém essa solução depende de um parâmetro que deve ser ajustado manualmente, baseado em informação prévia sobre o problema a ser tratado.

O SOFMLS (*Self-Organizing Fuzzy Modified Least-Squares Network*) (Rubio, 2009) utiliza uma versão recursiva do algoritmo de agrupamento do vizinho mais próximo (Duda et al., 2000) para definir a estrutura do modelo. Além disso, esse modelo utiliza regras Takagi-Sugeno de ordem zero:

$$R_i : \text{Se } x_1 \text{ é } A_{i1} \text{ e } \dots \text{ e } x_m \text{ é } A_{im} \text{ então } y_i = v_i$$

sendo  $v_i$  uma constante.

A utilização desse tipo de regras permite representar a estrutura do modelo como uma rede *feedforward* de duas camadas: a camada de estimação do grau de ativação de cada regra e a camada de saída, conforme ilustrado pela figura 2.4. Esse algoritmo também possui uma metodologia de remoção de regras irrelevantes baseada na densidade de cada regra, similar ao xTS e SimpLeTS.

O SAFIS (*Sequential Adaptive Fuzzy Inference System*) também utiliza regras Takagi-Sugeno de ordem zero, podendo também ser representado por uma rede *feedforward*. Esse modelo utiliza um critério de geração de novas regras baseado na distância Euclidiana entre as amostras e os centros dos grupos, e também na influência de uma regra. A influência de uma regra é uma medida da contribuição de uma regra para a saída do modelo e é utilizada para criar e excluir regras.

Diversos outros modelos nebulosos evolutivos são propostos na literatura definidos com estrutura de rede, (Er e Wu, 2002; Kasabov e Lee, 2001; Leng et al., 2005). Esses modelos possuem uma estrutura multicamada sendo possível extrair regras funcionais a partir de sua estrutura. Algoritmos de treinamento são propostos de forma a adaptar os pesos da rede (que podem ser interpretados como parâmetros dos antecedentes e consequentes de regras funcionais), assim como o número de nós em determinadas camadas dessas redes (que pode ser interpretado como o número de regras).

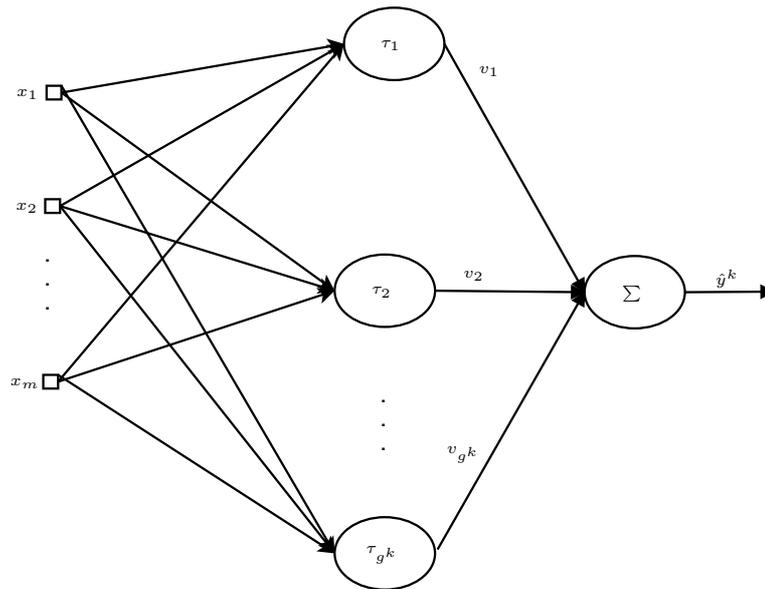


Figura 2.4: Topologia do modelo SOFMLS

O critério de geração de regras da maior parte das metodologias de treinamento de sistemas nebulosos evolutivos propostas (Er e Wu, 2002; Kasabov e Song, 2002; Leng et al., 2005; Lughofer, 2008a; Rubio, 2009) pode ser sumarizado em dois passos. Inicialmente, define-se uma medida de similaridade e um conjunto inicial de grupos (e seus respectivos centros) estimados a partir de um conjunto inicial de dados; ou define-se um único grupo centralizado na primeira amostra observada. Em seguida, para cada nova amostra, calcula-se sua medida de similaridade em relação aos grupos existentes. Caso essa medida ultrapasse um determinado limiar, um novo grupo é criado, caso contrário, o grupo de maior similaridade é atualizado utilizando um algoritmo recursivo de atualização de parâmetros. Embora seja efetivo, modelos gerados a partir desse critério tendem ser pouco robustos, uma vez que dados corrompidos por ruído ou *outliers* que excedam o limiar geram novos grupos, ao invés de serem rejeitados ou ter seu efeito suavizado.

Um modelo funcional evolutivo robusto foi proposto por Lima et al. (2009), denominado ePL (*evolving Participatory Learning*). Esse modelo possui uma topologia similar ao modelo eTS, porém baseia-se em um algoritmo de agrupamento recursivo inspirado no conceito do aprendizado participativo (Yager, 1990). O aprendizado participativo é um modelo de aprendizagem baseado na forma de aprendizagem humana, em que o processo de aprendizagem e o conhecimento a respeito de um sistema a ser modelado dependem do que já foi aprendido. Esse modelo de aprendizagem já foi utilizado para construção de redes neurais (Yager, 2004), modelos nebulosos híbridos (Hell et al., 2008), algoritmos de agrupamento (Silva, 2003; Silva et al., 2005), entre outras aplicações.

Uma característica essencial desse modelo de aprendizagem é que o impacto causado por uma nova amostra na revisão dos conhecimentos a respeito do sistema, depende do grau de

compatibilidade dessa amostra com os conhecimentos já adquiridos (Lima et al., 2009). Dessa maneira, algoritmos de agrupamento baseados nesse paradigma de aprendizagem tendem a ser mais robustos a ruído, uma vez que dados corrompidos por ruído *outliers* tendem a ser incompatíveis com o conhecimento corrente e, conseqüentemente, podem ser descartados ou ter seu efeito suavizado.

## 2.3 Classificadores Nebulosos Evolutivos

Apesar da grande maioria dos modelos nebulosos propostos na literatura serem propostos para problemas de regressão, classificadores nebulosos evolutivos também já foram propostos. Esses classificadores possuem um aprendizado supervisionado e diferem dos classificadores clássicos, principalmente por possuírem uma estrutura flexível, ou seja, uma nova classe pode ser definida durante o processo de aprendizagem. O classificador pode iniciar com uma única regra, descrevendo apenas uma classe, e adicionar regras para novas classes, baseando-se em um aprendizado supervisionado, ou seja, os dados de entrada do treinamento a cada iteração são definidos como pares de entrada e saída  $[x^k \ C_i^k]$ , sendo  $C_i^k$  a classe da amostra de entrada  $x^k$   $1 \times m$ .

Angelov e Zhou (2008) propõem dois modelos evolutivos para resolver problemas de classificação de padrões. O primeiro modelo, denominado eClass0, utiliza um conjunto de regras do tipo Takagi-Sugeno de ordem zero. Os conjuntos nebulosos presentes nos antecedentes das regras são definidos utilizando um algoritmo de agrupamento baseado no potencial das amostras, similar ao utilizado pelo modelo eTS. O conseqüente das regras é definido como uma das possíveis classes do problema:

$$R_i : \text{ Se } x_1 \text{ é } A_{i1} \text{ e } \dots \text{ e } x_m \text{ é } A_{im} \text{ então } y_i \text{ é } C_i$$

sendo  $C_i$  uma das possíveis classes do problema.

Mais de uma regra pode ser definida para uma classe, porém o número de regras deve ser no mínimo igual ao número de classes já apresentadas ao classificador.

O algoritmo atualiza o conjunto de regras a partir do algoritmo de agrupamento e os pares de entrada e saída. Cada novo grupo encontrado pelo algoritmo de agrupamento define uma nova regra. Os parâmetros do antecedente são extraídos do grupo correspondente e o conseqüente é definido a classe associada à entrada.

Para realizar a classificação de uma dada amostra, inicialmente os graus de ativação de todas as regras são estimados e a saída é definida como a classe presente no antecedente da regra mais ativa:

$$C = C_i ; i = \arg \max_i \tau_i \quad (2.11)$$

O segundo modelo proposto em Angelov e Zhou (2008) é denominado eClass1. Esse modelo estende o modelo eClass0 através de regras Takagi-Sugeno de primeira ordem de múltiplas

entradas e múltiplas saídas. Cada regra possui no conseqüente um modelo linear das entradas para cada possível classe do problema:

$$R_i : \text{ Se } x_1 \text{ é } A_{i1} \text{ e } \dots \text{ e } x_m \text{ é } A_{im} \text{ então } y_i = x^k \Theta$$

sendo que nesse caso,  $y_i$  é um vetor de dimensão igual ao número de classes já apresentadas ao classificador, e  $\Theta$  é uma matriz de dimensão  $m + 1$  vezes o número de classes.

A saída dos modelos lineares são normalizadas no intervalo unitário. Ao contrário do eClass0 em que cada regra está associada a uma classe, no eClass1 cada regra é capaz de estimar o grau de pertinência da entrada para cada uma das possíveis classes do problema.

A saída do classificador é definida como um vetor de dimensão igual ao número de classes. Cada elemento do vetor representa o grau de pertinência da entrada para cada uma das classes. Cada elemento do vetor é definido como a soma ponderada das saídas dos modelos lineares presentes nas regras associados à classe em questão.

Angelov et al. (2008b) propõem classificadores similares aos eClass0 e eClass1 utilizando o algoritmo de agrupamento proposto pelo modelo FLEXFIS em (Lughofer et al., 2007). Esses classificadores são denominados FLEXFIS-Class.

## Capítulo 3

# Modelagem Baseada no Agrupamento Evolutivo Participativo Gaussiano

### 3.1 Introdução

Esse capítulo propõe uma técnica de modelagem nebulosa evolutiva baseada em um algoritmo de agrupamento recursivo não supervisionado, denominado *agrupamento evolutivo participativo Gaussiano*. Os modelos propostos nesse capítulo se assemelham à maior parte dos modelos nebulosos evolutivos propostos na literatura quanto à metodologia de definição de sua topologia. Esses modelos utilizam informação sobre a organização espacial das variáveis de entrada para definir um conjunto de regras de forma adaptativa, dado um fluxo de dados.

O algoritmo de agrupamento proposto é baseado no conceito do aprendizado participativo (Yager, 1990) e pode ser visto como uma extensão do algoritmo proposto por Silva et al. (2005). Porém, diferentemente de Silva et al. (2005), o algoritmo proposto assume que cada grupo pode ser modelado por uma função de pertinência Gaussiana multivariável, a estrutura de grupos (número de grupos, centro e formato de cada grupo) é atualizada recursivamente a cada iteração e os limiares são definidos automaticamente. Esse algoritmo, descrito na seção 3.2, é utilizado para gerar, atualizar e remover regras nebulosas de forma incremental nos modelos nebulosos evolutivos propostos nesse capítulo.

O algoritmo de agrupamento proposto considera a possibilidade de que as entradas possam interagir entre si. Grupos são estimados baseados em uma medida normalizada de distância (similar à distância de *Mahalanobis* (Duda et al., 2000)), gerando grupos elipsoidais cujos eixos não são necessariamente paralelos aos eixos das variáveis de entrada, o que ocorreria caso a distância Euclidiana fosse utilizada (Angelov e Filev, 2004; Kasabov e Song, 2002; Lughofer, 2008a). Essa medida de distância é utilizada para preservar as possíveis interações entre as variáveis de entrada. Assim, os grupos são representados por funções de pertinência Gaussianas multivariáveis caracterizadas por um vetor central e uma matriz de dispersão, que representa

a dispersão presente em cada variável, assim como as iterações entre estas.

A maior parte dos algoritmos de agrupamento evolutivos propostos na literatura criam novos grupos sempre que uma dada medida de distância excede um determinado limiar (Kasabov e Song, 2002; Lughofer, 2008a; Rubio, 2009). Conforme discutido anteriormente, para se evitar a *maldição da dimensionalidade*, o valor do limiar deve ser definido considerando a dimensão do espaço de entrada, uma vez que quanto maior a dimensão, maior a distância entre dois pontos adjacentes. Caso o valor do limiar seja definido sem se utilizar informação relativa à dimensão, à medida que a dimensão aumenta, mais observações tendem a corresponder à distâncias que excedam o limiar, acarretando no aumento do número de grupos criados e no aumento da complexidade do modelo, o que pode levar ao problema de sobreajuste. O algoritmo proposto evita a *maldição da dimensionalidade* através de um mecanismo automático, capaz de ajustar o valor do limiar de distância baseado na dimensão do espaço de entrada.

Todos os modelos nebulosos evolutivos propostos nesse capítulo são formados por regras com antecedentes definidos por conjuntos nebulosos, com funções de pertinência Gaussianas multivariáveis, com parâmetros extraídos dos grupos encontrados pelo algoritmo de agrupamento proposto. Assim, a seção 3.3 descreve a metodologia de geração dos antecedentes das regras nebulosas a partir dos grupos, e detalha as vantagens dessa abordagem em comparação com metodologias clássicas, que geram os antecedentes através da projeção de grupos encontrados por algoritmos de agrupamento nos eixos das variáveis de entrada.

Finalmente, dois modelos nebulosos evolutivos são propostos. Na seção 3.4 um modelo funcional evolutivo é proposto. Esse modelo é formado por um conjunto de regras Takagi-Sugeno de primeira ordem e pode ser utilizado para identificação de sistemas e previsão de séries temporais. Esse modelo é denominado eMG (*evolving Multivariable Gaussian*), inicialmente proposto em Lemos et al. (2011c). Em seguida, na seção 3.5, um modelo linguístico evolutivo é descrito para tratar o problema de detecção e diagnóstico adaptativo de falhas em sistemas dinâmicos, inicialmente proposto em Lemos et al. (2010c).

## 3.2 Agrupamento Evolutivo Participativo Gaussiano

O algoritmo de agrupamento evolutivo proposto nesse trabalho é baseado no conceito do aprendizado participativo (Yager, 1990). O aprendizado participativo é um modelo de aprendizagem que assume que o processo de atualização do conhecimento adquirido sobre um sistema depende do que já foi aprendido anteriormente. Ou seja, o conhecimento adquirido sobre o sistema faz parte do processo de aprendizagem e influencia na forma como novas observações são utilizadas neste processo.

A característica essencial desse modelo de aprendizagem consiste no fato que o impacto causado por uma nova observação no aprendizado ou na revisão do conhecimento existente depende de sua compatibilidade com o que já foi aprendido. Assim, algoritmos de agrupamento baseados nesse modelo de aprendizagem tendem a ser robustos, uma vez que amostras corrompidas por ruído ou *outliers* tendem a ser incompatíveis com o conhecimento adquirido previamente e, conseqüentemente, tendem a ser descartados ou ter seu efeito suavizado. O algoritmo proposto nesse trabalho provê um mecanismo automático para decidir se uma nova

observação incompatível com o conhecimento atual (representado por uma estrutura de grupos) é o primeiro representante de um novo grupo ou apenas um *outlier*, que deve ser descartado ou ter seu efeito suavizado.

O mecanismo de suavização de *outliers* fornecido pelo aprendizado participativo também introduz uma forma de estabilidade no processo de aprendizagem. Modelos baseados no aprendizado participativo têm a característica de proteger os conhecimentos adquiridos do sistema modelado de amplas oscilações das entradas causadas por observações anômalas ou errôneas, e ainda permitir o aprendizado de novas informações (Yager, 2007).

O algoritmo de agrupamento proposto assume que o conhecimento adquirido sobre o sistema a ser modelado a cada iteração  $k$  é descrito por uma estrutura de grupos, isto é, o número de grupos  $g^k$ , seus respectivos centros  $c_i^k$  para  $i = 1, \dots, g^k$  e a forma de cada grupo, representada por uma matriz de dispersão  $\Sigma_i^k$ . A cada iteração, o processo de aprendizagem pode criar um novo grupo, modificar os parâmetros de um grupo existente, ou unir dois grupos similares.

Para cada nova amostra apresentada ao algoritmo de aprendizado, a estrutura de grupos é atualizada a partir de uma medida de compatibilidade  $\rho_i^k \in [0, 1]$ , e um índice de alerta  $a_i^k \in [0, 1]$  calculados para cada um dos  $g^k$  grupos. A medida de compatibilidade representa o quanto a nova amostra é compatível com cada um dos grupos que compõe a estrutura, enquanto o índice de alerta é o resultado de um mecanismo de alerta que age como um crítico, alertando quando o conhecimento existente deve ser revisado, isto é, quando a estrutura de grupos atual não descreve adequadamente o sistema.

Limiares são definidos para a medida de compatibilidade ( $T_\rho$ ) e para o índice de alerta ( $T_a$ ) e, a cada iteração, se a medida de compatibilidade da amostra atual for menor que seu respectivo limiar para todos os grupos, isto é, se  $\rho_i^k < T_\rho \forall i = 1, \dots, g^k$ , e o índice de alerta do grupo com o maior grau de compatibilidade for maior que seu respectivo limiar, ou seja,  $a_i^k > T_a$  para  $i = \arg \max_i \rho_i^k$ , um novo grupo é criado. Caso contrário, o centro do grupo com o maior grau de compatibilidade é ajustado da seguinte maneira:

$$c_i^{k+1} = c_i^k + G_i^k(x^k - c_i^k) \quad (3.1)$$

onde  $x^k$  é o vetor de entrada da iteração  $k$  e  $G_i^k$  é definido como:

$$G_i^k = \beta(\rho_i^k)^{1-a_i^k} \quad (3.2)$$

e  $\beta \in [0, 1]$  é a taxa de aprendizado, definida como um dos parâmetros do algoritmo.

De acordo com Yager (1990), a medida de compatibilidade  $\rho_i^k$  é definida como uma função que mede a compatibilidade entre o conhecimento atual adquirido sobre o sistema, representado por cada um dos centros dos grupos, e a observação atual:

$$\rho_i^k = F(x^k, c_i^k) \quad (3.3)$$

A função  $F(x^k, c_i^k) \in [0, 1]$  deve retornar valores próximos de 0 à medida que as observações se tornem incompatíveis com o conhecimento atual, ou seja, os centros dos grupos, e retor-

nar valores próximos de 1 à medida que as observações estejam em completo acordo com o conhecimento atual. Por exemplo, caso  $x^k$  seja igual ao centro do grupo, então  $F(x^k, c_i^k) = 1$ .

Caso  $a_i^k = 0$ , então  $G_i^k = \beta \rho_i^k$ , logo o processo de aprendizagem não possui mecanismo de alerta e a taxa de aprendizagem é controlada apenas pela medida de compatibilidade. Observações com  $\rho_i^k = 0$  não fornecem nenhuma informação sobre o sistema, já que  $c_i^{k+1} = c_i^k$ , enquanto observações com  $\rho_i^k$  próximo a 1 fornecem muita informação.

A taxa de aprendizado  $\beta$  é modulada pelo índice de compatibilidade  $\rho$ . Quando não há considerações participativas no processo de aprendizagem,  $\beta$  é normalmente definido como um valor pequeno, para evitar grandes oscilações causadas por valores espúrios das entradas que se encontram distantes dos grupos correntes. Pequenos valores de  $\beta$  protegem o processo de aprendizagem contra a influência de entradas anômalas, mas também podem desacelerá-lo. A introdução do termo participativo  $\rho$  permite o uso de maiores valores para  $\beta$ . A taxa de aprendizado do modelo de aprendizado participativo é dinâmica e  $\rho$  age de forma a reduzir a taxa de aprendizado quando altas variações ocorrem. Porém, quando a compatibilidade é alta, o valor de  $\rho$  é definido de forma a acelerar o processo de aprendizagem.

O índice de alerta  $a_i^k$  é resultado de um mecanismo utilizado para medir a confiabilidade do conhecimento já adquirido sobre o sistema a ser modelado. Por exemplo, enquanto um único valor baixo da medida de compatibilidade causa aversão à aprendizagem, uma sequência de valores baixos deve implicar em uma revisão do conhecimento, uma vez que esta sequência de baixos valores pode ser interpretada como um forte indício de que o conhecimento adquirido sobre o sistema não é adequado para descrevê-lo corretamente.

O mecanismo de alerta é definido como um mecanismo de monitoramento da dinâmica da medida de compatibilidade. Esse mecanismo monitora os valores da medida de compatibilidade e seu resultado é interpretado como o complemento da confiabilidade do conhecimento atual. Um baixo valor do índice de alerta  $a_i^k$  implica em uma alta confiabilidade sobre o conhecimento adquirido, enquanto altos valores indicam a necessidade de revisão deste conhecimento. Analisando a equação (3.2), percebe-se que, à medida que o índice de alerta aumenta, a medida de compatibilidade tem seu efeito reduzido. Assim, se uma sequência de amostras apresenta baixos valores de compatibilidade, é mais provável que o conhecimento adquirido esteja de alguma forma incorreto e deva ser revisado. Quando isso ocorre, o valor de compatibilidade é reduzido, e novas observações são capazes de prover mais informação sobre o sistema, se comparado com a informação que seria provida caso o mecanismo de alerta não fosse utilizado. O caso extremo ocorre quando o índice de alerta excede seu respectivo limiar e um novo grupo é criado, conforme será detalhado posteriormente.

A figura 3.1 ilustra o aprendizado participativo, incluindo o conceito da utilização dos conhecimentos adquiridos no processo de aprendizagem e o mecanismo de monitoramento, expresso pelo índice de alerta.

A medida de compatibilidade  $\rho_i^k$  proposta nesse trabalho utiliza o quadrado de uma medida de distância normalizada entre a observação atual e os centros dos grupos (*distância M*):

$$M(x^k, c_i^k) = (x^k - c_i^k)(\Sigma_i^k)^{-1}(x^k - c_i^k)^T \quad (3.4)$$

Para calcular a *distância M*, a matriz de dispersão de cada grupo  $\Sigma_i^k$  deve ser estimada a cada iteração. A estimação recursiva da matriz de dispersão é dada por:

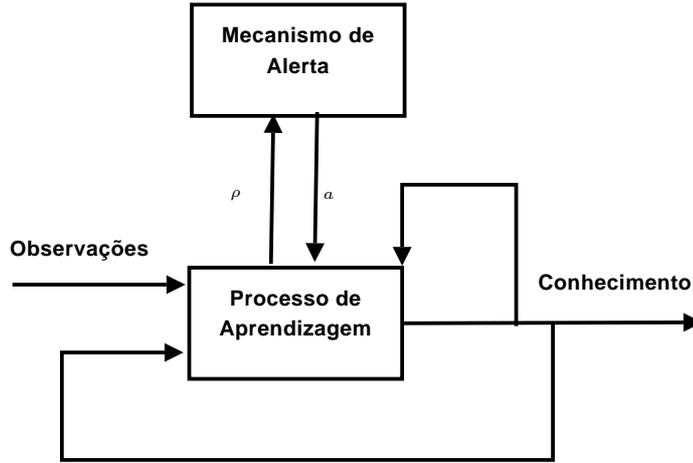


Figura 3.1: Aprendizado participativo

$$\Sigma_i^{k+1} = (1 - G_i^k)(\Sigma_i^k - G_i^k(x^k - c_i^k)(x^k - c_i^k)^T) \quad (3.5)$$

A medida de compatibilidade de cada grupo é então definida como:

$$\rho_i^k = F(x^k, c_i^k) = \exp \left[ -\frac{1}{2} M(x^k, c_i^k) \right] \quad (3.6)$$

A figura 3.2 ilustra a medida de compatibilidade entre uma amostra  $x$  e o centro de um grupo  $c$ , ambos com dimensão unitária. O grupo está centrado no ponto  $c = 0$  e possui dispersão igual a 0.1. O valor da amostra é variado no intervalo  $[-1, 1]$ . Analisando essa figura, percebe-se que a compatibilidade aumenta à medida que a amostra se aproxima do centro do grupo, atingindo o valor máximo (igual a 1), quando  $x = c$ . A compatibilidade tende a 0 à medida que  $x$  se afasta de  $c$ , ou seja, à medida que as observações se tornam incompatíveis com o grupo.

Para encontrar um limiar da medida de compatibilidade, definido como limiar de compatibilidade, assume-se que  $M(x^k, c_i^k)$  é modelada por uma distribuição *Qui-Quadrado* (Papoulis, 1984). Assim, dado um nível de significância  $\alpha$ , o limiar de compatibilidade pode ser calculado a partir da seguinte equação:

$$T_\rho = \exp \left[ -\frac{1}{2} \chi_{m,\alpha}^2 \right] \quad (3.7)$$

onde  $\chi_{m,\alpha}^2$  é o  $\alpha$  intervalo de confiança unilateral superior de uma distribuição *Qui-Quadrado* com  $m$  graus de liberdade, e  $m$  é a dimensão do espaço de entrada.

A medida de compatibilidade é baseada em uma medida de distância normalizada (3.4). Conforme discutido anteriormente, o limiar de compatibilidade (3.7) deve ser definido considerando-se a dimensão do espaço de entrada, de forma a evitar a *maldição da dimensionalidade*. Isto porque a distância entre dois pontos adjacentes cresce de acordo com a dimensão do espaço de

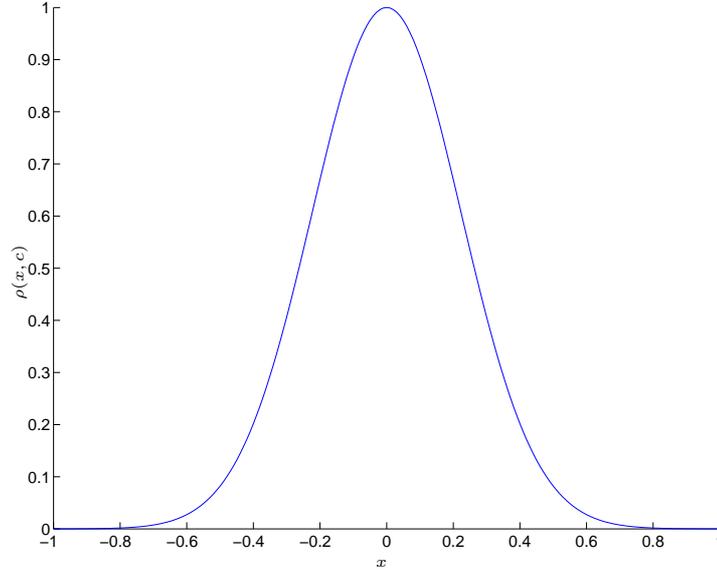


Figura 3.2: Exemplo unidimensional da medida de compatibilidade

entrada. Se um limiar fixo, que não dependa da dimensão do espaço de entrada, for adotado, o número de violações desse limiar aumentará em função do aumento da dimensão, o que pode levar à criação excessiva de grupos. Analisando a expressão (3.7), percebe-se que o limiar de compatibilidade inclui informação sobre a dimensão do espaço de entrada, uma vez que  $\chi_{m,\alpha}^2$  é função do número  $m$  de entradas. A figura 3.3 ilustra o valor de  $T_\rho$  em função da dimensão do espaço de entrada para  $\alpha = 0.05$ . Sumarizando, a metodologia de agrupamento proposta possui um mecanismo automático para ajustar o limiar de compatibilidade em função da dimensão do espaço de entrada. A dimensão é diretamente proporcional à distância entre dois pontos adjacentes e inversamente proporcional à medida de compatibilidade. Assim, à medida que a dimensão aumenta, o limiar de compatibilidade diminui, evitando violações excessivas desse limiar.

Com o intuito de demonstrar empiricamente a eficácia do ajuste automático de  $T_\rho$ , a tabela 3.1 sumariza os resultados de um experimento numérico realizado para computar o número médio de violações do limiar em função do crescimento do número de entradas. Conjuntos de dados foram gerados aleatoriamente, utilizando distribuições Gaussianas para diferentes dimensões  $m \in [1, 100]$ . Para cada conjunto de dados, foi definido um grupo, centrado no vetor médio e com a matriz de dispersão igual à matriz de covariância utilizada para gerar os dados. A medida de compatibilidade foi computada entre cada amostra e o centro do grupo. Em seguida, contou-se o número de violações do limiar de compatibilidade. Uma vez que o limiar de compatibilidade é baseado em um intervalo de confiança unilateral construído a partir de um nível de significância  $\alpha$ , o número esperado de violações é  $n \times \alpha$ . Foram gerados 100 conjuntos de dados com  $n = 1000$  amostras para cada uma das  $m$  dimensões. O teste  $t$

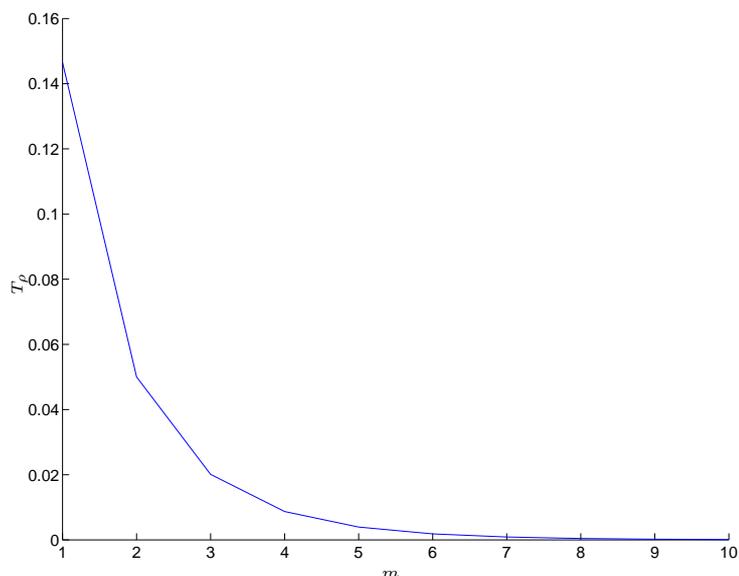


Figura 3.3: Limiar de compatibilidade em função da dimensão do espaço de entrada

(Papoulis, 1984) foi então utilizado para avaliar se as diferenças entre os valores esperados e os valores computados do número de violações eram estatisticamente significantes. A hipótese nula desse teste assume que a diferença entre os valores computados e esperados são amostras aleatórias de uma distribuição normal, com média zero e variância desconhecida, isto é, não existe diferença estatisticamente significativa entre o valor esperado e o computado do número de violações do limiar. O valor esperado de violações é igual a 50 para todos os conjuntos de dados, para  $\alpha = 0.05$  e  $n = 1000$ . A tabela 3.1 detalha o número médio de violações do limiar para cada valor de  $m$  avaliado. Além disso, são apresentados os p-valores da estatística de teste. Baixos p-valores indicam diferenças estatisticamente significantes entre os valores esperados e computados. Analisando a tabela 3.1 percebe-se que, à medida que a dimensão cresce, não há diferença estatisticamente significativa entre o número computado e estimado de violações do limiar de compatibilidade, para um nível de significância do teste igual a 0.01 (todos os p-valores são maiores que 0.01). Esses resultados comprovam, experimentalmente, a eficácia do limiar de compatibilidade proposto em prevenir a *maldição da dimensionalidade*.

Esse trabalho adota um mecanismo de alerta para monitorar a dinâmica da medida de compatibilidade baseado uma janela deslizante construída a partir dos seus últimos  $w$  valores. Dessa forma, o índice de alerta é definido como a probabilidade de observar menos de  $v$  violações do limiar de compatibilidade em uma sequência de  $w$  observações. Baixos valores do índice de alerta estão associados a nenhuma ou poucas violações do limiar de compatibilidade, implicando em uma alta confiabilidade sobre a estrutura de grupos. Altos valores do índice de alerta estão associados a várias violações do limiar, implicando que o conjunto de grupos deve ser revisto.

Para calcular o índice de alerta, para cada nova observação, calcula-se o valor de *ocorrência*

Tabela 3.1: Número Médio de Violações do Limiar de Compatibilidade em Função do Crescimento da Dimensão dos Dados

Dimensão dos Dados	# médio de violações observado	p-valor
1	48.96	0.13
5	49.00	0.19
10	49.92	0.91
15	49.40	0.38
25	51.09	0.09
50	49.55	0.55
100	50.00	1.00

$o^k$  correspondente, definido como:

$$o^k = \begin{cases} 0 & \text{para } M(x^k, c_i^k) < \chi_{m,\alpha}^2 \\ 1 & \text{caso contrário} \end{cases} \quad (3.8)$$

Nota-se que um valor de ocorrência  $o^k = 1$  indica uma violação do limiar de compatibilidade.

O valor de ocorrência  $o^k$  pode ser considerado também como o resultado de um teste estatístico utilizado para avaliar se os valores de  $M(x^k, c_i^k)$  são normais. A hipótese nula deste teste assume que  $M(x^k, c_i^k)$  pode ser modelado por uma distribuição *Qui-Quadrado* com  $m$  graus de liberdade. Sendo a hipótese nula verdadeira, a probabilidade de se observar  $o^k = 1$  é igual a  $\alpha$ , uma vez que  $\alpha$  define o intervalo  $\chi_{m,\alpha}^2$  e é a probabilidade de se observar um falso positivo, isto é,  $M(x^k, c_i^k) > \chi_{m,\alpha}^2$ .

Uma vez que  $o^k$  é uma variável binária e a probabilidade de se observar  $o^k = 1$  é conhecida, a variável aleatória associada a  $o^k$  pode ser modelada por uma distribuição de *Bernoulli* (Papoulis, 1984) com uma probabilidade de sucesso  $\alpha$ .

Dada uma sequência das últimas  $w$  observações, o número de violações do limiar, isto é,  $o^k = 1$ , é dado por:

$$v^k = \begin{cases} \sum_{j=0}^{w-1} o_i^{k-j} & k > w \\ 0 & \text{caso contrário} \end{cases} \quad (3.9)$$

Nota-se que  $v^K$  não é calculado nos primeiros  $w$  passos. Isso significa que o algoritmo possui uma latência inicial de  $w$  passos. Porém, essa latência não causa nenhum problema, uma vez que  $w$  é geralmente muito menor que o número de passos em que o aprendizado ocorre. Por exemplo, em aplicações de tempo real, o aprendizado pode ocorrer continuamente.

A distribuição de probabilidade discreta utilizada para modelar o número de violações do limiar em uma janela de  $w$  observações é  $p(V^k = v)$ , com  $V^k$  assumindo valores  $v = 0, 1, \dots, w$ .

Uma vez que  $V^k$  é definida como a soma de uma sequência de variáveis aleatórias independentes e identicamente distribuídas geradas por uma distribuição de *Bernoulli*, com a mesma probabilidade de sucesso  $\alpha$ ,  $p(V^k = v)$  pode ser descrita por uma distribuição *Binomial* (Papoulis, 1984):

$$p(V^k = v) = \begin{cases} \binom{w}{v} \alpha^v (1 - \alpha)^{w-v} & v = 0, \dots, w \\ 0 & \text{caso contrário} \end{cases} \quad (3.10)$$

A distribuição *Binomial* descrita acima fornece a probabilidade de se observar  $v$  violações do limiar de compatibilidade em uma sequência de  $w$  observações. Altos valores de probabilidade evidenciam a hipótese de que as observações podem ser descritas pela estrutura de grupos atual, enquanto baixos valores sugerem que as observações devem ser descritas por um novo grupo.

Por exemplo, a figura 3.4 ilustra os valores de  $p(V^k = v)$  para  $\alpha = 0.1$  e  $w = 50$ , isto é, a probabilidade de se observar 0 a 50 violações do limiar de compatibilidade em um sequência de 50 observações, dado  $T_\rho = \exp[-1/2\chi_{2,0.1}^2]$ . Analisando essa figura, nota-se que existe uma probabilidade significativa de se observar 10 ou menos violações e uma baixa probabilidade de se observar 20 ou mais violações.

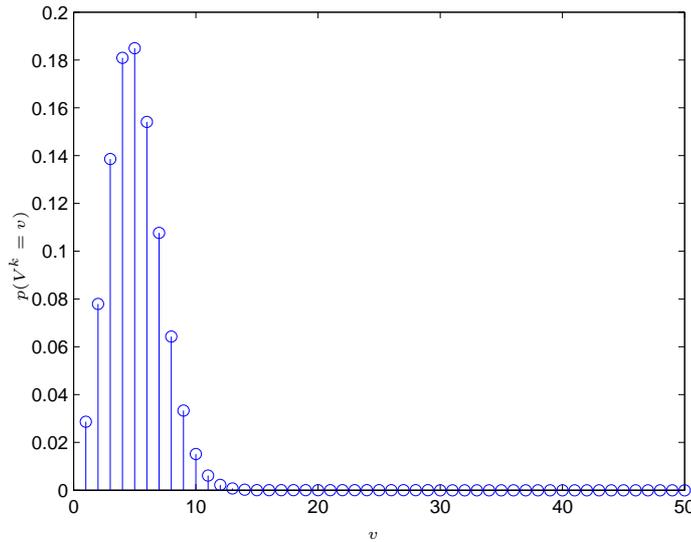


Figura 3.4: Probabilidade de se observar  $v$  violações do limiar de compatibilidade em uma sequência de 50 observações dado  $\alpha = 0.1$

O índice de alerta é definido como a probabilidade cumulativa de  $V^k$ , isto é,  $a_i^k = p(V^k < v)$ . A figura 3.5 ilustra os valores do índice de alerta em função do número de violações do limiar de compatibilidade para  $\alpha = 0.1$  e  $w = 50$ . Analisando essa figura, percebe-se que  $a_i^k \approx 1$  para 12 ou mais violações.

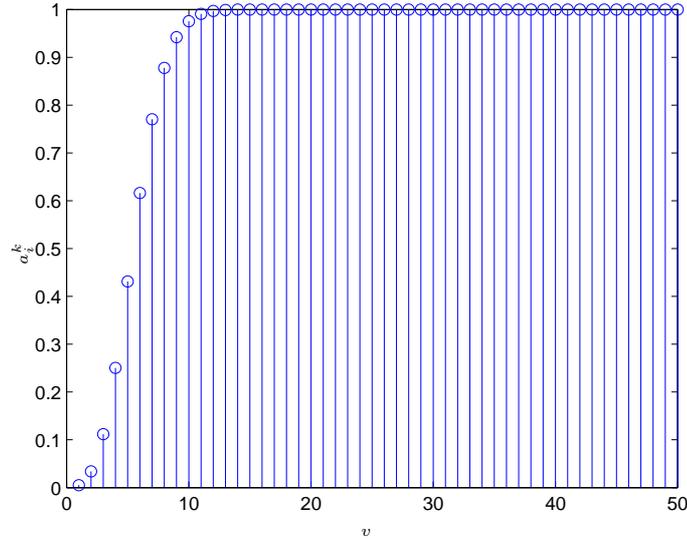


Figura 3.5: Índice de alerta, definido como a probabilidade de ser observar menos de  $v$  violações do limiar em uma sequência de  $w = 50$  observações dado  $\alpha = 0.1$

O limiar do índice de alerta, ou limiar de alerta, utilizado para criar novos grupos, pode então ser definido como o  $1 - \alpha$  (o mesmo  $\alpha$  utilizado para calcular o limiar de compatibilidade). No entanto, considerando-se que cada observação é utilizada em  $w$  janelas para o cálculo do índice de alerta, uma correção de comparações múltiplas deve ser aplicada, com o objetivo de se evitar a criação espúria de grupos, uma vez que o teste estatístico (avaliação do limiar) é aplicado  $w$  vezes para cada observação. Assim, a *correção de Bonferroni* (Miller, 1966) é aplicada, dividindo-se o nível de significância desejado pelo tamanho da janela. Então, dado um nível de significância  $\alpha$ , o limiar de alerta é definido como:

$$T_a = 1 - \frac{\alpha}{w} \quad (3.11)$$

e um novo grupo é criado quando a medida de compatibilidade é menor que o limiar de compatibilidade para todos os grupos, e o índice de alerta é maior que o limiar de alerta para o grupo com o maior valor de compatibilidade.

O valor mínimo de violações do limiar de compatibilidade em uma janela de tamanho  $w$ , necessário para exceder  $T_a$ , pode ser calculado numericamente procurando-se pelo primeiro valor de  $v$  em que sua distribuição cumulativa discreta é igual ou menor que  $1 - \frac{\alpha}{w}$ . Mais formalmente:

$$v^* = \arg \min_v \left| \sum_{l=1}^v \binom{w}{l} \alpha^l (1 - \alpha)^{w-l} - \left(1 - \frac{\alpha}{w}\right) \right| \quad (3.12)$$

Conforme descrito anteriormente, o algoritmo de agrupamento proposto revisa continuamente os grupos existentes e eventualmente une dois grupos similares. Para isso, a compatibilidade entre o grupo atualizado ou criado e todos os outros grupos remanescentes é calculada a cada iteração. Se, para um dado par, a compatibilidade é maior que o limiar de compatibilidade, os dois grupos são unidos, isto é, se  $\rho_i^k(c_j^k, c_i^k) > T_\rho$  ou  $\rho_i^k(c_i^k, c_j^k) > T_\rho$ , então os grupos são unidos, formando um único grupo.

A compatibilidade entre dois grupos  $i$  e  $j$  é definida como:

$$\rho_i^k(c_j^k, c_i^k) = \exp \left[ -\frac{1}{2} M(c_j^k, c_i^k) \right] \quad (3.13)$$

onde  $M(c_j^k, c_i^k)$  é a *Distância-M* entre os centros  $i$  e  $j$ , ou seja:

$$M(c_j^k, c_i^k) = (c_j^k - c_i^k)(\Sigma_i^k)^{-1}(c_j^k - c_i^k)^T \quad (3.14)$$

Nota-se que, para checar se dois grupos são similares, é necessário computar  $\rho(c_j^k, c_i^k)$  e  $\rho(c_i^k, c_j^k)$ , sendo que, normalmente,  $\Sigma_i^k \neq \Sigma_j^k$ .

O procedimento de agrupamento descrito pode iniciar a partir de uma única observação, ou a partir de um conjunto de observações. Caso um conjunto de observações esteja disponível, um algoritmo de agrupamento em batelada deve ser utilizado, para estimar a estrutura inicial de grupos. Este algoritmo deve ser capaz de estimar os centros dos grupos, assim como suas respectivas matrizes de dispersão. Caso o procedimento de agrupamento inicie a partir de uma única observação, essa observação é definida como o centro do único grupo inicial, e uma matriz de dispersão inicial  $\Sigma_{init}$  deve ser definida como parâmetro.

Durante o processo de agrupamento, caso um novo grupo seja criado, seu centro é definido como a observação atual, e sua matriz de dispersão como  $\Sigma_{init}$ .

Se dois grupos são unidos, o centro e a matriz de dispersão do grupo resultante são definidos como o valor médio dos centros e  $\Sigma_{init}$ , respectivamente.

O algoritmo de agrupamento proposto possui quatro parâmetros:

- a taxa de aprendizado  $\beta$ , utilizada para atualizar os parâmetros de cada grupo;
- o tamanho da janela,  $w$  usada pelo mecanismo de alerta para monitorar a dinâmica da medida de compatibilidade;
- o nível de significância  $\alpha$ , usado para calcular os limiares de compatibilidade e alerta.
- a matriz de dispersão inicial  $\Sigma_{init}$ , utilizada na criação de novos grupos.

A taxa de aprendizado é normalmente ajustada para um valor baixo, por exemplo,  $\beta \in [10^{-1}, 10^{-5}]$ .

O tamanho da janela  $w$  é dependente do problema, já que este parâmetro define quantas observações consecutivas devem ser consideradas para calcular o índice de alerta. Em outras palavras, considerando o conhecimento adquirido, este parâmetro define o tamanho do padrão anômalo utilizado para classificar cada nova observação como um *outlier*, ou o primeiro membro de um novo grupo.

O valor do nível de significância  $\alpha$  depende do tamanho da janela  $w$ . Esse parâmetro deve ser ajustado de forma que o limiar de alerta  $T_a$  corresponda a mais de uma violação do limiar de compatibilidade, isto é,  $v > 1$  quando  $a_i^k > T_a$ . Valores sugeridos para  $\alpha$  são:

$$\alpha \geq \begin{cases} 0.01, & \text{se } w \geq 100 \\ 0.05, & \text{se } 20 \leq w < 100 \\ 0.1, & \text{se } 10 \leq w < 20 \end{cases} \quad (3.15)$$

Finalmente, a matriz de dispersão inicial  $\Sigma_{init}$  também é um parâmetro dependente do problema, e deve ser definida a partir de conhecimento *a priori* sobre a dispersão das variáveis de entrada, ou a partir de um conjunto de dados iniciais, por exemplo, definindo  $\Sigma_{init}$  como a covariância amostral do conjunto de dados.

O algoritmo 2 sumariza os passos realizados a cada iteração do processo de agrupamento evolutivo proposto.

---

**Algoritmo 2** Algoritmo de agrupamento evolutivo participativo Gaussiano

---

- 1: Calcula  $\rho_i$  e  $a_i$  para todos os grupos
  - 2: Seleciona o grupo com maior compatibilidade ( $j$ )
  - 3: **if**  $\rho_i < T_\rho \ \forall i$  e  $a_j > T_a$  **then**
  - 4:   Cria um novo grupo
  - 5: **else**
  - 6:   Atualiza os parâmetros do grupo  $j$
  - 7: **end if**
  - 8: Seleciona grupo atualizado ou criado ( $idx$ )
  - 9: **for all** Grupos ( $i$ ) **do**
  - 10:   **if** Compatibilidade entre  $c_i$  e  $c_{idx}$  for maior que  $T_\rho$  **then**
  - 11:     Une grupos redundantes
  - 12:   **end if**
  - 13: **end for**
- 

### 3.3 Funções de Pertinência Gaussianas Multivariáveis

Todos os modelos nebulosos evolutivos descritos nesse capítulo utilizam funções de pertinência Gaussianas multivariáveis descritas por:

$$H(x) = \exp \left[ -\frac{1}{2}(x - c)\Sigma^{-1}(x - c)^T \right] \quad (3.16)$$

onde  $x$  é um vetor de entrada  $1 \times m$ ,  $c$  é o centro do grupo  $1 \times m$  e  $\Sigma$  é uma matriz positiva definida simétrica  $m \times m$ . O centro do grupo é definido como o valor modal e representa o valor típico de  $H(x)$ . A matriz  $\Sigma$  denota a dispersão e representa o espalhamento de  $H(x)$  (Pedrycz e Gomide, 2007).

Existem diversas motivações para se escolher funções de pertinência Gaussianas, tais como:

- o suporte infinito dessas funções, o que faz com que entradas não sejam omitidas;
- apenas dois parâmetros são necessários e podem ser facilmente estimados;
- iterações entre variáveis de entrada podem ser facilmente capturadas pela matriz de dispersão.

É interessante ressaltar que, como o processo de modelagem é adaptativo e é executado recursivamente, o suporte infinito dessas funções é conveniente, uma vez que o limite dos valores das variáveis de entrada geralmente não é conhecido *a priori*.

Também é conhecido da literatura que funções de base radial Gaussianas são boas candidatas para se representar propriedades locais (Lippmann, 1991). Além disso, Gaussianas proveem uma forma de construir funções de base e sistemas nebulosos podem ser representados como uma combinação linear de funções de base nebulosas. Combinações lineares de funções de base nebulosas são capazes de aproximar uniformemente qualquer função real contínua em um conjunto compacto a uma precisão arbitrária, isto é, são aproximadores universais (Wang e Mendel, 1992b). Modelos nebulosos funcionais com Gaussianas também são aproximadores universais (Kreinovich et al., 1998).

A grande maioria das metodologias de modelagem nebulosa que geram regras a partir de algoritmos de agrupamento de dados, sejam estas evolutivas (Angelov e Filev, 2005; Kasabov e Song, 2002; Lima et al., 2009; Lughofer, 2008a) ou não (Babuska e Verbruggen, 1996; Chang e Chang, 2006; Chiu, 1994; Hell, 2008; Yager e Filev, 1994), realizam o agrupamento no espaço de entrada (ou entrada e saída) e criam regras utilizando conjuntos nebulosos unidimensionais (ou univariáveis), que são gerados a partir da projeção dos grupos encontrados no eixo de cada variável. Em seguida, durante o processo de inferência nebulosa, as relações nebulosas induzidas pelo antecedente de cada regra são calculadas, utilizando um operador de agregação (por exemplo, uma *t-norma*) dos conjuntos nebulosos de cada entrada. Apesar dessa abordagem ser comumente utilizada, pode gerar perda de informação, caso as variáveis de entrada interajam (Abonyi et al., 2002; Kim et al., 1998). Por exemplo, é comum, em identificação de sistemas e previsão de séries temporais, utilizar valores atrasados de entradas e/ou saídas como entradas de um modelo. Esses valores atrasados tendem, geralmente, ser altamente relacionados.

Com o objetivo de evitar essa perda de informação, todos os modelos nebulosos evolutivos propostos nesse capítulo utilizam funções de pertinência Gaussianas multivariáveis para representar os conjuntos nebulosos dos antecedentes de cada regra. Para isso, cada grupo encontrado pelo algoritmo de agrupamento é associado a uma regra nebulosa e os parâmetros da função

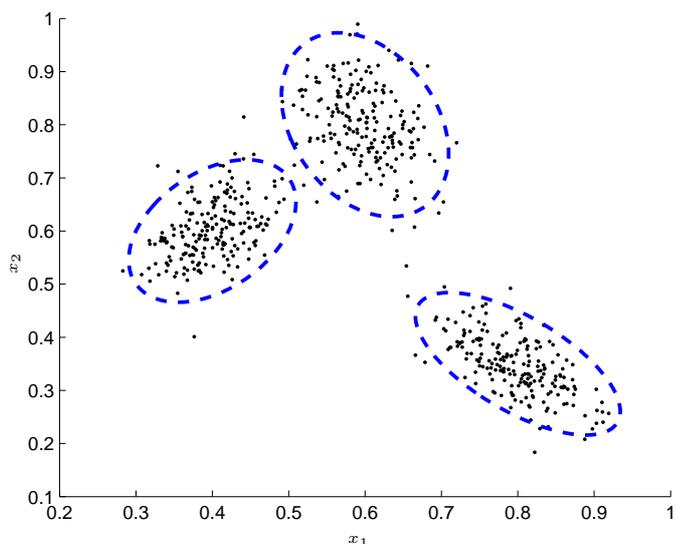


Figura 3.6: Estrutura de grupos do espaço de entrada

de pertinência do antecedente dessa regra são definidos como os parâmetros do grupo correspondente. Essas funções de pertinência utilizam a informação de interação entre as variáveis, presente nas matrizes de dispersão de cada grupo, evitando a perda dessa informação<sup>1</sup>.

A figura 3.6 ilustra um cenário típico onde a abordagem de geração de regras nebulosas, baseada no uso da projeção de grupos, pode implicar em perda de informação (as elipses, delimitando cada grupo, representam seu limiar de compatibilidade). Enquanto a figura 3.6 ilustra a estrutura de grupos do espaço de entrada, a figura 3.7 ilustra as funções de pertinência Gaussianas univariáveis criadas pela projeção dos grupos em cada um dos eixos do espaço de entrada. O exemplo em questão é formado por 3 grupos, sendo que cada grupo representa o antecedente de uma regra nebulosa. As 3 relações nebulosas das entradas são:

$$\begin{aligned}
 x_1 \text{ é } A_1 \text{ e } x_2 \text{ é } B_1 \\
 x_1 \text{ é } A_2 \text{ e } x_2 \text{ é } B_2 \\
 x_1 \text{ é } A_3 \text{ e } x_2 \text{ é } B_3
 \end{aligned} \tag{3.17}$$

A figura 3.8 ilustra as relações nebulosas induzidas pelas funções de pertinência univariáveis, utilizando a *t-norma* produto (*e*) como operador de agregação para o antecedente de cada regra, e a figura 3.9 ilustra as funções de pertinência Gaussianas multivariáveis construídas utilizando o centro e matriz de dispersão de cada grupo. As funções de pertinência multivariáveis são utilizadas para gerar 3 regras com os seguintes antecedentes:

<sup>1</sup>o termo *interação* refere-se à valores não nulos nos elementos não diagonais da matriz de dispersão

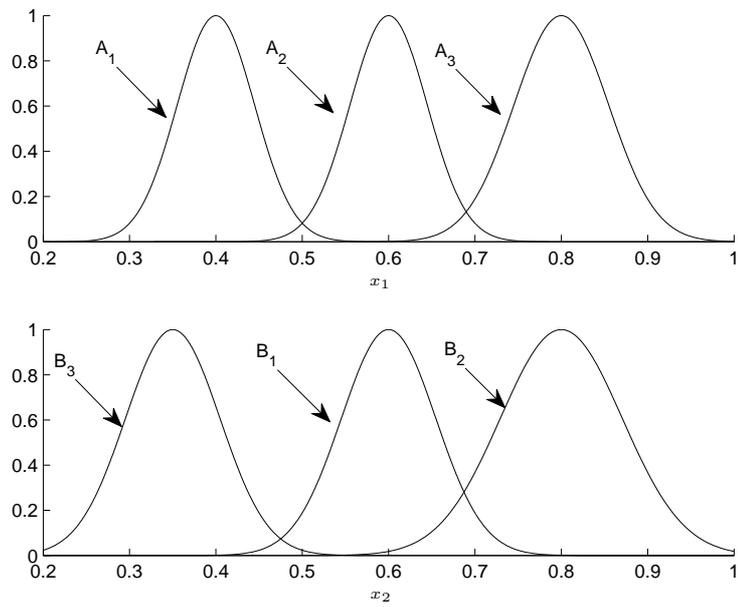


Figura 3.7: Funções de pertinência Gaussianas univariáveis (projeção dos grupos da figura 3.6 nos eixos do espaço de entrada)

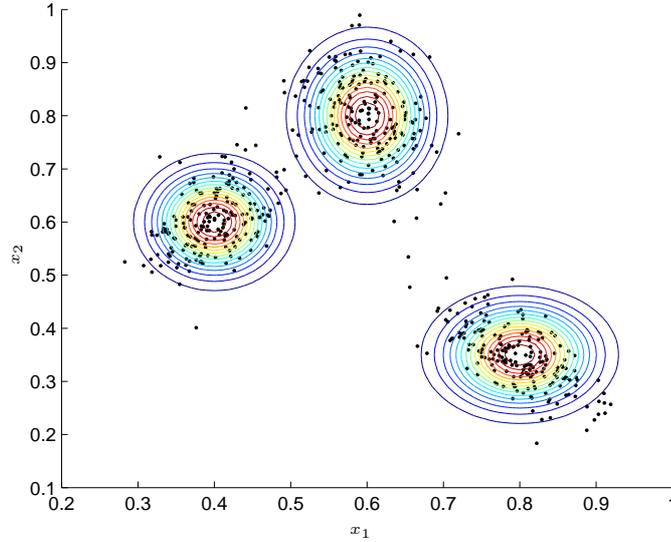


Figura 3.8: Relações nebulosas induzidas pelo antecedente das regras utilizando a *t-norma* produto como operador lógico  $E$

$$\begin{aligned}
 x \text{ é } A_1 \\
 x \text{ é } A_2 \\
 x \text{ é } A_3
 \end{aligned}
 \tag{3.18}$$

Comparando as relações nebulosas, ilustradas na figura 3.8, geradas pela projeção dos grupos, e as funções de pertinência multivariáveis presentes na figura 3.9, pode-se notar a perda de informação causada pela metodologia de geração de regras nebulosas a partir da projeção dos grupos. O formato e a orientação das relações geradas (figura 3.8) diferem dos grupos originais (figura 3.6).

Na verdade, caso o produto seja utilizado como *t-norma*, pode-se demonstrar que as relações nebulosas induzidas pela metodologia, baseada na projeção, são equivalentes às funções Gaussianas multivariáveis, somente se não houver interação entre as entradas. Para visualizar isso em um espaço bi-dimensional, sem perda de generalidade, assume-se que o centro e a matriz de dispersão do grupo são dados por:

$$\begin{aligned}
 c &= (c_1 \ c_2) \\
 \Sigma &= \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix}
 \end{aligned}$$

onde  $\sigma_{12} = \sigma_{21}$

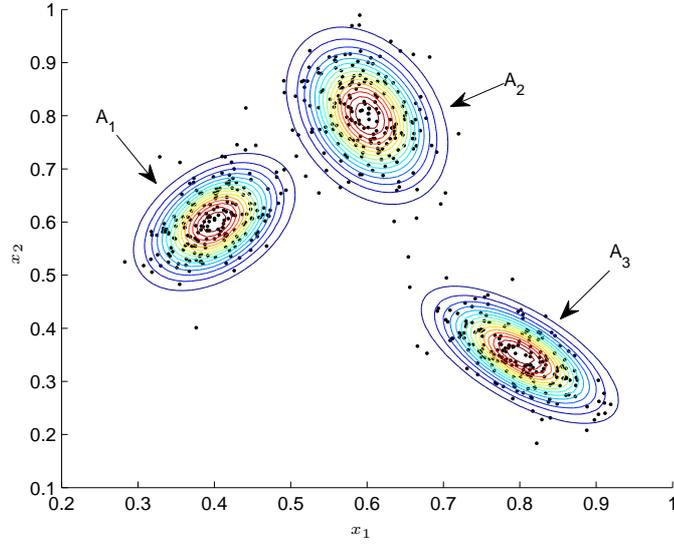


Figura 3.9: Funções de pertinência Gaussianas multivariáveis definidas a partir do centro e matriz de dispersão de cada grupo

Caso a metodologia de geração de regras nebulosas baseada na projeção seja utilizada e os parâmetros das funções de pertinência univariáveis sejam extraídos do centro do grupo e dos valores da diagonal da matriz de dispersão correspondente, tem-se que

$$\begin{aligned}\mu_1(x_1) &= \exp \left[ -\frac{1}{2} \frac{(x_1 - c_1)^2}{\sigma_1^2} \right] \\ \mu_2(x_2) &= \exp \left[ -\frac{1}{2} \frac{(x_2 - c_2)^2}{\sigma_2^2} \right]\end{aligned}\quad (3.19)$$

Utilizando-se a *t-norma* produto, a relação nebulosa induzida pelo antecedente de cada regra é dada por:

$$\begin{aligned}R(x_1, x_2) &= \exp \left[ -\frac{1}{2} \frac{(x_1 - c_1)^2}{\sigma_1^2} \right] \exp \left[ -\frac{1}{2} \frac{(x_2 - c_2)^2}{\sigma_2^2} \right] \\ &= \exp \left[ -\frac{1}{2} \left( \frac{(x_1 - c_1)^2}{\sigma_1^2} + \frac{(x_2 - c_2)^2}{\sigma_2^2} \right) \right] \\ &= \exp \left[ -\frac{1}{2} (x - c) S^{-1} (x - c)^T \right]\end{aligned}\quad (3.20)$$

onde  $S$  é:

$$S = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$$

Analisando a relação nebulosa expressa por (3.20) e a função de pertinência Gaussiana multivariável expressa por (3.16) fica claro que a relação nebulosa equivale à função de pertinência multivariável se e somente se os elementos descrevendo a interação entre as variáveis ( $\sigma_{12}$  e  $\sigma_{21}$ ) forem ignorados.

### 3.4 Modelo Funcional Evolutivo Gaussiano Multivariável

Essa seção descreve a metodologia de modelagem funcional evolutiva proposta baseada em regras com antecedentes gerados a partir do algoritmo de agrupamento evolutivo descrito na seção 3.2.

O modelo proposto nessa seção pode ser visto como um conjunto de regras nebulosas do tipo Takagi-Sugeno de primeira ordem, em que os antecedentes possuem funções de pertinência multivariáveis e os parâmetros dos consequentes são estimados por otimização local utilizando o algoritmo recursivo de mínimos quadrados ponderados.

A cada iteração, o número de regras do modelo é igual ao número de grupos já encontrados pelo algoritmo de agrupamento. Assim, a cada iteração um novo grupo pode ser criado, um grupo existente pode ter seus parâmetros modificados ou dois grupos redundantes podem ser unidos. Em outras palavras, uma regra pode ser criada, atualizada ou duas regras podem ser unidas. Os antecedentes de cada regra apresentam-se na forma:

$$x^k \text{ é } A_i \tag{3.21}$$

onde  $x^k$  é um vetor de entrada  $1 \times m$  e  $A_i$  é um conjunto nebuloso com função de pertinência Gaussiana multivariável (3.16) com parâmetros extraídos do grupo correspondente.

O modelo é formado por regras do tipo:

$$R_i : \text{ Se } x^k \text{ é } A_i \text{ então } y_i = a_{i0} + \sum_{j=1}^m a_{ij}x_j \tag{3.22}$$

onde  $R_i$  é a  $i$ -ésima regra nebulosa, para  $i = 1, \dots, g^k$  ( $g^k$  é o número de grupos no instante  $k$ ), e  $a_{ij}$  são os parâmetros do consequente da regra  $i$ .

A saída do modelo é calculada como a média ponderada da saída de cada regra, isto é:

$$\hat{y}^k = \sum_{i=1}^{g^k} \Psi(x^k)_i y_i \tag{3.23}$$

com funções de pertinência normalizadas:

$$\Psi(x^k)_i = \frac{\exp [(x^k - c_i^k)\Sigma_i^{-1}(x^k - c_i^k)^T]}{\sum_{j=1}^{g^k} \exp [(x^k - c_j^k)(\Sigma_j^k)^{-1}(x^k - c_j^k)^T]} \quad (3.24)$$

onde  $c_i^k$  e  $\Sigma_i^k$  são o centro e a matriz de dispersão do grupo  $i$  na iteração  $k$ .

Os parâmetros do conseqüente de cada regra são atualizados utilizando-se o algoritmo recursivo de mínimos quadrados ponderados (WRLS<sup>2</sup>) (Astrom e Wittenmark, 1988; Ljung, 1999), de forma similar a outros modelos nebulosos evolutivos (Angelov e Filev, 2004; Lughofer, 2008a). Assim, os parâmetros do conseqüente e a matriz  $Q_i$  da fórmula de atualização para a regra  $i$  na iteração  $k$  é dada por:

$$\begin{aligned} a_i^{k+1} &= a_i^k + Q_i^{k+1} x^k \Psi_i(x^k) [y_i^k - ((x^k)^T a_i^k)] \\ Q_i^{k+1} &= Q_i^k - \frac{\Psi_i(x^k) Q_i^k x^k (x^k)^T Q_i^k}{1 + (x^k)^T Q_i^k x^k} \end{aligned} \quad (3.25)$$

Assim como o algoritmo de agrupamento, o modelo funcional proposto nessa seção pode iniciar a partir de uma única observação ou a partir de um conjunto de amostras.

Caso seja iniciado a partir de um conjunto de amostras, um algoritmo de agrupamento em batelada é utilizado para estimar os parâmetros dos antecedentes das regras iniciais e o algoritmo de mínimos quadrados ponderados é utilizado para estimar os parâmetros dos conseqüentes.

Caso seja iniciado a partir de uma única observação, uma única regra é criada com função de pertinência do antecedente centralizada na observação e matriz de dispersão ajustada como um valor pré-definido ( $\Sigma_{init}$ ). Os parâmetros dos conseqüentes são ajustados como  $a_1^0 = [y^0 \ 0 \ \dots \ 0]$  e  $Q^k = \omega I_{m+1}$ , onde  $I_{m+1}$  é a matriz identidade  $m + 1 \times m + 1$  e  $\omega$  é um valor real grande, por exemplo,  $\omega \in [100, \ 10000]$ , (Astrom e Wittenmark, 1988).

À medida que novas observações são processadas pelo algoritmo de agrupamento, esse pode criar, atualizar ou unir grupos. Assim, o conjunto de regras do modelo também deve ser atualizado.

Caso um novo grupo seja criado, a regra correspondente é criada com parâmetros do antecedente extraídos do grupo. O parâmetro  $a_{new}$  do conseqüente é definido como a média ponderada dos parâmetros das regras existentes:

$$a_{new} = \frac{\sum_{i=1}^{g^k} a_i \rho_i}{\sum_{i=1}^{g^k} \rho_i} \quad (3.26)$$

e a matriz de covariância é ajustada para  $\omega I_{m+1}$ .

Caso um novo grupo seja atualizado, os parâmetros dos antecedentes também são atualizados de acordo com o novo centro do grupo e matriz de dispersão, e os parâmetros dos conseqüentes são atualizados de acordo com (3.25).

---

<sup>2</sup>do inglês *weighted recursive least squares*

Finalmente, se dois grupos  $i$  e  $j$  são unidos, então os parâmetros dos consequentes da regra resultante são computados da seguinte forma:

$$a_{new} = \frac{a_i \rho_i + a_j \rho_j}{\rho_i + \rho_j} \quad (3.27)$$

e a matriz de covariância é ajustada para  $\omega I_{m+1}$

O modelo proposto nessa seção difere dos demais modelos funcionais nebulosos evolutivos baseados em um conjunto de regras, propostos na literatura, pois:

- utiliza funções de pertinência multivariáveis para os conjunto nebulosos dos antecedentes das regras para evitar a perda de informação relacionada a interação entre as variáveis de entrada;
- o algoritmo de agrupamento proposto é robusto a ruídos e *outliers* uma vez que é derivado do aprendizado participativo, um conceito que provê um mecanismo de suavização de dados incompatíveis (Yager, 2007);
- a criação de novas regras é realizada por um mecanismo automático para ajustar valores de limiares considerando a dimensão do espaço de entrada, de forma a evitar a *maldição da dimensionalidade*.

O processo de modelagem nebulosa funcional evolutiva proposto é sumarizado pelo algoritmo 3.

---

**Algoritmo 3** Algoritmo de modelagem funcional evolutiva proposto

---

- 1: Calcula a saída do modelo
  - 2: Apresenta a nova observação ao algoritmo de agrupamento
  - 3: **if** Um grupo foi criado **then**
  - 4:   Cria uma nova regra
  - 5: **end if**
  - 6: **if** Um grupo foi modificado **then**
  - 7:   Atualiza os parâmetros do antecedente da regra relacionada
  - 8:   Atualiza os parâmetros do consequente da regra usando WRLS
  - 9: **end if**
  - 10: **if** Dois grupos foram unidos **then**
  - 11:   Une as regras correspondentes
  - 12: **end if**
-

O algoritmo proposto tem uma complexidade temporal  $O(g^k)^3$ , para avaliar e atualizar os parâmetros a cada iteração, onde  $g^k$  é o número de regras na iteração  $k$ . Note que um modelo nebuloso evolutivo típico, baseado em agrupamento, também possui uma complexidade temporal de, no mínimo,  $O(g^k)$ . A cada iteração, alguma medida de similaridade deve ser estimada entre a amostra corrente e todos os grupos existentes, sendo que cada grupo está, normalmente, associado a uma regra.

### 3.5 Modelo Linguístico Evolutivo para Detecção e Diagnóstico Adaptativo de Falhas em Sistemas Dinâmicos

O problema de detecção e diagnóstico de falhas em sistemas dinâmicos tem sido foco de diversos trabalhos durante as últimas décadas, dada a importância dessa área em aplicações reais, especialmente no campo da indústria. Basicamente, as estratégias de detecção de falhas em sistemas dinâmicos podem ser baseadas em modelos, que representam a física do processo, construídos a partir de conhecimento adquirido diretamente de especialistas ou em dados. Na prática, os métodos citados podem ser usados de forma combinada (Venkatasubramanian et al., 2003a,b,c).

Entre os métodos que empregam modelos da física do processo, pode-se citar o emprego de: observadores de estados do tipo Luenberger dedicados (Luenberger, 1964), filtro de Kalman (Kalman, 1960), observadores robustos, estimação de parâmetros (Isermann, 1984), entre outros.

As técnicas que utilizam o conhecimento adquirido de especialistas empregam lógica clássica ou lógica nebulosa. O termo “conhecimento” se refere à experiência que um ser humano tem em um determinado processo. Este método é normalmente utilizado quando o comportamento do sistema monitorado obedece a uma regra empírica.

Os métodos baseados em dados não utilizam informação sobre a física do processo. A detecção e diagnóstico de falhas é realizada a partir de dados históricos. Geralmente, esses métodos transformam dados lidos do processo em um conjunto de características, que são utilizadas para definir a topologia e estimar os parâmetros de um modelo capaz de realizar o diagnóstico (Venkatasubramanian et al., 2003c). O conjunto de dados utilizado deve descrever o comportamento do sistema monitorado durante operação normal, e também durante todas as possíveis condições de falha consideradas. Exemplos de métodos baseados em dados incluem sistemas especialistas (Cho e Park, 1997), métodos estatísticos (Mehranbod et al., 2005), redes neurais (Samanta, 2004), modelos nebulosos híbridos (Mendonça et al., 2009), entre outros.

Geralmente, os métodos convencionais de detecção e diagnóstico de falhas são baseados em informações extraídas do sistema monitorado durante todos os modos de operação a serem identificados, isto é, condições de operações normais e falhas (Venkatasubramanian et al.,

---

<sup>3</sup>notação assintótica (Cormen et al., 2001)

2003a,b,c). No entanto, em diversos casos práticos, muitos modos de operação podem ser desconhecidos, devido a alta confiabilidade e/ou complexidade do sistema monitorado. Nessas situações, torna-se impraticável a utilização de métodos tradicionais de detecção e diagnóstico de falhas, porém técnicas de monitoramento da condição de operação podem ser utilizadas para detectar condições de operações anormais, que podem ser relacionadas a falhas. Diversos trabalhos na literatura descrevem métodos de monitoramento da condição de operação. Esses métodos, em geral, são baseados em controle estatístico de processos (Montgomery, 2001) ou em aprendizado de máquina (Dasgupta e Forrest, 1996; Dong et al., 2006; Rajagopalan e Ray, 2006; Timusk et al., 2008; Wong et al., 2006), que aplicam técnicas de detecção de novidades (Markou e Singh, 2003a,b) para detectar condições de operação anormais.

Esses métodos de monitoramento podem ser utilizados para detecção de falhas, no entanto, geralmente, carecem da habilidade de diagnóstico, já que são capazes de detectar operações anormais, mas incapazes de classificá-las. Essa limitação ocorre pois, habitualmente, esses métodos são desenvolvidos somente com a utilização de informações do sistema monitorado durante modo de operação normal (Dasgupta e Forrest, 1996).

Uma característica importante e desejada para métodos de detecção e diagnóstico de falhas é a adaptabilidade (Venkatasubramanian et al., 2003a). Espera-se que o sistema monitorado mude ao longo do tempo, devido a variações em entradas exógenas, troca de componentes, manutenção, etc. Assim, metodologias de detecção e diagnóstico de falhas podem ser invalidadas com o passar do tempo, se os parâmetros ou até mesmo a estrutura do modelo utilizado não se adaptarem a essas mudanças.

Essa seção propõe um método de detecção e diagnóstico adaptativo de falhas em sistemas dinâmicos. A metodologia proposta é capaz de detectar novos modos de operação (mudanças no ponto de operação ou falhas) e também incorporar essas informações em um classificador nebuloso evolutivo. Esse classificador é, então, capaz de realizar um diagnóstico adaptativo de falhas, isto é, se a informação a respeito de um novo modo de operação é incorporada no classificador, este deve ser capaz de identificar esse novo modo de operação em suas próximas ocorrências. O classificador é treinado de forma evolutiva através de uma técnica de aprendizagem semi supervisionada, evitando ao máximo a utilização de dados rotulados. Optou-se por essa técnica de aprendizado, para que o sistema de diagnóstico de falhas funcionasse da forma mais automatizada possível, evitando ao máximo a necessidade de informações externas sobre o modo de operação do sistema.

Embora já existam métodos de detecção de falhas baseados em modelos nebulosos evolutivos (Filev e Tseng, 2006; Lughofer, 2008; Lughofer e Guardiola, 2008; Wang e Vrbanek, 2008), nenhum desses é capaz de realizar um diagnóstico adaptativo do sistema monitorado. Deste modo, a metodologia proposta nessa seção difere das encontradas na literatura, pois descreve um sistema de detecção e diagnóstico adaptativo de falhas capaz de realizar diagnóstico sem a necessidade de modelos e, simultaneamente, atualizar as informações sobre modos de operação já conhecidos à medida que esses mudem ao longo do tempo, além de incorporar informação sobre novos modos de operação à medida que estes sejam detectados.

Sumarizando, a metodologia de detecção e diagnóstico de falhas proposta nessa seção é aplicável em sistemas que apresentem as seguintes características:

- alta complexidade, em que a obtenção de um modelo matemático, baseado na física do

processo, é inviável;

- falta de dados que descrevam modos de operação de falha, por se tratar, por exemplo, de uma planta nova;
- possível presença de modos de falha desconhecidos, devido à complexidade do sistema monitorado. Além disso, os modos podem também mudar ao longo do tempo, devido a desgastes, manutenções e trocas de componentes, por exemplo.

### 3.5.1 Classificador Nebuloso Evolutivo

O classificador nebuloso evolutivo proposto é baseado em um conjunto de regras nebulosas linguísticas, geradas a partir do algoritmo de agrupamento evolutivo descrito na seção 3.2. Cada regra possui antecedentes criados de forma análoga ao modelo funcional evolutivo descrito na seção 3.4, e consequentes definidos como o rótulo do modo de operação que a regra descreve.

O classificador é formado por um conjunto de regras nebulosas da seguinte forma:

$$R_i : \text{ Se } x^k \text{ é } A_i \text{ então } y_i \text{ é } MO_i \quad (3.28)$$

sendo que  $R_i$  é a  $i$ -ésima regra nebulosa, para  $i = 1, \dots, g^k$ ,  $g^k$  é o número de regras na iteração  $k$ ,  $x^k$  é o vetor de entrada  $1 \times m$ ,  $m$  é o número de entradas utilizadas e  $MO_i$  é o rótulo do modo de operação descrito pela regra.

Mais de uma regra pode ser utilizada para descrever um modo de operação, uma vez que um dado modo de operação pode depender do contexto. Por exemplo, a operação normal de um sistema pode depender de variáveis externas, como temperatura, pressão, etc. Nessas circunstâncias, apenas uma regra pode não ser suficiente para descrever todas as possíveis variações do mesmo modo de operação. Assim, o classificador agrega regras com o mesmo consequente, utilizando uma *s-norma*. O resultado da agregação pode ser interpretado como regras da seguinte forma:

$$\begin{aligned} & (\text{Se } x^k \text{ é } A_i) \text{ ou } (\text{se } x^k \text{ é } A_j) \text{ ou } \dots \\ & \text{ou } (\text{se } x^k \text{ é } A_k) \text{ então } y_i \text{ é } MO_i \end{aligned}$$

O resultado dessa agregação é um grau de pertinência para cada modo de operação conhecido, de forma que o diagnóstico final a cada iteração corresponde ao modo de operação com o maior grau de pertinência. Assim, a saída do classificador é definida como o modo de operação estimado  $\hat{y}^k$  e seu respectivo grau de pertinência  $\mu_{\hat{y}^k}$ . A figura 3.10 ilustra o classificador proposto.

### 3.5.2 Metodologia de Detecção de Diagnóstico Adaptativo de Falhas

Inicialmente, características devem ser extraídas (ou selecionadas) a partir de medidas de sensores do sistema monitorado. O método de extração/seleção de características a ser utili-

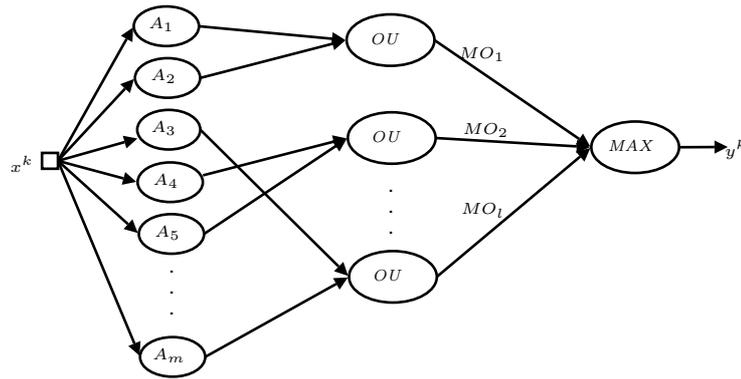


Figura 3.10: Estrutura do classificador nebuloso evolutivo proposto para diagnóstico de falhas

zado depende do problema, porém a técnica utilizada deve ser capaz de executar em tempo real, uma vez que a metodologia de detecção e diagnóstico de falhas proposta também deve executar em tempo real. Esse processo é realizado para transformar medidas de sensores do sistema monitorado em informações (características), capazes de distinguir diferentes modos de operação do sistema e, por isso, exigem conhecimento *a priori* sobre o sistema. Esse processo pode ser definido como simplesmente a seleção de algumas medidas de sensores ou transformações no domínio do tempo, domínio da frequência ou domínio misto. Exemplos de características comumente utilizadas no domínio do tempo são: valor eficaz, kurtose, média e parâmetros de modelos autorregressivos. Exemplos de características do domínio da frequência são: espectro de potência e picos da transformada de Fourier. Finalmente, exemplos de características de domínio misto são coeficientes wavelet.

Essas características são utilizadas como entradas do algoritmo de agrupamento evolutivo descrito na seção 3.2 e do classificador nebuloso evolutivo, para estimar o modo de operação atual. O algoritmo de agrupamento processa cada vetor de características, atualizando sua estrutura de grupos. O classificador evolutivo também processa essas características, estimando o modo de operação do sistema e o grau de pertinência correspondente.

Caso um novo grupo seja criado, uma nova regra é criada, com os parâmetros do antecedente extraídos do grupo. O consequente da regra é definido da seguinte forma: caso o grau de pertinência do modo de operação estimado pelo classificador seja menor que um limiar pré-definido  $T_{\mu_y} \in (0, 1]$ , o operador do sistema deve ser notificado sobre um possível novo modo de operação. Esse operador age como um oráculo capaz de rotular as características apresentadas, ou seja, identificar o modo de operação atual, baseado nas características extraídas do sistema monitorado. Nesse caso, o modo de operação definido pelo operador é utilizado como consequente da nova regra. Caso o grau de pertinência do modo de operação estimado pelo classificador seja maior ou igual ao limiar  $T_{\mu_y}$ , o operador não é notificado, e o consequente da regra é definido como o modo de operação estimado pelo classificador.

Caso um grupo seja atualizado, os parâmetros da regra correspondente são atualizados como os novos valores do centro e matriz de dispersão do grupo.

Conforme descrito anteriormente, é esperado que as características escolhidas como entradas

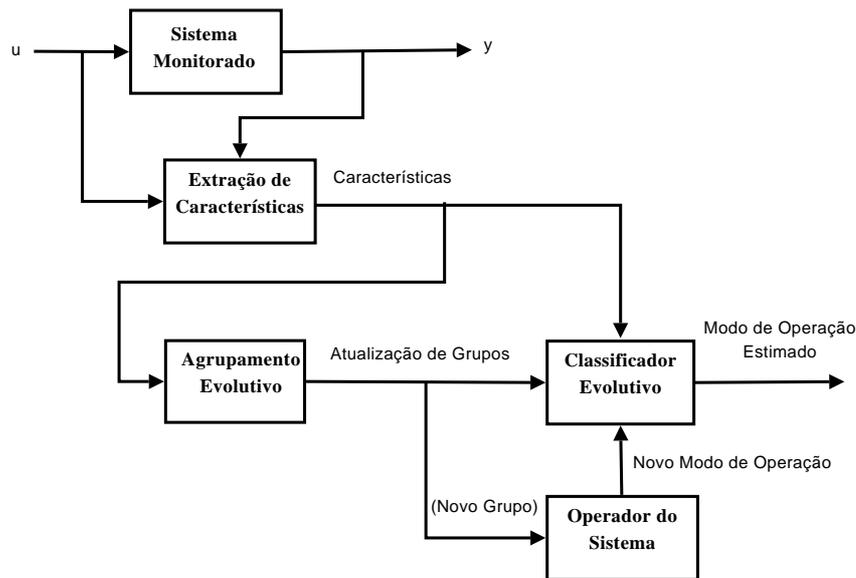


Figura 3.11: Metodologia de detecção e diagnóstico adaptativo de falhas proposta

da metodologia proposta sejam capazes de distinguir todos os modos de operação possíveis. Porém, isso apenas é assumido, uma vez que alguns modos de operação podem ser desconhecidos durante o processo de definição do conjunto de características. Afim de tratar esse problema, regras similares são unidas apenas quando possuem o mesmo consequente, mesmo se o critério de união de grupos do algoritmo de agrupamento for válido para os grupos associados às regras. Essa restrição permite a existência de duas regras com antecedentes similares e consequentes distintos. Isso pode ser interpretado como condições de falhas que possuam os mesmos “sintomas”, como, por exemplo, quando um médico tenta distinguir entre uma gripe comum e H1N1 somente pelos sintomas gripais clássicos. Trabalhos futuros tratarão dessa limitação através da integração de metodologias adaptativas de extração/seleção de características (Lughofer, 2010, 2011; Ozawa et al., 2010), capazes de adaptar o conjunto de entrada baseado em variações do fluxo de dados.

A figura 3.11 ilustra o diagrama da metodologia de detecção e diagnóstico adaptativa de falhas proposta.

A metodologia proposta pode ser iniciada a partir de uma única característica ou a partir de um conjunto de características, descrevendo um ou diversos modos de operação. Todas as características iniciais utilizadas devem ser rotuladas, ou seja, o modo de operação correspondente deve ser conhecido. Caso o processo inicie a partir de uma característica, uma única regra nebulosa é criada com função de pertinência do antecedente centrada nessa característica e com matriz de dispersão pré-definida,  $\Sigma_{init}$ . Assim como no modelo funcional, o valor da matriz de dispersão  $\Sigma_{init}$  deve ser definido a partir de conhecimento *a priori* sobre o sistema monitorado. O consequente da regra é então definido como o modo de operação correspondente. Caso o processo inicie a partir de um conjunto de características, essas devem ser divididas de acordo com o modo de operação correspondente e cada subconjunto de características resultante é

utilizado como entrada de um algoritmo de agrupamento em batelada, gerando então os grupos associados ao modo de operação e, conseqüentemente, as regras nebulosas.

Conforme descrito anteriormente, o classificador nebuloso evolutivo proposto utiliza uma técnica de aprendizagem semi supervisionada. Apenas as características que gerem novos grupos e cujo modo de operação estimado pelo classificador esteja associado a um grau de pertinência menor que o limiar  $T_{\mu_y}$  necessitam ser rotuladas. É então esperado que, a maior parte do tempo, o operador não tenha que intervir no processo de diagnóstico, tornando a metodologia o mais automatizada o possível.

O grau de automação da metodologia está relacionado ao valor do limiar  $T_{\mu_y} \in (0, 1]$ . Valores baixos do limiar aumentam o grau de automação, uma vez que diminui a frequência de intervenções do operador, já que o conseqüente das regras tende a ser definido, na maior parte das vezes, como o modo de operação estimado pelo classificador. Porém, esse valor não pode ser definido como 0, pois, caso isso ocorra, novos modos de operação não serão identificados. Caso  $T_{\mu_y} = 1$ , o conseqüente de todas as novas regras será definido pelo operador, diminuindo ao máximo o grau de automação.

Nota-se que, geralmente  $T_{\mu_y}$  será definido como um valor pequeno. Como esse limiar é utilizado para decidir a forma de definição do conseqüente de uma nova regra criada, é esperado que o grau de pertinência de uma entrada, responsável pela criação de uma nova regra, seja baixo. Assim, na prática,  $T_{\mu_y}$  assume valores no intervalo  $[0.01, 0.1]$ .

O algoritmo 4 detalha os passos do treinamento recursivo da metodologia de detecção e diagnóstico adaptativo de falhas proposta.

Assim como o modelo funcional proposto nesse capítulo, a metodologia de detecção e diagnóstico de falhas proposta possui complexidade temporal  $O(g^k)$ , para avaliar uma entrada e atualizar os parâmetros a cada iteração.

---

**Algoritmo 4** Algoritmo proposto para detecção e diagnóstico adaptativo de falhas

---

Estima a saída do classificador para a nova amostra

Apresenta a nova amostra ao algoritmo de agrupamento

**if** Um grupo foi criado **then**

**if**  $\mu_{\hat{y}^k} > T_{\mu_y}$  **then**

        Consequente da nova regra é definido como a saída do classificador

**else**

        Consequente da nova regra é definido pelo operador

**end if**

**end if**

**if** Um grupo foi modificado **then**

    Atualiza os parâmetros do antecedente da regra relacionada

**end if**

**if** Dois grupos foram unidos **then**

    Une as regras correspondentes

**end if**

---

## Capítulo 4

# Árvores de Regressão Linear Nebulosas Evolutivas

### 4.1 Introdução

Conforme descrito anteriormente, a grande maioria das técnicas de modelagem nebulosa evolutivas propostas na literatura utiliza informação sobre a organização espacial das variáveis de entrada e/ou saída para definir um conjunto de regras de forma adaptativa a partir de um fluxo de dados. Geralmente, a organização espacial dessas variáveis é estimada através de um algoritmo de agrupamento evolutivo. Os modelos propostos no capítulo anterior também seguem essa metodologia. Essa metodologia de definição da estrutura dos modelos pode ser vista como uma versão adaptativa de uma metodologia bastante difundida na modelagem nebulosa clássica (Babuska, 1998; Chiu, 1994; Jang e Sun, 1997; Yager e Filev, 1994), baseada em algoritmos de agrupamento em batelada, onde grupo, encontrados por algoritmos de agrupamento, definem o antecedente das regras do modelo.

Esse capítulo propõe uma nova abordagem para a modelagem nebulosa evolutiva para problemas de regressão. Os modelos propostos nesse capítulo não utilizam informação sobre a organização espacial das variáveis de entrada e/ou saída para definição da topologia do modelo. A metodologia proposta para definição da topologia dos modelos descritos nesse capítulo utiliza informação sobre a qualidade do modelo resultante. A qualidade leva em consideração a precisão e também o número de parâmetros livres, resultando em modelos eficientes e evitando o problema de sobreajuste.

Os modelos evolutivos nebulosos propostos nesse capítulo são baseados em árvores de regressão nebulosas evolutivas construídas a partir de um fluxo de dados. Árvores de regressão linear (Quinlan, 1992) são generalizações de árvores de regressão (Breiman et al., 1984). Para isso, os modelos de ordem zero (valor médio da saída) presentes nas folhas de uma árvore de regressão clássica são substituídos por modelos lineares das variáveis de entrada. Essa substi-

tuição é realizada para aumentar a eficiência do modelo e diminuir sua complexidade (tamanho da árvore) (Quinlan, 1992). Árvores de regressão linear possuem critérios de divisão em seus nós internos utilizados para particionar o espaço de entrada em diferentes regiões associadas aos modelos lineares presentes nas folhas. Árvores de regressão linear nebulosas substituem os critérios de divisão rígidos presentes nos nós internos por um par de funções de pertinência, de forma similar às árvores de decisão nebulosas (Janikow, 1998; Yuan e Shaw, 1995). Ao introduzir conjuntos nebulosos nos nós internos da árvore, as fronteiras exatas presentes nas árvores originais desaparecem e, para cada vetor de entrada, todos os galhos da árvore são ativados (com diferentes graus de ativação). Essa modificação resulta em partições sobrepostas do espaço de entrada e um modelo de regressão baseado na soma ponderada de modelos lineares locais.

O algoritmo incremental proposto nesse trabalho atualiza a topologia da árvore a partir de um fluxo de dados, utilizando testes estatísticos de seleção de modelos. Os testes são computados recursivamente e constroem a árvore substituindo folhas por subárvores, que melhorem a qualidade do modelo resultante. Para avaliar a qualidade do modelo, os testes utilizados levam em consideração a precisão e o número de parâmetros livres, gerando modelos eficientes e parcimoniosos.

A metodologia de partição do espaço de entrada proposta nesse capítulo é similar às metodologias propostas em Wang et al. (2010) e Nelles (2001). Nessas metodologias, a partição também é realizada recursivamente, refinando partições existentes. A topologia do modelo inicia com uma partição única e o refinamento é realizado dividindo as partições recursivamente, dado um critério de particionamento, geralmente associado ao erro do modelo. Esses artigos realizam o refinamento da partição do espaço de entrada utilizando um conjunto de dados de treinamento, isto é, são algoritmos em batelada. A metodologia de partição do espaço de entrada proposta nesse capítulo difere das metodologias similares encontradas na literatura pois é realizada de forma incremental, ou seja, cada amostra é processada apenas uma vez.

A seção 4.2 descreve a árvore de regressão linear nebulosa proposta, descrevendo as diferenças entre esse modelo e sua versão clássica e apresentando um algoritmo de aprendizado incremental. Esse modelo é denominado eFT (*evolving Fuzzy linear regression Tree*). Uma versão simplificada desse modelo foi inicialmente proposta em Lemos et al. (2010b) e depois estendida em Lemos et al. (2011a).

O modelo proposto na seção 4.2 é então estendido na seção 4.3 para realizar seleção de características de forma adaptativa durante o processo de aprendizagem. Esse modelo é denominado xFT (*eXtended evolving Fuzzy linear regression Tree with feature selection*).

## 4.2 Árvores de Regressão Linear Nebulosas Evolutivas

### 4.2.1 Estrutura do Modelo

Árvores de regressão linear são generalizações de árvores de regressão em que cada folha da árvore é associada a um modelo linear, em contrapartida com os valores constantes (isto é,

a média da variável de saída) utilizados nas árvores de regressão. Essas árvores são definidas como estruturas recursivas, capazes de realizar regressão linear por partes, isto é, o espaço de entrada é particionado em regiões disjuntas e um modelo linear é associado a cada uma das regiões. A figura 4.1 ilustra uma árvore de regressão linear.

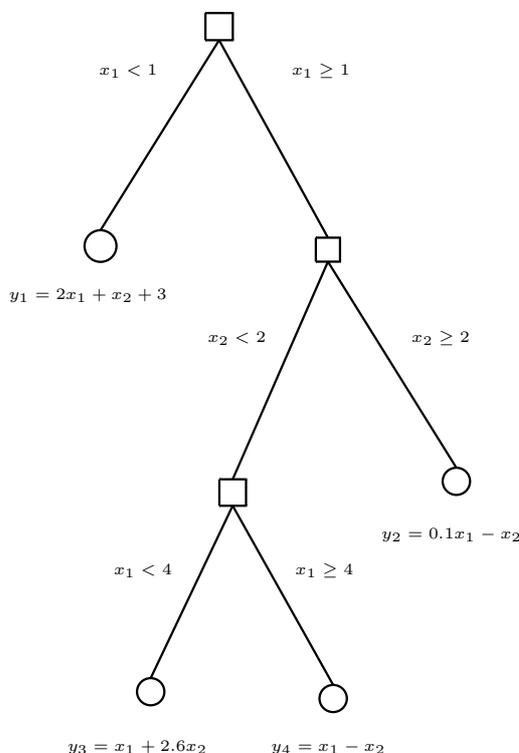


Figura 4.1: Exemplo de uma árvore de regressão linear

Essas árvores são formadas por dois tipos de nós:

- internos - nós que possuem um nó filho, ou seja, são conectados à outro nó de nível inferior;
- folhas - nós que não possuem nós filhos, ou seja, presentes nas extremidades da árvore.

Os nós internos possuem um critério de divisão, ou seja, possuem uma operação do tipo *menor que* e *maior que ou igual a* um determinado valor de corte de uma das variáveis de entrada. Por exemplo, para o nó presente no primeiro nível da árvore ilustrada na figura 4.1 (denominado nó raiz), o valor de corte é 1 e está associado à variável  $x_1$ .

As folhas não possuem critérios de divisão e sim modelos lineares da forma:  $y_i = a_0 + a_1x_1 + \dots + a_mx_m = \sum_{i=1}^m a_ix_i$ , sendo  $m$  o número de variáveis de entrada e  $x_0 = 1$ .

Dado um vetor de entrada, para estimar a saída do modelo, deve-se percorrer a árvore, iniciando pelo nó raiz, até atingir uma folha, avaliando os critérios de divisão presentes nos nós

internos da árvore. A saída do modelo é então estimada utilizando-se os valores de entrada e o modelo linear presente na folha alcançada. Por exemplo, para a árvore ilustrada na figura 4.1, a saída do modelo correspondente às entradas  $x_1 = 4$  e  $x_2 = 3$  é  $-2.6$ .

Cada folha da árvore representa uma região do espaço de entrada delimitada pelos valores de corte presentes nos nós internos percorridos a partir do nó raiz até a folha. Cada região é então modelada pelo modelo linear presente na folha. A figura 4.2 ilustra a partição do espaço de entrada induzida pela árvore de regressão linear ilustrada na figura 4.1 para o intervalo  $x_1 = [0, 6]$  e  $x_2 = [0, 6]$ .

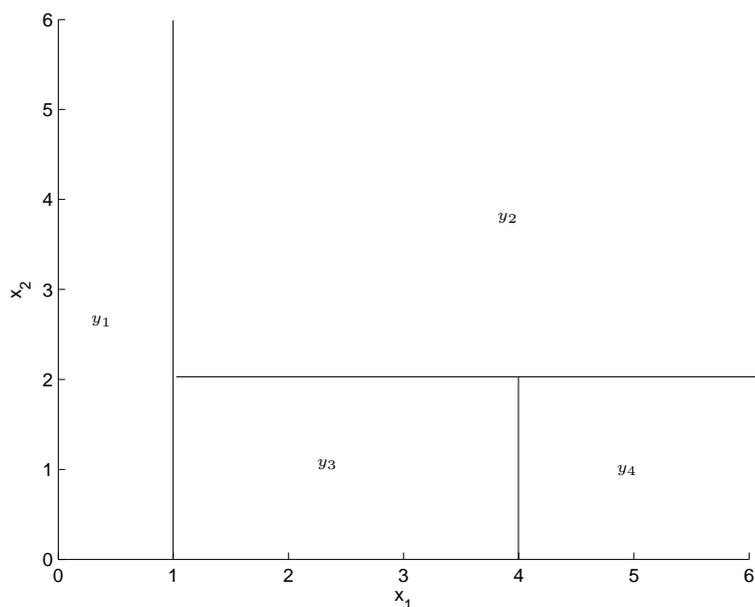


Figura 4.2: Partição do espaço de entrada induzida pela árvore de regressão linear ilustrada na figura 4.1

Existem diversos algoritmos para construção de árvores de regressão linear baseados em treinamento em batelada (Dobra e Gehrke, 2002; Quinlan, 1992; Torgo, 1997) e incremental (Potts, 2004).

Árvores de regressão linear nebulosas substituem os critérios de divisão rígidos, presentes nos nós internos das árvores, por duas funções de pertinência descrevendo os conceitos de *menor que* e *maior que*. Ao introduzir conjuntos nebulosos nos nós internos da árvore, as fronteiras exatas presentes nas árvores originais desaparecem e, para cada vetor de entrada, todos os galhos da árvore são ativados (com diferentes graus de ativação). Essa modificação resulta em partições sobrepostas do espaço de entrada e um modelo de regressão baseado na soma ponderada de modelos lineares locais.

A metodologia proposta nessa seção utiliza funções de pertinência sigmoidais (4.1) nos nós internos da árvore. Funções de pertinência sigmoidais possuem dois parâmetros, o centro  $c$  e o

espalhamento  $\sigma$ . Dependendo do sinal do parâmetro  $\sigma$ , a sigmóide é inerentemente aberta para a direita ou para a esquerda. Assim, essas funções são candidatas naturais para representar os conceitos de *menor que* e *maior que*  $c$ . A figura 4.3 ilustra um exemplo de duas funções de pertinência sigmoidais, descrevendo os conceitos de *menor que 5* e *maior que 5*, assumindo  $c = 5$ ,  $\sigma = -0.5$  (*menor que*) e  $\sigma = 0.5$  (*maior que*).

$$\mu(x) = \frac{1}{1 + \exp -\frac{1}{\sigma}(x - c)} \quad (4.1)$$

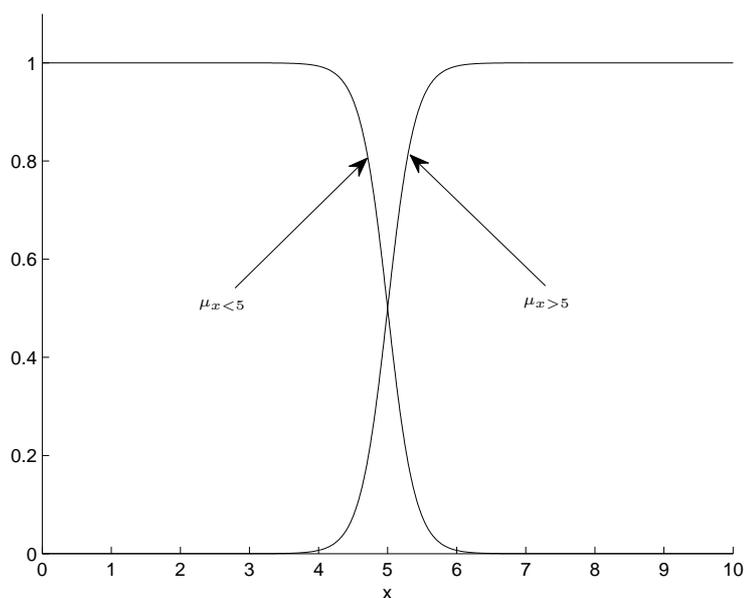


Figura 4.3: Funções de pertinência sigmoidais descrevendo *menor que* and *maior que 5*

Cada folha da árvore está associada a uma região do espaço de entrada e a um modelo linear. Porém, a partição do espaço de entrada é sobreposta, isto é, algumas regiões do espaço estão associadas a mais de uma folha. A figura 4.4 ilustra a partição do espaço de entrada ( $x_1 \in [0, 6]$  e  $x_2 \in [0, 6]$ ) quando os critérios de divisão da árvore presentes na figura 4.1 são substituídos por pares de funções de pertinência sigmoidais, com parâmetros  $c$  iguais aos valores de corte e  $|\sigma| = 0.2$ , para todos os nós internos.

Para calcular a saída correspondente a um dado vetor de entrada, deve-se calcular os valores de pertinência para cada par de funções de pertinência presentes nos nós internos, a partir da raiz até todas as folhas. Em seguida, para todas as folhas, deve-se agregar os valores de pertinência presentes no caminho da raiz até a folha em questão utilizando-se um operador de agregação, como por exemplo uma *t-norma*. O valor de pertinência resultante é então associado à folha em questão, e é denominado como seu grau de ativação. O algoritmo 5 é utilizado para calcular as saídas dos modelos lineares locais e os graus de ativação associados a cada uma

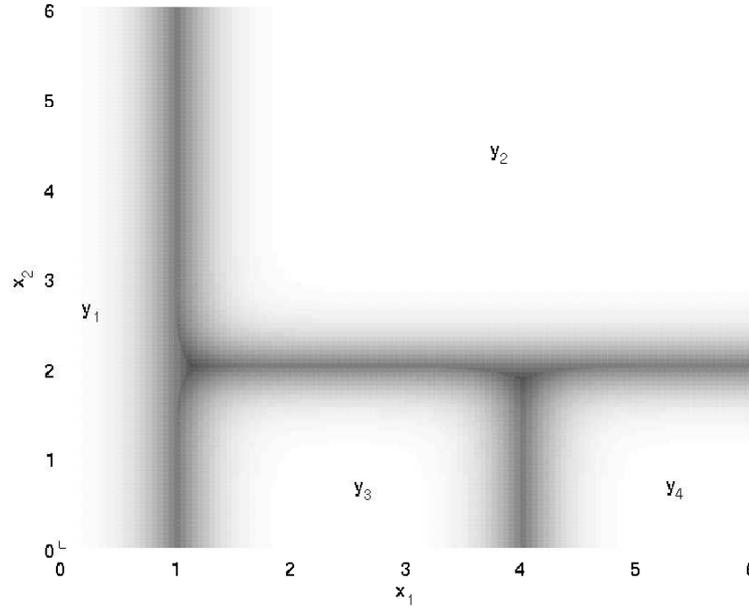


Figura 4.4: Partição do espaço de entrada induzida por uma árvore de regressão linear nebulosa

das folhas da árvore. O algoritmo é iniciado com o parâmetro  $no_i$  igual ao nó raiz da árvore e  $w = 1$ . Esses valores são então utilizados para calcular a saída do modelo, definida como a média ponderada da saída de todos os modelos lineares:

$$\hat{y} = \frac{\sum_{i=1}^{l^k} y_i w_i}{\sum_{i=1}^{l^k} w_i} \quad (4.2)$$

onde  $l^k$  é o número de folhas da árvore e  $w_i$  é o grau de ativação da folha  $i$ , definido como:

$$w_i = T_{j=1}^r \mu_j(x) \quad (4.3)$$

onde  $T$  é uma  $t$ -norma,  $r$  corresponde ao número de nós internos percorridos da raiz até a folha  $i$  e  $\mu_j$  é uma das funções de pertinência sigmoidais associadas com o nó interno  $j$ .

Regras nebulosas funcionais podem ser facilmente extraídas da estrutura da árvore. Cada folha representa uma regra nebulosa cujo antecedente é definido como a agregação de todas as funções de pertinência dos nós internos presentes no caminho iniciado do nó raiz até a folha em questão. Por exemplo, dada uma versão nebulosa da árvore presente na figura 4.1, é possível extrair as seguintes regras:

---

**Algoritmo 5** Algoritmo para calcular os valores de pertinência e a saída dos modelos lineares de todas as folhas da árvore

---

- 1: **if**  $no_i$  é um nó interno **then**
  - 2:   Agrega a função de pertinência da esquerda com o valor de  $w$  utilizando uma  $t$ -norma
  - 3:    $no_i$  = nó raiz da subárvore à esquerda
  - 4:   Aplica esse algoritmo à subárvore da esquerda
  - 5:   Agrega a função de pertinência da direita com o valor de  $w$  utilizando uma  $t$ -norma
  - 6:    $no_i$  = nó raiz da subárvore à direita
  - 7:   Aplica esse algoritmo à subárvore da direita
  - 8: **else**
  - 9:   Armazena o valor do grau de ativação da folha  $i$  ( $w_i$ )
  - 10:   Computa a saída do modelo linear da folha  $i$  ( $y_i$ )
  - 11: **end if**
- 

$$\begin{aligned}
&\text{Se } x_1 < 1 \text{ então } y = 2x_1 + x_2 + 3 \\
&\text{Se } x_1 > 1 \text{ e } x_2 < 2 \text{ e } x_1 < 4 \text{ então } y = x_1 + 2.6x_2 \\
&\text{Se } x_1 > 1 \text{ e } x_2 < 2 \text{ e } x_1 > 4 \text{ então } y = x_1 - x_2 \\
&\text{Se } x_1 > 1 \text{ e } x_2 > 2 \text{ então } y = 0.1x_1 - x_2
\end{aligned} \tag{4.4}$$

onde “ $< c$ ” e “ $> c$ ” são conjuntos nebulosos com funções de pertinência sigmoidais representando os conceitos *menor que c* e *maior que c*.

Além disso, nem todas as regras extraídas possuem conjuntos nebulosos associados a todas as variáveis de entrada em seus antecedentes. Por exemplo, o antecedente da primeira regra só possui um conjunto nebuloso associado a  $x_1$ . Isto ocorre porque o antecedente de cada regra é formado analisando os nós internos presentes no caminho do nó raiz até uma determinada folha, e nem todas as variáveis de entrada estão presentes em todos os caminhos.

É possível reduzir o número de conjuntos nebulosos para cada regra (folha), assumindo que é possível agregar dois ou mais conjuntos de uma mesma regra, associados à mesma entrada em um único conjunto. Por exemplo, os conjuntos nebulosos  $x_1 > 1$  e  $x_1 > 4$  podem ser substituídos por  $x_1 > 4$ , preservando a interpretabilidade da regra. Além disso, os conjuntos  $x_1 > 1$  e  $x_1 < 4$  podem ser vistos como um conjunto nebuloso descrevendo o intervalo  $1 < x_1 < 4$  com função de pertinência  $\mu_{1 < x_1 < 4} = \mu_{x_1 > 1} \tilde{t} \mu_{x_1 < 4}$ . Por exemplo, caso a  $t$ -norma utilizada seja o produto, então a expressão algébrica dessa função de pertinência, dados  $c_1$  e  $c_2$  (com  $c_1 < c_2$ ) e  $\sigma$ , é dada por:

$$\mu_{c_1 < x < c_2} = \frac{1}{1 + \exp -\frac{1}{\sigma}(x - c_1) + \exp \frac{1}{\sigma}(x - c_2) + \exp \frac{1}{\sigma}(c_1 - c_2)} \tag{4.5}$$

Realizando essas técnicas de reduções de conjuntos, o conjunto de regras resultante seria:

$$\begin{aligned}
 &\text{Se } x_1 < 1 \quad \text{então } y = 2x_1 + x_2 + 3 \\
 &\text{Se } x_1 > 4 \text{ e } x_2 < 2 \quad \text{então } y = x_1 - x_2 \\
 &\text{Se } 1 < x_1 < 4 \text{ e } x_2 < 2 \quad \text{então } y = x_1 + 2.6x_2 \\
 &\text{Se } x_1 > 1 \text{ e } x_2 > 2 \quad \text{então } y = 0.1x_1 - x_2
 \end{aligned} \tag{4.6}$$

Claramente, o conjunto de regras (4.6) é muito mais simples que o conjunto original (4.4). A figura 4.5 ilustra as funções de pertinência extraídas da árvore de regressão linear nebulosa utilizada como exemplo.

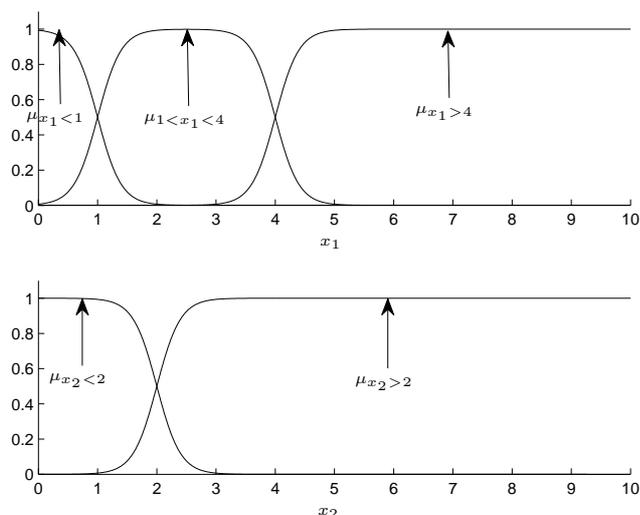


Figura 4.5: Funções de pertinência extraídas da árvore de regressão linear nebulosa

## 4.2.2 Algoritmo de Aprendizado Incremental

Essa seção descreve o algoritmo de aprendizagem incremental proposto para as árvores de regressão linear nebulosas. O algoritmo proposto inicia-se a partir de uma árvore contendo uma única folha e seu modelo linear correspondente, e evolui a árvore substituindo folhas por subárvores com a utilização de um teste de seleção de modelos estatístico e os vetores de entrada. O algoritmo não armazena valores passados, processa cada entrada apenas uma vez e todas as decisões são tomadas com base em estatísticas computadas recursivamente.

Para construir a árvore, assume-se que cada folha possui  $\gamma$  pontos de corte candidatos para cada uma das  $m$  variáveis de entrada, totalizando  $\gamma \times m$  possíveis pontos de corte por folha. Cada ponto de corte candidato é definido como uma subárvore composta por um nó interno, contendo duas funções de pertinência sigmoidais (*menor que* e *maior que*) centradas no valor de corte, seguida por duas folhas, a folha da esquerda e da direita, contendo modelos lineares.

Os pontos de corte candidatos podem ser utilizados para substituir a folha da árvore. A figura 4.6 ilustra um ponto de corte candidato genérico, onde  $x_0 = 1$ .

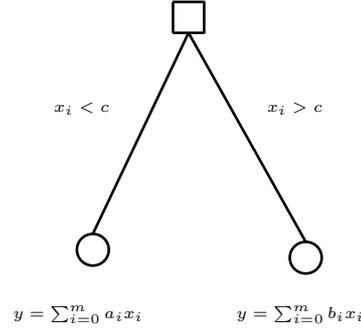


Figura 4.6: Ponto de corte candidato genérico

Conforme discutido anteriormente, cada folha representa uma região do espaço de entrada. Para definir os pontos de corte candidatos, o intervalo de cada variável de entrada é calculado e os pontos de corte são definidos de forma a particionar uniformemente o intervalo em  $\gamma + 1$  subintervalos. Apesar de cada folha definir uma região nebulosa do espaço de entrada, o intervalo utilizado nesse processo é exato e é computado utilizando-se os centros das funções de pertinência dos nós internos alcançáveis da raiz até a folha em questão. Por exemplo, para a árvore ilustrada na seção anterior, a folha associada com o modelo linear  $y_1 = 0.1x_1 - x_2$  é associada ao seguinte intervalo:  $x_1 \in [1, \max(x_1)]$  e  $x_2 \in [2, \max(x_2)]$ , onde  $\max(x_i)$  corresponde ao máximo valor observado para a variável  $i$ . Os valores máximos e mínimos para cada variável são ajustados à medida que novas observações excedam os limites observados.

Para cada nova observação, a saída correspondente é estimada utilizando (4.2) e os modelos lineares associados a cada folha são atualizados utilizando o algoritmo recursivo de mínimos quadrados ponderados (WRLS). Os parâmetros lineares e a matriz  $Q_i$  da fórmula de atualização da folha  $i$  é na iteração  $k$  é definido como:

$$\begin{aligned} a_i^{k+1} &= a_i^k + Q_i^{k+1} x^k \Psi_i(x^k) [y_i^k - ((x^k)^T a_i^k)] \\ Q_i^{k+1} &= Q_i^k - \frac{\Psi_i(x^k) Q_i^k x^k (x^k)^T Q_i^k}{1 + (x^k)^T Q_i^k x^k} \end{aligned} \quad (4.7)$$

onde  $\Psi_i(x^k)$  para  $i = 1, \dots, l^k$  são os graus de ativação normalizados associados com cada modelo linear:

$$\Psi_i(x^k) = \frac{w_i}{\sum_{j=1}^{l^k} w_j} \quad (4.8)$$

onde  $l^k$  é o número de folhas presentes na árvore na iteração  $k$ .

Em seguida, o espalhamento  $\sigma$  das funções de pertinência de todos os nós internos presentes no caminho do nó raiz à folha associada ao maior grau de ativação são atualizados. Para cada função de pertinência presente no caminho, o espalhamento da função sigmoideal é ajustado utilizando o algoritmo do gradiente descendente. Por exemplo, considere uma árvore de regressão linear com topologia descrita pela figura 4.1. Assumindo que, para um dado vetor de entrada, a folha com o maior grau de ativação é a folha associada ao modelo linear  $y_2$ . Nesse caso, as funções de pertinência dos conjuntos nebulosos  $x_1 > 1$  e  $x_2 > 2$  terão seus espalhamentos atualizados.

A atualização dos espalhamentos das funções de pertinência tem como objetivo minimizar uma medida de erro a cada iteração. Essa medida é estimada utilizando a saída do modelo e a saída desejada correspondente:

$$e^k = \frac{1}{2} (\hat{y}^k - y^k)^2 \quad (4.9)$$

A equação recursiva para atualizar os espalhamentos das funções de pertinência presentes nos nós internos é:

$$\sigma_i = \sigma_i - \beta \frac{\partial e^k}{\partial \hat{y}^k} \frac{\partial \hat{y}^k}{\partial \mu_i} \frac{\partial \mu_i}{\partial \sigma_i} \quad (4.10)$$

onde  $\beta$  é a taxa de aprendizado. As derivadas parciais são definidas como:

$$\frac{\partial e^k}{\partial \hat{y}^k} = \hat{y}^k - y^k \quad (4.11)$$

$$\frac{\partial \mu_i}{\partial \sigma_i} = \frac{\left( \exp -\frac{1}{\sigma_i} (x^k - c_i) \right) (x^k - c_i)}{1 + \exp -\frac{1}{\sigma_i} (x^k - c_i)^2 \sigma_i^2} \quad (4.12)$$

Para estimar  $\partial \hat{y}^k / \partial \mu_i$ , devem-se encontrar todos os caminhos entre o nó raiz e as folhas que passem pela função de pertinência  $\mu_i$  e então computar a derivada parcial como:

$$\frac{\partial \hat{y}^k}{\partial \mu_i} = \frac{\sum_{r=1}^{np^k} \frac{w_r}{\mu_i} y_r - \hat{y}^k \sum_{r=1}^{np^k} \frac{w_r}{\mu_i}}{\sum_{j=1}^{lk} w_j} \quad (4.13)$$

onde  $w_r$  é o grau de ativação associado uma folha  $r$  cujo caminho a partir do nó raiz até a folha passa pela função de pertinência  $\mu_i$  e  $np^k$  é o número de folhas que satisfaz essa condição.

A figura 4.7 ilustra os caminhos que passam pela função de pertinência  $\mu_{x_2 < 2}$  para uma árvore de regressão linear nebulosa evolutiva com topologia definida pela figura 4.1. Assim, na atualização dessa função de pertinência, utiliza-se  $w_3$  e  $w_4$  (grau de ativação das folhas associadas aos modelos  $y_3$  e  $y_4$ ) e  $r = 2$ , gerando a seguinte fórmula de atualização:

$$\frac{\partial \hat{y}^k}{\partial \mu_{x_2 < 2}} = \frac{\frac{w_3 y_3 + w_4 y_4}{\mu_{x_2 < 2}} - \hat{y}^k \frac{w_3 + w_4}{\mu_{x_2 < 2}}}{w_1 + w_2 + w_3 + w_4} \quad (4.14)$$

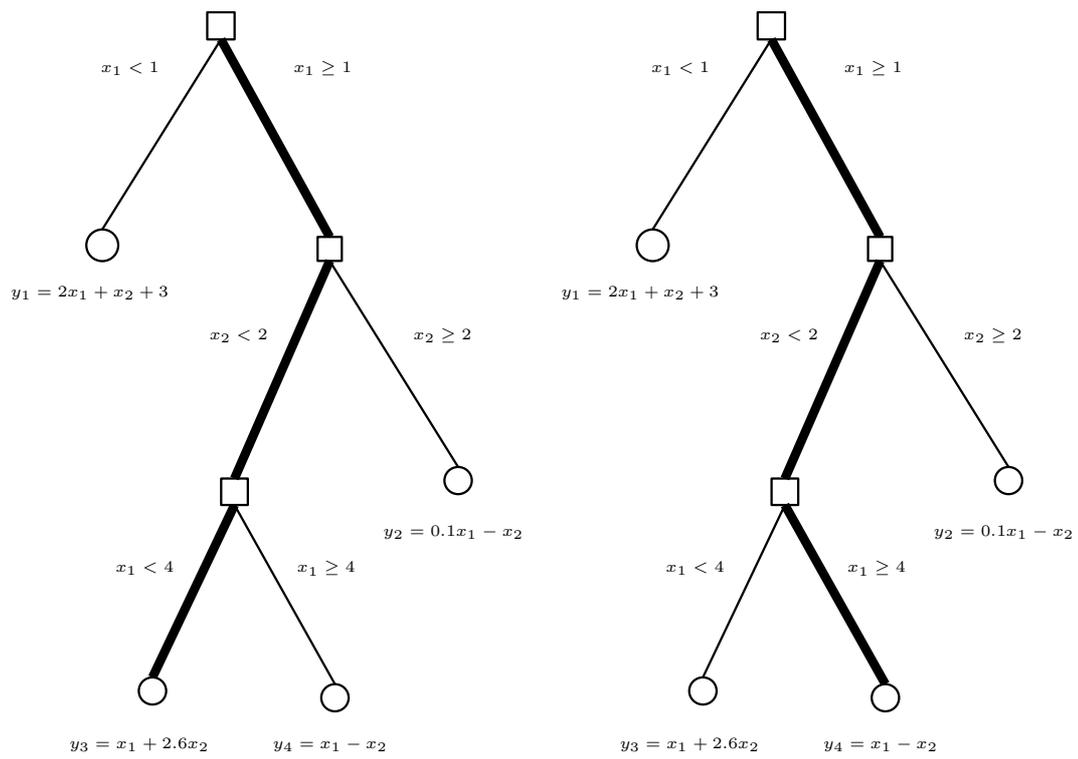


Figura 4.7: Caminhos entre o nó raiz e as folhas que passam pela função de pertinência  $\mu_{x_2 < 2}$

Nota-se que (4.13) só é válida para a  $t$ -norma produto, assim somente essa  $t$ -norma deve ser utilizada pela árvore proposta.

Os modelos lineares de todos os pontos de corte candidatos, associados à folha selecionada, também são atualizados. Para cada ponto de corte candidato, os graus de ativação são revistos, incluindo os valores de pertinência relacionados às funções de pertinência presentes no nó interno do ponto de corte em questão. Em seguida, a saída da árvore resultante (árvore original com a folha selecionada substituída pelo ponto de corte candidato) é computada. Finalmente, os modelos lineares nas folhas do ponto de corte candidato são atualizados, utilizando (4.7). O algoritmo 6 descreve os passos realizados para atualizar os parâmetros da árvore.

---

**Algoritmo 6** Algoritmo para computar os parâmetros dos pontos de corte candidatos e atualizar a árvore

---

- 1: Computa a saída e os graus de ativação de todas as folhas
  - 2: Atualiza os modelos lineares das folhas utilizando WRLS
  - 3: Seleciona a folha associada ao maior grau de ativação
  - 4: Atualiza o espalhamento das funções de pertinência presentes no caminho entre o nó raiz e a folha selecionada
  - 5: **for all** entradas ( $m$ ) **do**
  - 6:   **for all** pontos de corte candidatos ( $\gamma$ ) **do**
  - 7:     Substitui a folha selecionada pela subárvore (ponto de corte candidato)
  - 8:     Computa a saída da árvore resultante
  - 9:     Atualiza os modelos lineares do ponto de corte candidato usando WRLS
  - 10:   **end for**
  - 11: **end for**
- 

Uma vez que a folha correspondente ao maior grau de ativação é selecionada e os parâmetros da árvore são atualizados, testes são realizados para avaliar se alguma das subárvores (pontos de corte candidatos), podem ser utilizadas para substituir a folha em questão. O teste utilizado é um teste da qualidade do ajuste do modelo que leva em conta a acurácia e o número de parâmetros do modelo original e de um modelo estendido. O teste compara a qualidade de dois modelos, um modelo mais simples (a árvore original) e um mais complexo (a árvore original substituindo a folha em questão por uma subárvore), assumindo que o modelo mais simples pode ser aninhado ao modelo mais complexo, e que o modelo mais complexo é mais preciso. O teste tenta responder a seguinte pergunta: o ganho em precisão (medido através dos pontos já observados) vale o *custo* de se adicionar mais parâmetros livres ao modelo (o que pode levar ao sobreajuste)?

O teste original para modelos aninhados (Allen, 1997) assume que os parâmetros dos dois

modelos são estimados utilizando a mesma base de dados e utiliza a seguinte estatística:

$$F = \frac{(RSS_1 - RSS_2) \times (n - p_2)}{RSS_2 \times (p_2 - p_1)} \quad (4.15)$$

onde  $RSS_1$  e  $RSS_2$  correspondem, respectivamente, às somas dos quadrados dos resíduos do modelo mais simples e mais complexo,  $p_1$  e  $p_2$  correspondem ao número de parâmetros dos dois modelos e  $n$  é o tamanho da base de dados utilizada para estimar os parâmetros dos modelos. O número de parâmetros de uma árvore corresponde ao número de parâmetros dos modelos lineares de todas as folhas:

$$p = (m + 1) \times \text{numero de folhas} \quad (4.16)$$

onde  $m$  é a dimensão do espaço de entrada.

Assumindo que a distribuição dos resíduos é normal,  $F$  (4.15) segue uma distribuição de *Fisher* (Papoulis, 1984) com  $(p_2 - p_1, n - p_2)$  graus de liberdade.

Porém, esse teste não pode ser utilizado para evoluir a árvore, porque o número de amostras utilizadas para estimar os parâmetros da árvore pode não ser igual ao número de amostras utilizadas para estimar o número de parâmetros dos pontos de corte candidatos. Em outras palavras, como um novo ponto de corte candidato é criado cada vez que uma nova folha é adicionada à árvore, então o número de amostras utilizadas para estimar os parâmetros da árvore sempre serão iguais ou maiores que o número de amostras utilizadas para os pontos de corte candidatos. Uma modificação do mecanismo de teste para tratar números de amostras distintos é proposta por Potts (2004), que utiliza a seguinte estatística:

$$F_{inc} = \frac{(RSS_1 - RSS_2) \times (n_2 - p_2)}{RSS_2 \times (n_1 - n_2 + p_1)} \quad (4.17)$$

onde  $n_1$  e  $n_2$  correspondem aos números de amostras utilizadas para estimar os parâmetros da árvore e dos pontos de corte candidatos, respectivamente.

$F_{inc}$  segue uma distribuição de *Fisher* com  $(n_2 - n_1 + p_2 - p_1, n_2 - p_2)$  graus de liberdade. Assim, a utilização dessa estatística requer o cálculo dos p-valores (probabilidade na cauda da distribuição) para todos os pontos de corte candidatos da folha correspondente ao maior grau de ativação. O ponto de corte candidato, associado ao menor p-valor, é selecionado. A subárvore presente no ponto de corte candidato selecionado substitui a folha se o p-valor é menor que um nível de significância  $\alpha$ , definido como parâmetro do algoritmo. Porém, é necessário introduzir uma correção de comparações múltiplas, pois a mesma hipótese está sendo testada  $\gamma \times m$  vezes utilizando o mesmo conjunto de dados (Potts, 2004). A correção de Bonferroni deve ser aplicada pela divisão do nível de significância desejado pelo número de testes. Finalmente, a subárvore selecionada substitui a folha se:

$$\text{p-valor} < \frac{\alpha}{\gamma \times m} \quad (4.18)$$

Para utilizar o teste de seleção de modelos descrito, a soma do quadrado dos resíduos e o número de amostras da árvore e de todos os pontos de corte candidatos deve ser atualizada, sempre que os respectivos modelos sejam atualizados. Inicialmente, a saída da árvore é estimada utilizando (4.2). Em seguida, a folha associada ao maior grau de ativação é selecionada e, para todos seus pontos de corte candidatos, a saída é estimada pela substituição da folha pelo ponto de corte candidato. A forma mais simples de realizar essa operação consiste em substituir a folha pelo ponto de corte candidato, e recomputar a saída utilizando o algoritmo 5 e (4.2). Porém, essa operação pode ser otimizada, utilizando as saídas dos modelos lineares já estimadas, os graus de ativação de todas as outras folhas da árvore, e substituindo a saída do modelo linear e o grau de ativação da folha selecionada pelas saídas dos modelos lineares e graus de ativação dos pontos de corte candidatos.

O algoritmo 7 sumariza o processo de evolução da árvore a partir de um fluxo de dados.

---

**Algoritmo 7** Algoritmo de aprendizagem utilizado para evoluir a árvore de regressão linear nebulosa

---

```

1: Computa a saída e o grau de ativação de todas as folhas
2: Atualiza os parâmetros da árvore
3: Seleciona a folha com maior grau de ativação
4: for all Entradas ( $m$ ) do
5:   for all Pontos de corte candidatos ( $\gamma$ ) do
6:     Estima a saída substituindo a folha selecionada pelo o ponto de corte candidato
7:     Computa o p-valor do teste de seleção de modelos para o ponto de corte candidato
8:   end for
9: end for
10: Seleciona o ponto de corte candidato associado ao menor p-valor
11: if p-valor  $< \frac{\alpha}{\gamma \times m}$  then
12:   Substitui a folha selecionada pelo ponto de corte candidato
13: end if

```

---

O algoritmo proposto possui 3 parâmetros:

- o nível de significância  $\alpha$ , utilizado no teste de seleção de modelo;
- o número de pontos de corte candidatos para cada variável,  $\gamma$ ;
- o espalhamento inicial das funções de pertinência sigmoidais,  $\sigma_{init}$ .

O nível de significância é geralmente ajustado para valores típicos, tais como, 0.01 ou 0.05.

O número de pontos de corte candidatos para cada variável deve ser escolhido baseado no compromisso entre a precisão e o custo computacional. Baixos valores de  $\gamma$  podem denigrir a precisão do modelo, já que poucos pontos de corte candidatos serão criados para cada folha. Em contrapartida, altos valores de  $\gamma$  podem aumentar a precisão do modelo, mas também aumentam o número de modelos lineares a serem atualizados.

Finalmente, o espalhamento inicial das funções de pertinência sigmoidais deve ser escolhido, baseado em conhecimento *a priori* sobre a escala dos dados. A seleção apropriada desse parâmetro aumenta a velocidade de convergência para os valores ótimos.

O algoritmo também necessita de informação inicial sobre os intervalos das variáveis de entrada, para criar os pontos de corte candidatos iniciais. Esse intervalo pode ser calculado a partir de um conjunto de dados iniciais. O algoritmo atualiza o intervalo à medida que seja necessário.

A complexidade temporal desse modelo, para avaliar uma amostra, é  $O(l^k + s^k)$ , sendo  $l^k$  o número de folhas e  $s^k$  o número de nós internos na iteração  $k$ . Para cada amostra de entrada, todos os caminhos, do nó raiz até todas as folhas, devem ser percorridos para estimar os graus de ativação e as saídas dos modelos de cada folha.

Essa ordem de complexidade é, geralmente, maior que a de modelos nebulosos evolutivos baseados em algoritmos de agrupamento. Conforme discutido anteriormente, para esses modelos, a ordem de complexidade é geralmente  $O(g^k)$ , sendo  $g^k$  o número de regras na iteração  $k$ . Nota-se que  $g^k$  é igual a  $l^k$ , já que cada folha da árvore pode ser interpretada como uma regra nebulosa.

Para atualizar os parâmetros da rede, primeiramente os modelos lineares de todas as folhas e pontos de corte candidatos associados a folha com maior grau de ativação, devem ser atualizados. Em seguida, todos pontos de corte atualizados devem ser avaliados como possíveis substitutos da folha relacionada. A complexidade temporal necessária para atualizar os modelos lineares é  $O(l^k)$ , e a complexidade para atualizar e testar os pontos de corte candidatos é  $O(\gamma \times m)$ . Assim, a complexidade temporal para atualizar o modelo é  $O(\max(l^k, \gamma \times m))$ .

Caso  $l^k > \gamma \times m$ , a complexidade temporal para atualizar o modelo é similar a típicos modelos nebulosos baseados em agrupamento evolutivo. Caso contrário, a complexidade é superior.

### 4.3 Árvores de Regressão Linear Nebulosas com Seleção de Características

O modelo proposto nessa seção é uma variação do modelo descrito na seção 4.2, que é capaz de realizar seleção de características durante o processo de aprendizagem incremental. Esse modelo aborda uma deficiência presente na maioria dos modelos nebulosos evolutivos. Geralmente, esses modelos tratam o problema de modelagem de sistemas não estacionários através de uma estrutura flexível, isto é, capaz de se modificar para lidar com variações no

sistema modelado ao longo do tempo. Porém, essas abordagens ainda possuem uma limitação na flexibilidade do modelo: a definição das variáveis de entrada. Geralmente são definidas antes do processo de aprendizagem e então fixas.

Por exemplo, um modelo de previsão de séries temporais evolutivo, que utiliza valores atrasados da série como entradas. Geralmente, os atrasos utilizados são definidos antes do início do aprendizado, baseando-se, por exemplo, em técnicas de seleção de características. O mesmo ocorre com classificadores evolutivos, em que as entradas são definidas e permanecem fixas antes do início do processo de aprendizagem (Angelov e Zhou, 2008; Lughofer et al., 2007).

Um recente desafio para modelos evolutivos consiste em definir métodos de seleção/extração de características incorporados na metodologia de aprendizagem evolutiva, para tratar mudanças nas características do processo, diferentes modos de operação, etc (Lughofer, 2011). Apesar de já existirem métodos de seleção/extração de características incrementais (Li et al., 2003), o desafio consiste em integrar essa informação variante no tempo em um modelo evolutivo, sem causar descontinuidades no aprendizado (Lughofer, 2011). Ou seja, o processo de seleção/extração de características deve ser parte do processo de aprendizagem.

Trabalhos recentes abordam essa deficiência para problemas de classificação de padrões, com a aplicação de metodologias de extração de características incorporadas em classificadores. Por exemplo, Lughofer (2011) propõe uma metodologia de extração de características incremental aplicada no processo de aprendizagem do classificador nebuloso evolutivo FLEXFIS-Class (Lughofer et al., 2007). Cada variável de entrada está associada a um peso definido no intervalo  $[0, 1]$ . Variáveis importantes para discriminação entre duas ou mais classes são associadas a pesos próximos de 1, e variáveis irrelevantes, a pesos próximos de 0. Esses pesos são permanentemente atualizados durante o processo de aprendizagem. Quando os pesos se aproximam de 0, as variáveis relacionadas não causam impacto na aprendizagem.

O modelo proposto nessa seção é visto como uma variação da árvore de regressão nebulosa evolutiva proposta na seção anterior, em que uma metodologia de seleção de características é incorporada no processo de aprendizagem. Nessa variação da árvore, os modelos lineares presentes nas folhas não são funções de todas as variáveis de entrada. A árvore inicia com apenas uma folha, contendo um modelo linear de uma das variáveis de entrada e um teste de significância da regressão é utilizado, juntamente com o teste de seleção de modelos descrito na seção 4.2.2, para crescer a árvore e adicionar variáveis de entrada relevantes ao problema de regressão a ser tratado.

### 4.3.1 Estrutura do Modelo

Conforme descrito anteriormente, a árvore de regressão linear nebulosa proposta nessa seção não utiliza todas as variáveis de entrada nos modelos lineares presentes nas folhas. Apenas as variáveis presentes nos valores de corte dos nós internos da árvore, encontrados no caminho entre o nó raiz e uma determinada folha, fazem parte do modelo linear correspondente. A figura 4.8 ilustra um exemplo da árvore de regressão linear nebulosa modificada. Nesse exemplo, o modelo  $y_1$  é uma função apenas da variável  $x_1$ , uma vez que, apenas  $x_1$  está presente nos nós internos do caminho correspondente.

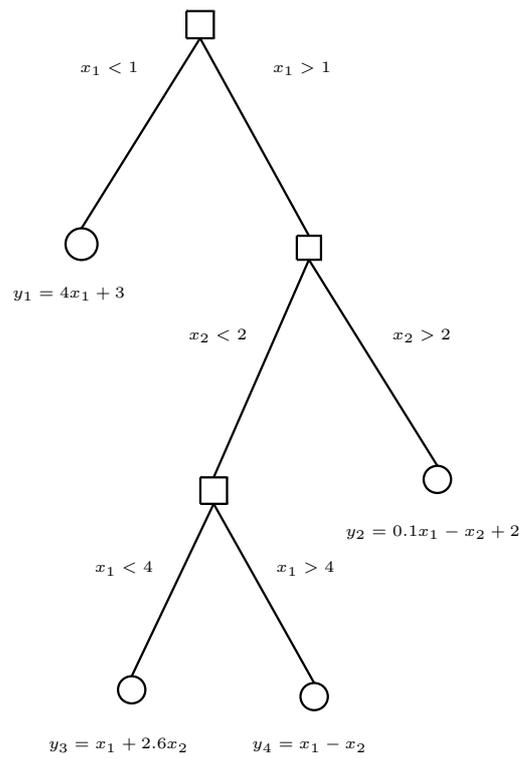


Figura 4.8: Árvore de regressão linear nebulosa com seleção de características

Ao definir os modelos lineares com um número restrito de entradas em cada folha, permite-se que a árvore realize uma seleção de características adaptativa durante o processo de aprendizagem. A cada iteração, o modelo de regressão gerado pela árvore é função apenas das variáveis associadas às funções de pertinência do nós internos da árvore. Por exemplo, se apenas  $x_1$  fosse utilizado em todos os nós internos da árvore da figura 4.8, a árvore seria função apenas dessa entrada. Essa modificação pode também reduzir o número de parâmetros livres da árvore, resultando em modelos mais compactos.

O processo de estimação da saída da árvore é similar ao discutido para sua versão original, na seção 4.2.1. Assim como a partição do espaço de entrada e a extração de regras nebulosas funcionais.

### 4.3.2 Algoritmo de Aprendizado Incremental

Essa seção descreve o algoritmo de aprendizado incremental proposto, capaz de adaptar a topologia da árvore e realizar seleção de características adaptativas a partir de um fluxo de dados.

Assim como o algoritmo descrito na seção 4.2.2, o algoritmo proposto nessa seção inicia o treinamento a partir de uma árvore composta por apenas uma folha, e cresce a árvore substituindo folhas por subárvores, utilizando testes estatísticos e os dados de entrada.

Para realizar seleção de características, um conjunto de possíveis variáveis deve ser definido. O modelo linear da folha inicial pode ser uma função de apenas uma variável ou de um conjunto inicial de variáveis, escolhidas a partir de conhecimento *a priori* sobre o sistema a ser modelado. Caso nenhuma informação inicial esteja disponível, apenas uma variável é selecionada, utilizando qualquer método de ranqueamento de características (Guyon e Elisseeff, 2003) aplicado a todas as possíveis variáveis.

Assim como o algoritmo proposto na seção 4.2.2, são definidos  $\gamma \times m$  pontos de corte candidatos (subárvores) para cada folha da árvore, sendo  $m$  (nesse caso) o número de possíveis variáveis de entrada. O nó interno de cada subárvore possui funções de pertinência centradas em um valor de corte de uma das possíveis variáveis de entrada do problema. Os modelos lineares, presentes nas folhas da subárvore, são funções da variável utilizada no valor de corte, e todas outras variáveis presentes nos nós internos da árvore encontrados no caminho do nó raiz até a a folha em questão. Assim, em cada modelo linear das folhas da subárvore, o número de variáveis de entrada utilizadas será igual ao número de variáveis presentes no caminho correspondente à folha mais uma (a variável utilizada no valor de corte); ou simplesmente ao número de variáveis presentes no caminho correspondente a folha (caso a variável utilizada no valor de corte já seja utilizada em um dos nós internos do caminho correspondente a folha).

A figura 4.9(a) ilustra um ponto de corte candidato para a folha associada ao modelo  $y_1$  da árvore, com topologia ilustrada pela figura 4.8. Nesse exemplo, a variável utilizada no nó interno do ponto de corte candidato é  $x_2$ . A figura 4.9(b) ilustra outro exemplo de um ponto de corte para a mesma folha, com  $x_1$  no nó interno. Analisando essa figura, percebe-se que, no primeiro exemplo, são utilizadas duas variáveis nos modelos lineares das folhas do ponto de corte candidato,  $x_1$  e  $x_2$ , já que  $x_1$  está presente nos nós internos da árvore no caminho até a

folha em questão e  $x_2$  é utilizada no nó interno do ponto de corte candidato. Já no segundo exemplo, apenas a variável  $x_1$  é utilizada.

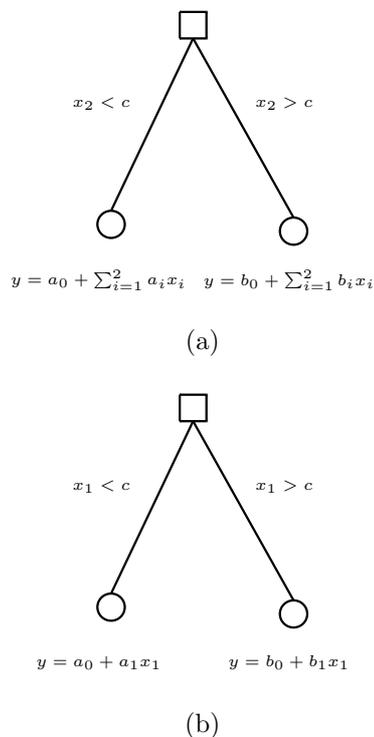


Figura 4.9: Exemplos de pontos de corte candidatos

O processo de atualização dos parâmetros da árvore (modelos lineares e espalhamento das funções de pertinência) é realizado de forma análoga ao algoritmo proposto na seção 4.2.2.

Uma vez realizada a atualização dos parâmetros, e selecionada a folha associada ao maior grau de ativação, dois testes são realizados com o intuito de crescer a árvore e adicionar variáveis relevantes ao modelo. O primeiro teste sempre é executado e corresponde ao teste de seleção de modelos descrito na seção 4.2.2. Porém o número de parâmetros presentes nos dois modelos não podem ser computados utilizando (4.16), uma vez que cada modelo linear das folhas pode ter um número de parâmetros diferente.

O segundo teste é aplicado a todos os pontos de corte candidatos que satisfaçam (4.18) e contenham novas variáveis no nó interno, ou seja, que contenham uma variável no nó interno (do ponto de corte candidato) que não esteja presente nos nós internos da árvore no caminho do nó raiz até a folha em questão. Esse teste é realizado para verificar a significância do coeficiente de regressão associado à nova variável. As hipóteses do teste são:

$$\begin{aligned}
 H_0 & : a_i = 0 \\
 H_1 & : a_i \neq 0
 \end{aligned}
 \tag{4.19}$$

onde  $a_i$  é o coeficiente de regressão associado à nova variável nos modelos lineares das folhas do ponto de corte candidato.

O teste verifica se a variável a ser adicionada é importante para o modelo de regressão, uma vez que adicionar variáveis relevantes aumenta a qualidade do modelo, enquanto adicionar variáveis irrelevantes pode levar ao sobreajuste.

O teste utiliza a seguinte estatística (Allen, 1997):

$$T_{inc} = \frac{a_i}{SE(a_i)} \quad (4.20)$$

onde  $SE(a_i)$  é o erro padrão de  $a_i$  e é definido como:

$$SE(a_i) = MSE_i Q_i^k(i, i) \quad (4.21)$$

onde  $Q_i^k(i, i)$  é o valor da diagonal da matriz  $Q_i^k$  associado ao coeficiente  $a_i$  e  $MSE$  é dado por:

$$MSE_i = \frac{RSS_2}{n_i} \quad (4.22)$$

sendo  $n_i$  o número de amostras utilizadas para estimar  $a_i$ .

$T_{inc}$  é distribuída de acordo com uma distribuição t de Student (Papoulis, 1984) com  $n_i - p_i$  graus de liberdade (Allen, 1997), sendo  $p_i$  o número de parâmetros presentes no modelo linear. O p-valor é então calculado e a hipótese nula é rejeitada se:

$$\text{p-valor}_2 < \alpha \quad (4.23)$$

Finalmente, um ponto de corte candidato pode substituir a folha relacionada se satisfizer (4.18) e não for associado a uma nova variável de entrada; ou se satisfizer (4.18) e (4.23) para os dois modelos lineares do ponto de corte candidato. Caso mais de um ponto de corte seja elegível para substituir a folha correspondente, então o ponto de corte associado ao menor valor de (4.18) é escolhido. O algoritmo 8 sumariza o processo de aprendizagem proposto para esse modelo.

A complexidade temporal desse modelo para avaliar uma entrada é similar ao modelo eFT.

Para atualizar o modelo, o número de pontos de corte candidatos é igual à  $\gamma \times m$ , sendo  $m$ , nesse caso, o número de possíveis variáveis de entrada. Portanto, a ordem de complexidade desse modelo é definida como  $O(\max(l^k, \gamma \times m))$ , que, em geral, é maior que a do eFT, uma vez que, geralmente, o processo de seleção de características é realizado em um conjunto mais amplo de variáveis.

---

**Algoritmo 8** Algoritmo de aprendizagem utilizado para evoluir a árvore de regressão linear nebulosa com seleção de características

---

- 1: Computa a saída e o grau de ativação de todas as folhas
- 2: Atualiza os parâmetros da árvore
- 3: Seleciona a folha com maior grau de ativação
- 4: **for all** Entradas ( $m$ ) **do**
- 5:   **for all** Pontos de corte candidatos ( $\gamma$ ) **do**
- 6:     Estima a saída substituindo a folha selecionada pelo o ponto de corte candidato
- 7:     Computa o p-valor do teste de seleção de modelos para o ponto de corte candidato
- 8:   **end for**
- 9: **end for**
- 10: Seleciona todos os pontos de corte candidatos cujo p-valor  $< \frac{\alpha}{\gamma \times m}$
- 11: Ordena os pontos de corte de selecionados de acordo com o p-valor
- 12: **for all** Pontos de corte selecionados **do**
- 13:   **if** Ponto de corte está relacionado com uma variável que ainda não está presente na árvore **then**
- 14:     Computa o p-valor<sub>2</sub> do teste de significância do parâmetro da regressão
- 15:     **if** p-valor<sub>2</sub>  $< \alpha$  **then**
- 16:       Substitui a folha selecionada pelo ponto de corte candidato (uma nova variável será adicionada ao modelo)
- 17:       Sai do laço
- 18:     **end if**
- 19:   **else**
- 20:     Substitui a folha selecionada pelo ponto de corte candidato
- 21:     Sai do laço
- 22:   **end if**
- 23: **end for**

---

# Capítulo 5

## Resultados Numéricos

### 5.1 Introdução

Esse capítulo descreve os experimentos realizados com o objetivo de avaliar os modelos propostos nos capítulos anteriores. Os modelos são avaliados em problemas de regressão (previsão de séries temporais e identificação de sistemas), seleção de características e detecção e diagnóstico adaptativo de falhas em sistemas dinâmicos.

A seção 5.2 descreve os resultados da avaliação dos modelos nebulosos funcionais eMG e eFT para problemas de previsão de séries temporais e modelagem de sistemas dinâmicos.

A seção 5.3 descreve resultados experimentais relativos à avaliação do desempenho do modelo nebuloso evolutivo xFT, definido como uma extensão do modelo eFT, em problemas de regressão, em que o modelo realiza seleção de características durante o processo de aprendizagem.

A seção 5.4 descreve os resultados da avaliação do classificador nebuloso evolutivo proposto no capítulo 3.2 em problemas de detecção e diagnóstico adaptativo de falhas.

### 5.2 Problemas de Regressão

Nessa seção, os modelos eMG (capítulo 3 seção 3.4) e eFT (capítulo 4 seção 4.2) propostos são avaliados em problemas de previsão de séries temporais e modelagem de sistemas dinâmicos.

Os resultados obtidos para os problemas dessa seção foram comparados com modelos nebulosos evolutivos alternativos consagrados na literatura. Os modelos utilizados na comparação foram: DENFIS (Kasabov e Song, 2002), eTS (Angelov e Filev, 2004) e xTS (Angelov e Zhou, 2006).

Na maior parte dos experimentos, os parâmetros de todos os métodos utilizados foram selecionados de forma a gerar modelos com estrutura semelhante, isto é, número similar de regras (ou folhas). O modelo xTS possui apenas 1 parâmetro que não influencia no número de

regras geradas. Assim, para todos os experimentos, os parâmetros dos modelos foram ajustados para gerar um número de regras semelhantes ao número gerado pelo xTS. Porém, em alguns dos experimentos, alguns métodos não ofereceram flexibilidade paramétrica suficiente para gerar um modelo com estrutura exatamente similar.

O KEDRI <sup>1</sup> (*Knowledge Engineering and Discovery Research Institute*) disponibiliza uma versão desenvolvida em Matlab para o modelo DENFIS. O código em Java referente aos modelos xTS e eTS foram disponibilizados pelos respectivos autores.

Em todos os experimentos dessa seção, o desempenho dos modelos foi avaliado utilizando-se a raiz do erro quadrático médio RMSE <sup>2</sup>e/ou o índice de erro não dimensional NDEI<sup>3</sup>. O NDEI é uma medida de erro independente da escala dos dados, definida como a razão entre o RMSE e o desvio padrão da saída desejada. As medidas de erro utilizadas são definidas como:

$$RMSE = \left( \frac{1}{n} \sum_{k=1}^n (y^k - \hat{y}^k) \right)^{\frac{1}{2}} \quad (5.1)$$

$$NDEI = \frac{RMSE}{std(y^k)} \quad (5.2)$$

onde  $n$  é o tamanho do conjunto de teste utilizado,  $y^k$  é a saída esperada,  $\hat{y}^k$  a saída do modelo e  $std()$  o desvio padrão.

Essas medidas de erro são bons índices para se medir a acurácia dos modelos. Porém, não revelam se o resultado obtido por um modelo é estatisticamente superior ao de qualquer outro. Dessa forma, um teste estatístico foi utilizado para comparar a acurácia dos modelos.

O teste *MGN* (Diebold e Mariano, 1995) é um teste paramétrico que compara a acurácia de dois modelos de previsão. A estatística do teste é definida como:

$$MGN = \frac{\hat{\rho}_{sd}}{\sqrt{\frac{1 - \hat{\rho}_{sd}^2}{n-1}}} \quad (5.3)$$

onde  $\hat{\rho}_{sd}$  é o coeficiente de correlação amostral entre  $s = r_1 + r_2$  e  $d = r_1 - r_2$ , sendo  $r_1$  e  $r_2$  os resíduos dos dois modelos de previsão comparados. Nesse caso, a estatística segue uma distribuição t de *Student* com  $n - 1$  graus de liberdade. Para esse teste, caso os dois modelos sejam igualmente precisos, a correlação entre  $s$  e  $d$  será zero.

### 5.2.1 Previsão de Carga

Uma etapa importante do processo de operação de um sistema de energia elétrica consiste na determinação da programação de operação diária, que define um plano de produção de energia para o(s) próximo(s) dia(s) para cada uma das unidades geradoras do sistema, geralmente em

---

<sup>1</sup><http://www.aut.ac.nz/research/research-institutes/kedri/books>

<sup>2</sup>do inglês *root mean square error*

<sup>3</sup>do inglês *non-dimensional error index*

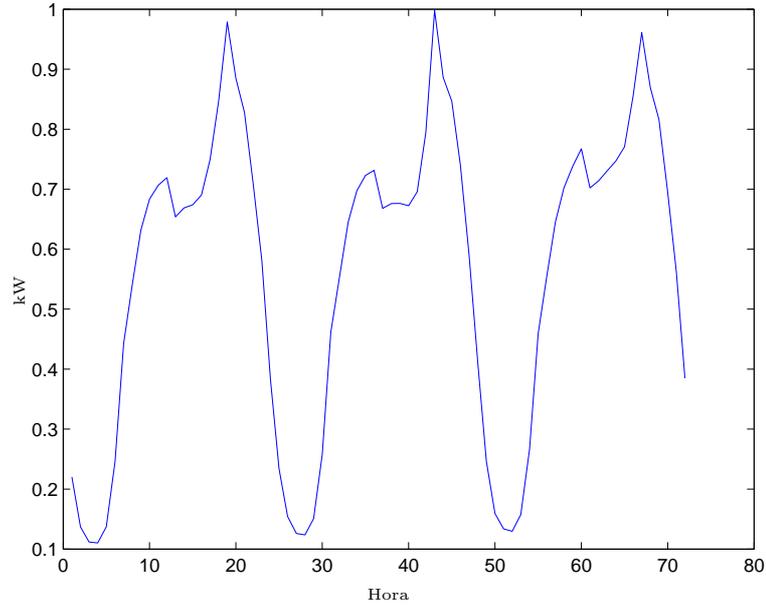


Figura 5.1: Valores normalizados de carga (Kw/Hora) correspondentes aos 3 primeiros dias de Agosto de 2000

base horária. Essa operação é baseada em previsões de demanda do sistema a partir de modelos construídos utilizando informações prévias.

Especificamente, no contexto do planejamento da operação de curto prazo de sistemas de energia elétrica, a previsão de carga tem um papel importante na elaboração do programa de operação diária, em análises de segurança e estabilidade, uma vez que erros de previsão podem gerar sérios problemas em termos de eficiência e segurança do sistema, tais como aumento de custos e não atendimento de demanda. Dessa forma, um modelo de previsão confiável e preciso é essencial, pois melhora o atendimento da demanda, aumentando a eficiência e confiabilidade do sistema (Gross e Galiana, 1987).

Essa seção descreve os resultados da utilização dos modelos funcionais evolutivos propostos na previsão de carga a curto prazo. A base de dados utilizada é composta por medições horárias do consumo de energia em uma região residencial da cidade de São Paulo, fornecida pela empresa Eletropaulo Metropolitana Eletricidade de São Paulo S.A.. Os dados utilizados são expressos em kilowatts (kW) e correspondem a 31 dias do mês de Agosto de 2000. A figura 5.1 ilustra os valores normalizados dos primeiros 3 dias.

Inicialmente, as amostras foram normalizadas no intervalo  $[0, 1]$  da seguinte forma:

$$\hat{x}^k = \frac{x^k - \min(x^k)}{\max(x^k) - \min(x^k)} \quad (5.4)$$

Embora os dois modelos propostos sejam independentes de escala, a normalização foi reali-

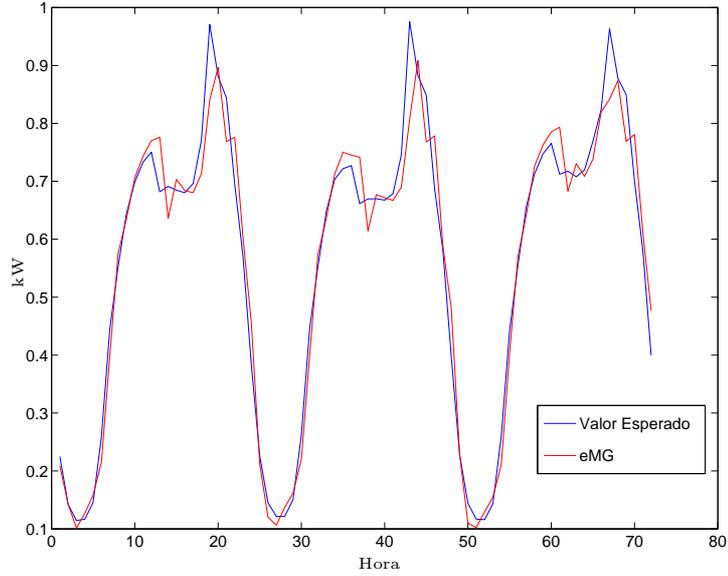


Figura 5.2: Resultado da previsão de carga para o eMG

zada para manter o sigilo dos dados fornecidos.

Os modelos nebulosos evolutivos realizam a previsão um passo a frente, baseado em valores atrasados da série. A função de autocorrelação amostral (Box e Jenkins, 1990) das primeiras 36 amostras sugere a utilização dos dois últimos valores como entradas do modelo, ou seja, o modelo de previsão possui a seguinte estrutura de entradas e saída:

$$\hat{y}^k = f(y^{k-1}, y^{k-2}) \quad (5.5)$$

Os experimentos foram realizados da seguinte forma: as amostras correspondentes aos primeiros 28 dias foram utilizados para o treinamento do modelo e as amostras remanescentes (últimos 3 dias) foram utilizadas para avaliar o seu desempenho, mantendo sua estrutura e parâmetros fixos, de acordo com os valores obtidos ao final dos 28 dias. Dessa forma, a série de carga composta por 744 amostras foi dividida em um conjunto de treinamento, composto por 672 amostras e um conjunto de teste, contendo 72 amostras.

Os valores esperados e previstos para o período de teste são ilustrados nas figuras 5.2 e 5.3 para os modelos eMG e eFT respectivamente. O modelo eMG foi iniciado a partir da primeira observação e seus parâmetros foram ajustados para:  $\alpha = 0.05$ ,  $w = 20$ ,  $\Sigma_{init} = 10^{-2}I_2$  e  $\beta = 0.01$ . O modelo eFT também foi iniciado a partir da primeira observação e seus parâmetros foram ajustados para:  $\alpha = 0.05$ ,  $\sigma_{init} = 0.01$ ,  $\gamma = 20$ , e  $\beta = 0.01$ .

A figura 5.4 ilustra os grupos resultantes do modelo eMG. Esse modelo é formado pelas seguintes regras, com parâmetros dos antecedentes definidos pelos parâmetros dos grupos correspondentes:

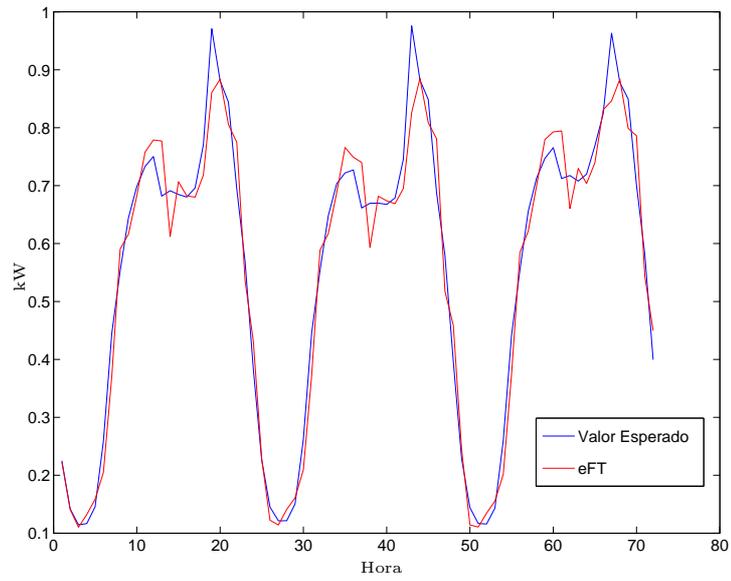


Figura 5.3: Resultado da previsão de carga para o eFT

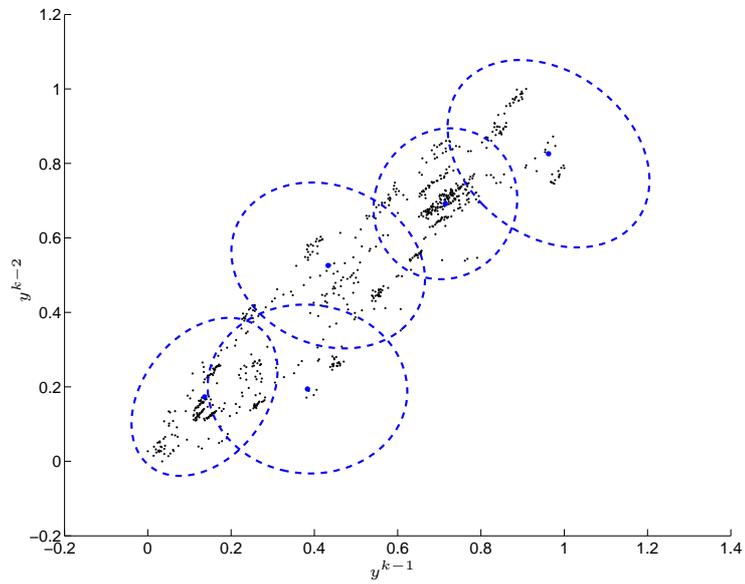


Figura 5.4: Estrutura de grupos do espaço de entrada gerados pelo modelo eMG para o problema de previsão de carga

$$\begin{aligned}
\text{Se } x^k \text{ é } A_{[0.1363 \ 0.1731]} \text{ então } y_1 &= 0.0361 + 2.0103y^{k-1} - 0.9894y^{k-2} \\
\text{Se } x^k \text{ é } A_{[0.9623 \ 0.8260]} \text{ então } y_2 &= 0.4985 + 0.6757y^{k-1} - 0.3350y^{k-2} \\
\text{Se } x^k \text{ é } A_{[0.4331 \ 0.5260]} \text{ então } y_3 &= 0.0756 + 1.7899y^{k-1} - 0.9618y^{k-2} \\
\text{Se } x^k \text{ é } A_{[0.7146 \ 0.6916]} \text{ então } y_4 &= -0.1018 + 1.8821y^{k-1} - 0.7274y^{k-2} \\
\text{Se } x^k \text{ é } A_{[0.3834 \ 0.1942]} \text{ então } y_5 &= 0.1334 + 1.5142y^{k-1} - 0.9020y^{k-2}
\end{aligned} \tag{5.6}$$

onde  $x^k$  é o vetor de entrada, ou seja,  $x^k = [y^{k-1} \ y^{k-2}]^T$ ; e  $A_{[x \ y]}$  representa um conjunto nebuloso Gaussiano multivariável centrado em  $[x \ y]^T$ .

A figura 5.5 ilustra a árvore resultante do modelo eFT. As seguintes regras podem ser extraídas dessa árvore.

$$\begin{aligned}
\text{Se } y^{k-1} < 0.8095 \text{ e } y^{k-2} < 0.6190 \text{ então } y_1 &= 0.0574 + 1.6736y^{k-1} - 0.8215y^{k-2} \\
\text{Se } y^{k-1} < 0.8095 \text{ e } y^{k-2} > 0.6190 \text{ então } y_2 &= -0.0329 + 2.1482y^{k-1} - 1.0931y^{k-2} \\
\text{Se } 0.8095 < y^{k-1} < 0.8730 \text{ então } y_3 &= 0.0608 + 1.7244y^{k-1} - 0.8448y^{k-2} \\
\text{Se } y^{k-1} > 0.8730 \text{ então } y_4 &= -0.1242 + 0.9342y^{k-1} + 0.1310y^{k-2}
\end{aligned} \tag{5.7}$$

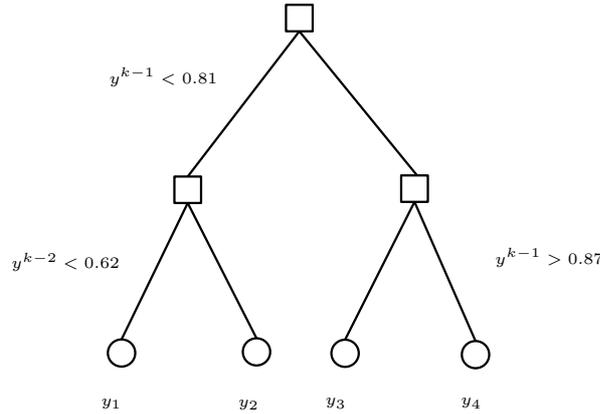


Figura 5.5: Árvore de regressão linear nebulosa evolutiva para o problema de previsão de carga

Os resultados obtidos foram comparados com outros modelos funcionais evolutivos. A tabela 5.1 descreve os resultados comparativos representados através do número de regras geradas e das medidas de erro RMSE e NDEI. Os parâmetros do modelo eTS foram ajustados para  $r = 0.5$  e  $\omega = 750$ . O modelo xTS tem  $\omega = 750$ . O modelo DENFIS utilizou o limiar de distância igual à 0.18. Essa tabela sugere que o modelo eFT possui o melhor desempenho dentre os modelos avaliados.

Comparando os resultados obtidos pelo modelo eMG com os resultados dos modelos eTS, xTS e DENFIS, pode-se especular que a utilização de informação sobre a interação entre as variáveis de entrada, presentes na matriz de dispersão dos conjuntos nebulosos multivariáveis, contribuem para o desempenho superior desse modelo. Já os bons resultados obtidos pelo modelo eFT podem ser atribuídos, à metodologia utilizada na definição de sua estrutura, baseada na qualidade do modelo resultante.

Tabela 5.1: Desempenho comparativo para a previsão de carga

Nome do modelo	Número de regras	RMSE	NDEI
DENFIS	5	0.0665	0.2568
xTS	4	0.0634	0.2447
eTS	5	0.0584	0.2254
eMG	5	0.0499	0.1929
eFT	4	0.0496	0.1916

A tabela 5.2 descreve comparações pareadas entre os resultados dos modelos utilizando o teste *MGN* (5.3). Cada linha dessa tabela descreve os resultados do teste realizado para avaliar se o desempenho do primeiro modelo é superior ao do segundo. Por exemplo, a primeira linha avalia se o desempenho do eMG é superior ao DENFIS. A tabela mostra a estatística *MGN* e o p-valor relacionado. Analisando essa tabela, percebe-se que os modelos eFT e eMG apresentam desempenho estatisticamente superior aos modelos xTS, eTS e DENFIS, para um nível de significância 0.05. Além disso, a última linha da tabela mostra que, apesar das medidas de erro serem menores para o modelo eFT, os dois modelos propostos nesse trabalho possuem desempenho similar para esse problema.

Tabela 5.2: Avaliação do teste *MGN* para o problema de previsão de carga

Modelos	<i>MGN</i>	p-valor
eMG vs DENFIS	4.0965	0.0001
eMG vs eTS	3.1094	0.0013
eMG vs xTS	4.2864	0.0000
eFT vs DENFIS	3.8964	0.0001
eFT vs eTS	3.3938	0.0006
eFT vs xTS	2.1605	0.0171
eFT vs eMG	0.2705	0.3938

## 5.2.2 Identificação de Sistemas Não Lineares

Nessa seção, os modelos nebulosos funcionais evolutivos propostos nesse trabalho são avaliados em um problema clássico de identificação de sistemas dinâmicos não lineares.

O sistema não linear a ser identificado é definido como:

$$y^k = \frac{y^{k-1}y^{k-2}(y^{k-1} - 0.5)}{1 + (y^{k-1})^2 + (y^{k-2})^2} + u^{k-1} \quad (5.8)$$

sendo  $y^0 = y^1 = 0$ .

O objetivo é prever a saída atual do sistema baseado em valores atrasados da entrada e da saída. Os modelos são da forma:

$$\hat{y}^k = f(u^{k-1}, y^{k-1}, y^{k-2}) \quad (5.9)$$

Para avaliar o desempenho dos métodos propostos, o seguinte experimento foi realizado: inicialmente, 1000 amostras foram geradas como conjunto de treinamento, seguidas de 300 amostras para teste (mantendo a estrutura e parâmetros dos modelos fixos). A entrada  $u^k$  foi definida como um ruído uniforme no intervalo  $[-1.5 \ 1.5]$  para treinamento, e como  $u^k = \sin(2\pi k/25)$  para teste.

Os valores esperados e previstos para o período de teste são ilustrados nas figuras 5.6 e 5.7 para os modelos eMG e eFT, respectivamente. O modelo eMG foi iniciado a partir da primeira observação e seus parâmetros foram ajustados para:  $\alpha = 0.05$ ,  $w = 25$ ,  $\Sigma_{init} = 5 \times 10^{-1}I_3$  e  $\beta = 0.05$ . O modelo eFT também foi iniciado a partir da primeira observação e seus parâmetros foram ajustados para:  $\alpha = 0.01$ ,  $\sigma_{init} = 0.04$ ,  $\gamma = 25$  e  $\beta = 0.01$ .

O modelo eMG resultante é formado pelas regras:

$$\begin{aligned} \text{Se } x^k \text{ é } A_{[-1.2394 \ -2.3116 \ -1.7919]} \text{ então } y_1 &= 0.2852 + 1.0265u^{k-1} + 0.4575y^{k-1} + 0.2150y^{k-2} \\ \text{Se } x^k \text{ é } A_{[0.1961 \ -0.2750 \ -0.7465]} \text{ então } y_2 &= 0.0747 + 0.9985u^{k-1} + 0.2632y^{k-1} + 0.1481y^{k-2} \\ \text{Se } x^k \text{ é } A_{[0.4100 \ -0.7121 \ 1.0513]} \text{ então } y_3 &= -0.0518 + 0.9757u^{k-1} - 0.3330y^{k-1} + 0.1587y^{k-2} \\ \text{Se } x^k \text{ é } A_{[-0.7453 \ 1.2520 \ -1.0418]} \text{ então } y_4 &= 0.2197 + 1.0019u^{k-1} - 0.2394y^{k-1} + 0.1691y^{k-2} \\ \text{Se } x^k \text{ é } A_{[0.4522 \ -1.3131 \ -1.0731]} \text{ então } y_5 &= 0.1745 + 1.0141u^{k-1} + 0.2924y^{k-1} + 0.3144y^{k-2} \\ \text{Se } x^k \text{ é } A_{[-0.7689 \ 0.7421 \ 1.0137]} \text{ então } y_6 &= 0.0021 + 1.0139u^{k-1} + 0.0506y^{k-1} + 0.0616y^{k-2} \\ \text{Se } x^k \text{ é } A_{[0.7468 \ 1.1391 \ 0.0524]} \text{ então } y_7 &= -0.0152 + 1.0038u^{k-1} + 0.0200y^{k-1} + 0.1544y^{k-2} \end{aligned} \quad (5.10)$$

onde  $x^k$  é o vetor de entrada, ou seja,  $x^k = [u^{k-1} \ y^{k-1} \ y^{k-2}]^T$ ; e  $A_{[x \ y \ z]}$  representa um conjunto nebuloso Gaussiano multivariável centrado em  $[x \ y \ z]^T$ .

A figura 5.8 ilustra a árvore resultante do modelo eFT. As seguintes regras podem ser extraídas dessa árvore:

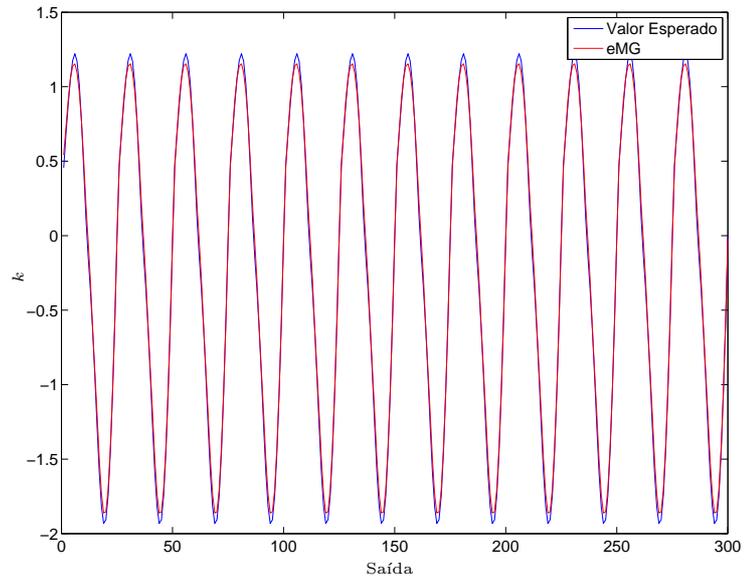


Figura 5.6: Resultado da identificação para o eMG

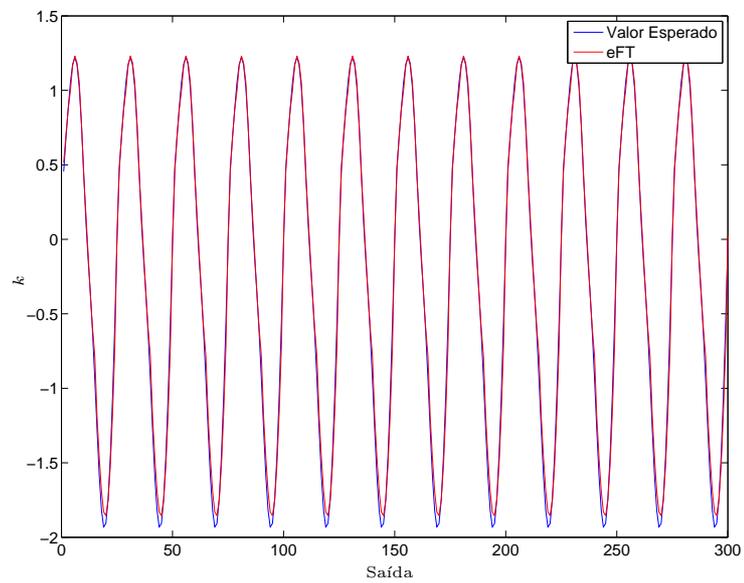


Figura 5.7: Resultado da identificação para o eFT

Se  $y^{k-1} < -0.8055$  então  $y_1 = 0.1400 + 0.9935u^{k-1} + 0.1221y^{k-1} + 0.5058y^{k-2}$   
Se  $-0.8055 < y^{k-1} < -0.4574$  e  $y^{k-2} < -0.4378$  então  $y_2 = -0.0818 + 0.9907u^{k-1} + 0.1962y^{k-1} + 0.0380y^{k-2}$   
Se  $-0.8055 < y^{k-1} < -0.4574$  e  $-0.4378 < y^{k-2} < 1.1462$  então  $y_3 = 0.0230 + 0.9940u^{k-1} - 0.0321y^{k-1} + 0.0739y^{k-2}$   
Se  $-0.8055 < y^{k-1} < -0.4574$  e  $y^{k-2} > 1.1462$  então  $y_4 = 0.0342 + 0.9919u^{k-1} + 0.0881y^{k-1} + 0.0529y^{k-2}$   
Se  $-0.4574 < y^{k-1} < 1.1462$ . então  $y_5 = -0.0032 + 1.0035u^{k-1} + 0.0073y^{k-1} + 0.0492y^{k-2}$   
Se  $y^{k-1} > 1.0048$  e  $y^{k-2} < -0.2539$  então  $y_6 = -0.0431 + 0.9984u^{k-1} - 0.0465y^{k-1} + 0.2139y^{k-2}$   
Se  $y^{k-1} > 1.0048$  e  $y^{k-2} > -0.2539$  então  $y_7 = 0.0116 + 0.9983u^{k-1} + 0.0095y^{k-1} + 0.2230y^{k-2}$

(5.11)

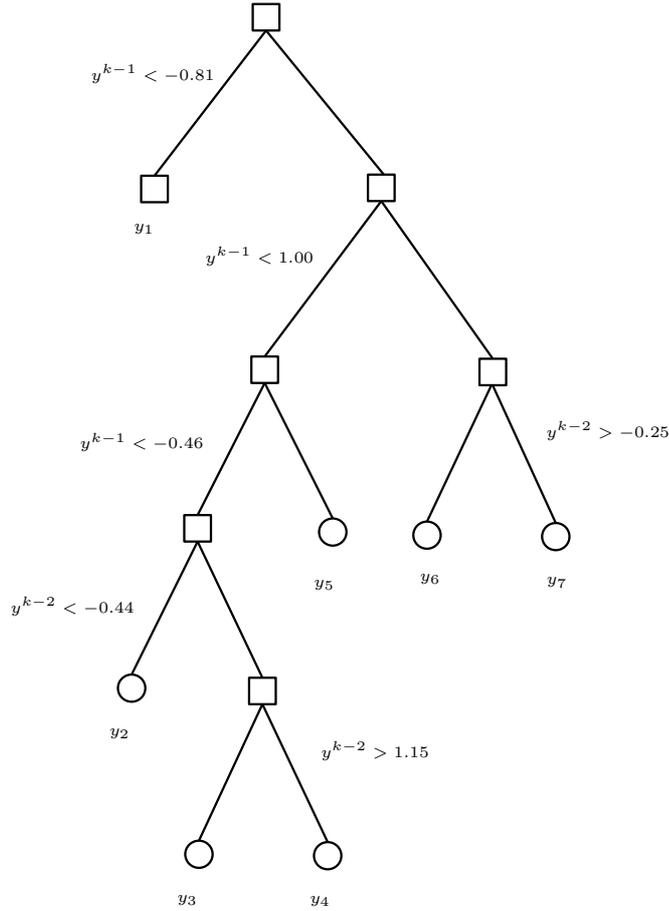


Figura 5.8: Árvore de regressão linear nebulosa evolutiva para o problema de identificação

Os resultados obtidos foram comparados com outros modelos funcionais evolutivos. A tabela 5.3 descreve os resultados comparativos representados através do número de regras geradas e do NDEI. O RMSE não foi utilizado nesse experimento, porque os modelos eTS e xTS necessitam de dados normalizados no intervalo  $[0, 1]$  para o treinamento. Como o restante dos modelos não requer essa premissa, a normalização foi realizada apenas para esses modelos,

e os resultados foram comparados utilizando uma medida de erro independente da escala dos dados. Os parâmetros do modelo eTS foram ajustados para  $r = 0.7$  e  $\omega = 750$ . O modelo xTS tem  $\omega = 750$ . O modelo DENFIS utilizou o limiar de distância igual a 0.28.

Tabela 5.3: Desempenho comparativo para o problema de identificação

Nome do modelo	Número de regras	NDEI
eTS	7	0.1036
xTS	7	0.0937
DENFIS	7	0.0849
eFT	7	0.0657
eMG	7	0.0507

A tabela 5.3 sugere que o modelo eMG possui o melhor desempenho dentre os modelos avaliados. A tabela 5.4 descreve comparações pareadas entre os resultados dos modelos, utilizando o teste *MGN* (5.3). Analisando essa tabela, percebe-se que os modelos eFT e eMG apresentam desempenho estatisticamente superior aos modelos xTS, eTS e DENFIS, para um nível de significância 0.05. Além disso, a última linha da tabela mostra o modelo eMG possui desempenho superior ao modelo eFT para esse problema.

Tabela 5.4: Avaliação do teste *MGN* para o problema de identificação

Modelos	<i>MGN</i>	p-valor
eFT vs eTS	8.9716	0.0000
eFT vs xTS	6.0409	0.0000
eFT vs DENFIS	3.1260	0.0010
eMG vs eTS	25.0673	0.0000
eMG vs xTS	20.6050	0.0000
eMG vs DENFIS	10.7121	0.0000
eMG vs eFT	6.64991	0.0000

### 5.2.3 Problema de Alta Dimensão

Nessa seção, o desempenho dos modelos propostos foi avaliado em um problema de previsão de séries temporais de alta dimensão. Esse experimento foi realizado para avaliar como os modelos propostos se comportam com espaços de entrada de alta dimensão.

A série utilizada é formada por medidas anuais da largura de anéis presentes no tronco de árvores expressas em valores adimensionais. A série utilizada foi medida na Argentina entre os anos 441 e 1974 e corresponde ao conjunto de dados *arge030* do repositório *Time Series Data Library* (Hyndman, 2010).

O modelo prevê o valor da largura no próximo ano ( $y^{k+1}$ ), utilizando o valor atual e valores atrasados da série. Pouzols e Lendasse (2010) sugere os 10 últimos valores da série como entradas, excluindo  $y^{k-4}$  e  $y^{k-6}$ , totalizando 8 entradas, ou seja, o modelo é definido da seguinte forma:

$$y^{k+1} = f(y^k, y^{k-1}, y^{k-2}, y^{k-3}, y^{k-5}, y^{k-7}, y^{k-8}, y^{k-9}) \quad (5.12)$$

Diferentemente dos experimentos anteriores, os parâmetros dos modelos não foram escolhidos de forma a gerar estruturas com o mesmo número de regras. O objetivo desse experimento é avaliar, além do desempenho, se os modelos possuem algum comportamento indesejável para problemas de alta dimensão, como por exemplo a geração exacerbada de regras. Assim, os parâmetros dos modelos exibidos nos experimentos a seguir são relacionados aos menores valores obtidos para o índice NDEI.

Os experimentos foram realizados da seguinte forma: as primeiras 1013 amostras da série foram utilizadas como conjunto de treinamento, e as 511 amostras remanescentes foram utilizadas para teste (com os parâmetros e estrutura dos modelos fixa).

Os valores esperados e previstos para o período de teste são ilustrados nas figuras 5.9 e 5.10, para os modelos eMG e eFT, respectivamente. O modelo eMG foi iniciado a partir da primeira observação e seus parâmetros foram ajustados para:  $\alpha = 0.05$ ,  $w = 20$ ,  $\Sigma_{init} = 8 \times 10^{-2}I_2$  e  $\beta = 0.01$ . O modelo eFT também foi iniciado a partir da primeira observação e seus parâmetros foram ajustados para:  $\alpha = 0.05$ ,  $\sigma_{init} = 0.01$ ,  $\gamma = 20$ , e  $\beta = 0.01$ .

Os modelos eMG e eFT possuem apenas duas regras. A figura 5.11 ilustra a árvore resultante do modelo eFT.

A tabela 5.5 descreve os resultados comparativos expressos pelo número de regras geradas e pelo NDEI. O RMSE não foi utilizado nesse experimento porque os dados não foram normalizados para todos os modelos. Os parâmetros do modelo eTS foram ajustados para  $r = 2$  e  $\omega = 750$ . O modelo xTS tem  $\omega = 750$ . O modelo DENFIS utilizou o limiar de distância igual à 0.23.

A tabela 5.5 sugere que o modelo eFT possui o melhor desempenho dentre os modelos avaliados. Além disso, os modelos propostos, juntamente com o modelo eTS, possuem o menor número de regras. Esses resultados também sugerem que o modelo xTS possui problemas com a dimensionalidade dos dados. Isso possivelmente pode ser atribuído a um limiar fixo para geração de novas regras, presente no algoritmo de treinamento desse modelo. Esse limiar possivelmente não leva em conta a dimensionalidade do espaço de entradas, ou seja, não é função da dimensão das entradas, como no eMG, nem é definido pelo usuário, como no eTS e

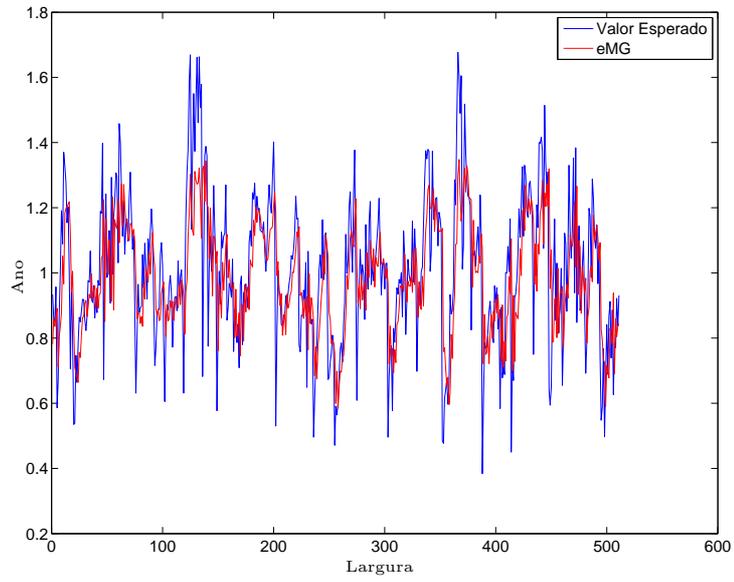


Figura 5.9: Resultado do modelo eMG para o problema de alta dimensão eFT

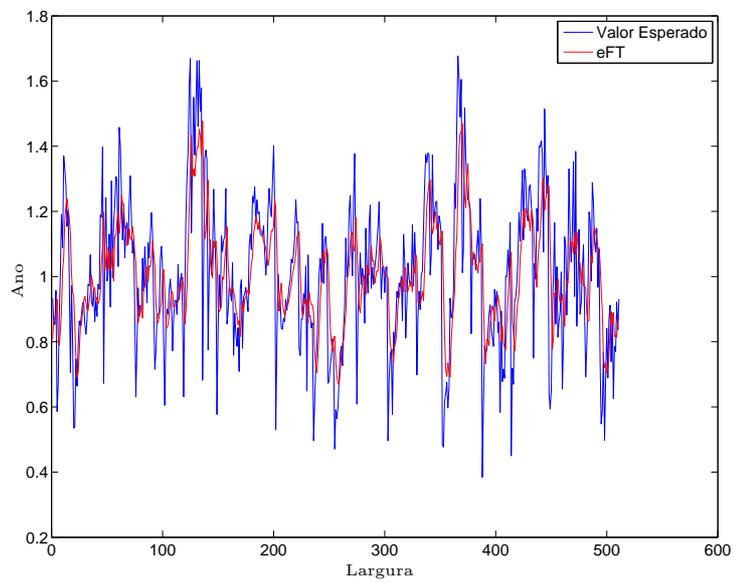


Figura 5.10: Resultado do modelo eFT para o problema de alta dimensão

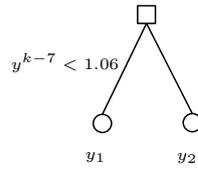


Figura 5.11: Árvore de regressão linear nebulosa evolutiva para o problema de alta dimensão

Tabela 5.5: Desempenho comparativo para o problema de alta dimensão

Nome do modelo	Número de regras	NDEI
DENFIS	3	0.8415
xTS	16	0.8093
eMG	2	0.7767
eTS	2	0.7731
eFT	2	0.7717

DENFIS.

A tabela 5.6 descreve comparações pareadas entre os resultados dos modelos utilizando o teste *MGN* (5.3). Analisando essa tabela, percebe-se que os modelos eFT e eMG apresentam desempenho estatisticamente superior aos modelos xTS, DENFIS e similares ao eTS, para um nível de significância 0.05.

## 5.3 Seleção de Características

Em todos os problemas de regressão descritos na seção 5.2, as variáveis de entrada dos modelos eram definidas antes do início do processo de aprendizagem, baseando-se em conhecimentos anteriores ou técnicas clássicas de definição da estrutura de modelos.

Essa seção avalia a capacidade do modelo xFT proposto no capítulo 4 seção 4.3 de realizar seleção de características durante o processo de aprendizagem. Isto é, avaliar a capacidade desse modelo de selecionar suas entradas de forma automática e incremental, a partir de um fluxo de dados de treinamento.

### 5.3.1 Dados Artificiais

Inicialmente, utilizou-se o problema de identificação de sistemas dinâmicos descrito na seção 5.2.2. Esse problema foi escolhido por ser um problema artificial, em que as variáveis de entrada

Tabela 5.6: Avaliação do teste *MGN* para o problema de alta dimensão

Modelos	<i>MGN</i>	p-valor
eMG vs DENFIS	1.7952	0.0366
eMG vs xTS	2.4959	0.0064
eMG vs eTS	0.2341	0.4075
eFT vs DENFIS	2.1625	0.0155
eFT vs xTS	2.0730	0.0193
eFT vs eTS	0.0846	0.4663
eFT vs eMG	0.0331	0.4868

do modelo são conhecidas. Assim, o objetivo do experimento é verificar se o modelo xFT é capaz de selecionar as variáveis de entrada corretas, dentre um conjunto pré estabelecido de possíveis variáveis.

O seguinte experimento foi realizado: inicialmente os conjuntos de treinamento e teste foram gerados, de forma análoga à seção 5.2.2. Em seguida, o modelo xFT foi iniciado como uma árvore de apenas uma folha, contendo um modelo linear da variável  $u^{k-1}$ . As possíveis variáveis a serem selecionadas foram definidas como os 10 primeiros atrasos da variável de saída, ou seja,  $y^{k-1}, y^{k-2}, \dots, y^{k-10}$ . É esperado que o algoritmo de treinamento seja capaz de selecionar as entradas  $y^{k-1}$  e  $y^{k-2}$ . Além disso, é esperado que o modelo resultante possua um bom desempenho. Os parâmetros utilizados pelo xFT são os mesmos utilizados pelo eFT na seção 5.2.2, ou seja,  $\alpha = 0.01$ ,  $\sigma_{init} = 0.04$ ,  $\gamma = 25$  e  $\beta = 0.01$ .

A figura 5.12 ilustra os valores esperados e previstos para o conjunto de teste. A figura 5.13 ilustra a árvore resultante do processo de aprendizagem. Analisando essa árvore, percebe-se que o algoritmo de aprendizagem selecionou corretamente as variáveis  $y^{k-1}$  e  $y^{k-2}$ .

As seguintes regras podem ser extraídas da árvore resultante:

$$\begin{aligned}
 &\text{Se } y^{k-1} < -0.8055 \text{ e } y^{k-2} < 0.1138 \text{ então } y_1 = 0.2721 + 1.0014u^{k-1} + 0.3533y^{k-1} + 0.3476y^{k-2} \\
 &\text{Se } y^{k-1} < -0.8055 \text{ e } y^{k-2} > 0.1138 \text{ então } y_2 = -0.2226 + 0.9971u^{k-1} - 0.3494y^{k-1} + 0.3271y^{k-2} \\
 &\text{Se } -0.8055 < y^{k-1} < -0.3877 \text{ então } y_3 = -0.0190 + 1.0039u^{k-1} - 0.0116y^{k-1} \\
 &\text{Se } -0.3877 < y^{k-1} < 1.0048 \text{ e } y^{k-2} < -0.9469 \text{ então } y_4 = 0.2591 + 1.0072u^{k-1} - 0.3719y^{k-1} + 0.0623y^{k-2} \\
 &\text{Se } -0.3877 < y^{k-1} < 1.0048 \text{ e } -0.9469 < y^{k-2} < 0.1138 \text{ então } y_5 = 0.1828 + 1.0055u^{k-1} - 0.1411y^{k-1} + 0.3288y^{k-2} \\
 &\text{Se } -0.3877 < y^{k-1} < 1.0048 \text{ e } y^{k-2} > 0.1138 \text{ então } y_6 = 0.0001 + 1.0058u^{k-1} + 0.0422y^{k-1} + 0.0380y^{k-2} \\
 &\text{Se } y^{k-1} > 1.0048 \text{ e } y^{k-2} < 0.4815 \text{ então } y_7 = 0.3572 + 1.0012u^{k-1} - 0.2843y^{k-1} + 0.2550y^{k-2} \\
 &\text{Se } y^{k-1} > 1.0048 \text{ e } y^{k-2} > 0.4815 \text{ então } y_8 = -0.3384 + 1.0002u^{k-1} + 0.3710y^{k-1} + 0.1231y^{k-2}
 \end{aligned} \tag{5.13}$$

Analisando as regras acima, percebe-se que nem todas as regras possuem consequentes que são funções das três variáveis de entrada do modelo. Por exemplo, o modelo linear presente

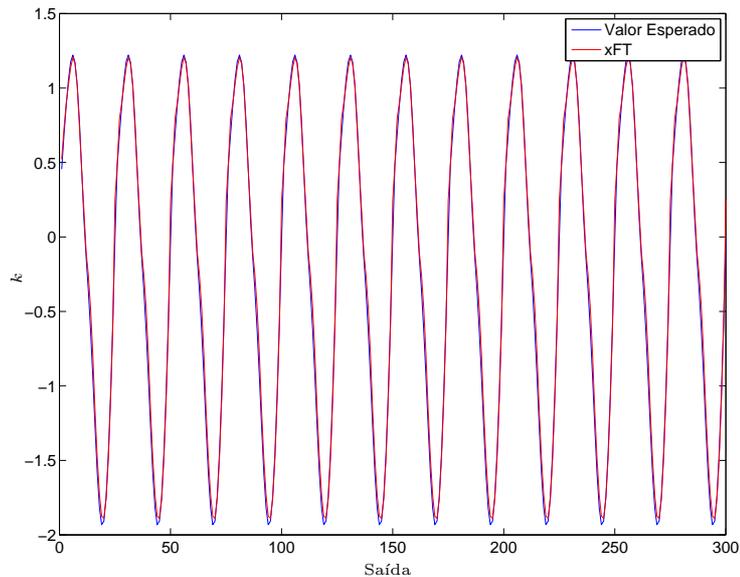


Figura 5.12: Resultado da identificação para o xFT

no conseqüente da terceira regra ( $y_3$ ) é função apenas de  $u^{k-1}$  e  $y^{k-1}$ , uma vez que somente a variável  $y^{k-1}$  está presente nos nós internos no caminho do nó raiz até a folha correspondente. Nota-se também que a variável  $u^{k-1}$  está presente no conseqüente de todas as regras, pois foi escolhida como variável inicial da árvore.

A tabela 5.7 sumariza o desempenho do modelo xFT comparado com outros modelos evolutivos. Os parâmetros e variáveis de entrada dos modelos alternativos são os mesmos descritos na seção 5.2.2. Essa tabela sugere que o modelo xFT possui desempenho superior aos modelos eTS, xTS e DENFIS, e inferior aos modelos eFT e eMG, para esse problema.

A tabela 5.8 descreve comparações pareadas entre os resultados dos modelos utilizando o teste *MGN* (5.3). Analisando essa tabela, percebe-se que os modelos xFT apresenta desempenho estatisticamente superior aos modelos xTS, eTS e DENFIS, equivalente ao modelo eFT e inferior ao modelo eMG, para um nível de significância 0.05.

### 5.3.2 Previsão de Séries Temporais

Em seguida, o modelo xFT foi avaliado no problema de alta dimensão, descrito na seção 5.2.3. Esse experimento foi realizado para avaliar o comportamento do modelo xFT em problemas com essa característica, uma vez que esse modelo poderia ser capaz de gerar regras mais compactas e, conseqüentemente, mais interpretáveis.

O seguinte experimento foi realizado: o valor atual  $y^k$  e os 9 primeiros atrasos da série  $y^{k-1}, \dots, y^{k-9}$  foram utilizados como possíveis variáveis a serem selecionadas. A variável  $y^k$

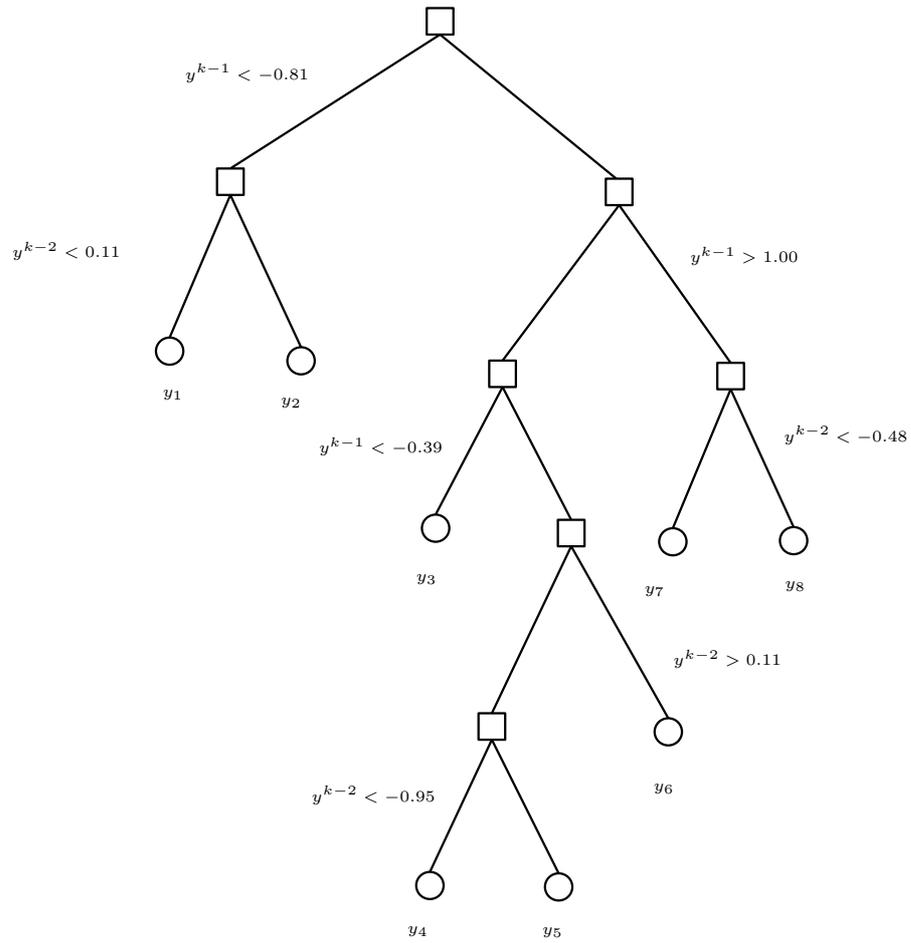


Figura 5.13: Árvore de regressão linear nebulosa evolutiva com seleção de características para o problema de identificação

Tabela 5.7: Desempenho comparativo do modelo xFT para o problema de identificação

Nome do modelo	Número de regras	NDEI
eTS	7	0.1036
xTS	7	0.0937
DENFIS	7	0.0849
xFT	8	0.0776
eFT	7	0.0657
eMG	7	0.0507

Tabela 5.8: Avaliação do teste *MGN* para o problema de seleção de características

Modelos	<i>MGN</i>	p-valor
xFT vs eTS	7.2806	0.0000
xFT vs xTS	4.8685	0.0000
xFT vs DENFIS	1.9499	0.0261
eFT vs xFT	1.5548	0.0605
eMG vs xFT	7.6768	0.0000

foi escolhida como variável inicial do modelo, utilizando o ranqueamento das variáveis pelo coeficiente de correlação amostral. Os parâmetros utilizados pelo xFT são os mesmos utilizados pelo eFT na seção 5.2.3, ou seja,  $\alpha = 0.05$ ,  $\sigma_{init} = 0.01$ ,  $\gamma = 20$ , e  $\beta = 0.01$ .

A figura 5.14 ilustra os valores esperados e previstos para o conjunto de teste. A figura 5.15 ilustra a árvore resultante. Analisando essa árvore, percebe-se que o modelo resultante é função apenas de 2 variáveis de entrada,  $y^k$  (escolhida inicialmente) e  $y^{k-2}$ .

As seguintes regras podem ser extraídas da árvore resultante:

$$\begin{aligned}
 &\text{Se } y^{k-2} < 0.8451 \text{ e } y^k < 1.1286 \text{ então } y_1 = 0.3861 + 0.3348y^k + 0.2626y^{k-2} \\
 &\text{Se } y^{k-2} < 0.8451 \text{ e } y^k > 1.1286 \text{ então } y_1 = 0.8426 + 0.2090y^k + 0.0170y^{k-2} \\
 &\text{Se } y^{k-2} > 0.8451 \text{ então } y_1 = 0.2925 + 0.3789y^k + 0.3231y^{k-2}
 \end{aligned} \tag{5.14}$$

A tabela 5.9 sumariza o desempenho do modelo xFT comparado com outros modelos evolutivos. Os parâmetros e entradas dos modelos alternativos são os mesmos descritos na seção 5.2.3.

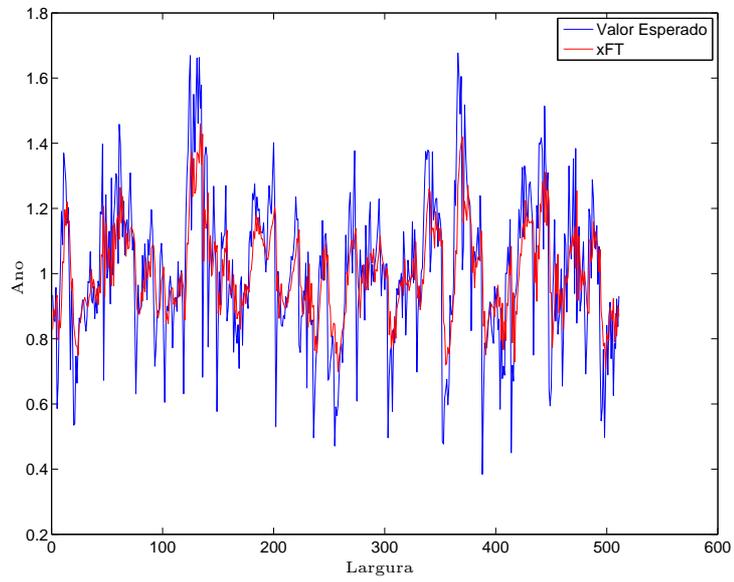


Figura 5.14: Resultado do modelo xFT para o problema de alta dimensão

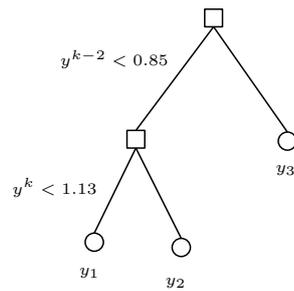


Figura 5.15: Árvore de regressão linear nebulosa evolutiva com seleção de características para o problema de alta dimensão

Tabela 5.9: Desempenho comparativo do modelo xFT para o problema de alta dimensão

Nome do modelo	Número de regras	NDEI
DENFIS	3	0.8415
xTS	16	0.8093
eMG	2	0.7767
eTS	2	0.7731
eFT	2	0.7717
xFT	3	0.7518

A tabela 5.9 sugere que o modelo xFT possui desempenho superior a todos os modelos, para esse problema. Além disso, apesar do modelo xFT possuir 3 regras, essas regras são funções de apenas 2 variáveis de entrada, ao contrário dos outros modelos que possuem 8 variáveis de entrada.

A tabela 5.10 descreve comparações pareadas entre os resultados dos modelos com o teste *MGN* (5.3). Analisando essa tabela, percebe-se que os modelos xFT apresenta desempenho estatisticamente superior aos modelos xTS, eTS, DENFIS e eFT e equivalente ao modelo eMG, para um nível de significância 0.05.

Tabela 5.10: Avaliação do teste *MGN* do modelo xFT para o problema de alta dimensão

Modelos	<i>MGN</i>	p-valor
xFT vs DENFIS	2.9719	0.0015
xFT vs xTS	3.2711	0.0006
xFT vs eMG	1.6060	0.0544
xFT vs eTS	2.0121	0.0224
xFT vs eFT	2.0103	0.0225

## 5.4 Detecção e Diagnóstico Adaptativo de Falhas

Nessa seção o modelo proposto no capítulo 3 seção 3.5 é avaliado em problemas de detecção e diagnóstico adaptativo de falhas.

Nos dois problemas avaliados, o modelo inicia o processo de monitoramento sem nenhum conhecimento sobre os possíveis modos de falha. É esperado, então, que o modelo seja capaz de detectar os modos de falha na primeira vez que estes são apresentados e, em seguida, seja capaz de identificar os modos de falha detectados na próxima vez que forem apresentados.

Os resultados obtidos não foram comparados com nenhuma outra metodologia, porque, embora existam modelos capazes de realizar detecção adaptativa de falhas (Filev e Tseng, 2006; Lughofer, 2008; Lughofer e Guardiola, 2008; Wang e Vrbanek, 2008), o presente autor não conhece nenhuma outra metodologia capaz de realizar diagnóstico adaptativo de falhas.

Como a metodologia é baseada em um classificador nebuloso evolutivo, ela poderia ser avaliada aplicando-se outros classificadores nebulosos evolutivos propostos na literatura (Angelov e Zhou, 2008; Lughofer et al., 2007), a título de comparação de desempenho. Porém, esses classificadores são baseados em aprendizado supervisionado, ao contrário do classificador proposto, que é baseado em aprendizado semi supervisionado. Conforme discutido anteriormente, essa metodologia de aprendizagem foi escolhida para maximização do grau de automatismo do processo de monitoramento, ou seja, para minimizar, ao extremo, a intervenção do operador do sistema no processo de aprendizagem.

#### 5.4.1 Detecção e Diagnóstico de Falhas em um Atuador

Nessa seção, a metodologia de detecção e diagnóstico adaptativo de falhas proposta é avaliada em um *benchmark* de detecção e diagnóstico de falhas definido por um grupo de estudos europeu, denominado DAMADICS (*Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems*) (Bartys et al., 2006). Diversos trabalhos utilizam esse *benchmark* para avaliar metodologias de detecção e diagnóstico de falhas (Bocaniala and da Costa, 2006; Laurentys et al., 2010a,b; Mendonca et al., 2009).

O sistema a ser monitorado consiste em um atuador que controla o fluxo de entrada de água em uma caldeira. Essa caldeira faz parte de um processo de evaporação composto por 5 estágios de uma fábrica de açúcar polonesa. O atuador é composto por três partes distintas: a válvula de controle  $V$ , o servo motor pneumático  $S$  e o posicionador  $P$ . A válvula controla o fluxo de água que passa por um cano. O servo motor pneumático realiza variações no plugue da válvula de controle, de forma a agir na taxa de fluxo de água. O motor é composto por um diafragma preenchido por um fluido, de maneira que, comprimindo esse fluido, é possível realizar o deslocamento da haste do motor. A haste do motor está conectada ao plugue da válvula, controlando sua abertura. O posicionador é um dispositivo utilizado para corrigir posicionamentos incorretos da haste do motor, causados por fontes externas ou internas, tais como: atrito, variações na pressão fornecida ao servo motor, etc. A figura 5.16<sup>4</sup> ilustra o diagrama esquemático do atuador.

Utilizou-se um simulador para criar cenários de falha no atuador. Esse simulador faz parte do *benchmark* e foi desenvolvido a partir de descrições analíticas e conhecimentos prévios provindos de um atuador industrial real. Esse modelo foi sintonizado e validado com dados reais.

---

<sup>4</sup> adaptada de Bartys et al. (2006)

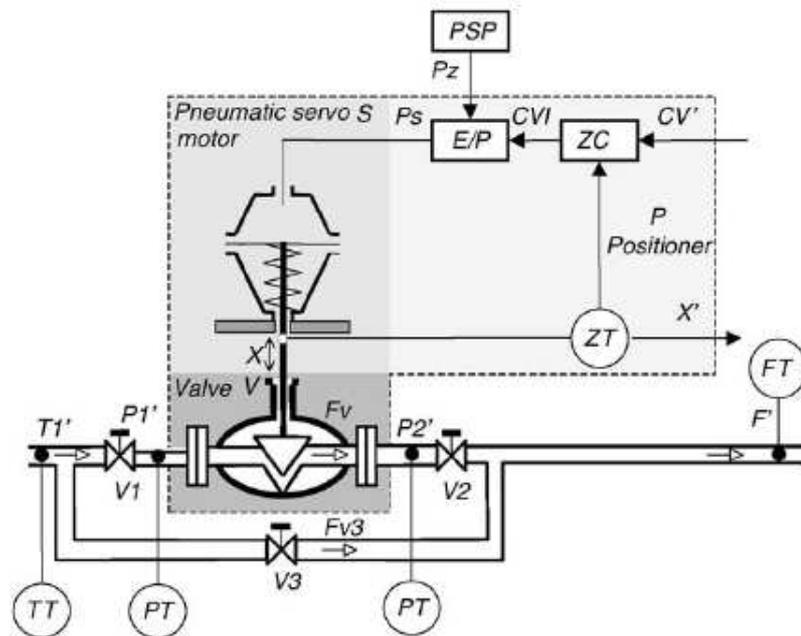


Figura 5.16: Diagrama esquemático do atuador

A figura 5.17 ilustra o modelo do simulador. As variáveis de entrada são: valor de referência do processo  $CV$ , a pressão da água na entrada da válvula  $P1$ , a pressão da água na saída da válvula  $P2$ , a temperatura da água  $T$  e o vetor de falhas  $f$ . As variáveis de saída são: a taxa de vazão da água  $F$  e a posição do tampão da válvula  $X$ . Todas as variáveis descritas são amostradas a 1 Hz e, com exceção do vetor de falhas, são acrescidas de ruído, simulando os erros de medição.

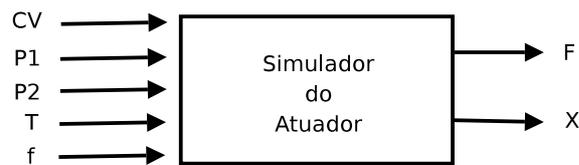


Figura 5.17: Diagrama do simulador do atuador

A tabela 5.11 descreve todos os modos de operação considerados nesse experimento.

As variáveis selecionadas como entradas foram: a taxa de vazão da água  $F$ , a posição do tampão da válvula  $X$ , a diferença  $dP$  entre a pressão da água na entrada da válvula  $P1$  e na saída da válvula  $P2$ . Segundo Bocaniala and da Costa (2006), essas variáveis proveem a melhor distinção entre modos de operação.

Tabela 5.11: Modos de Operação Simulados

Índice	Descrição
0	Operação Normal
1	Agarramento da válvula
2	Sedimentação na válvula ou em sua base
3	Variação inesperada de pressão ao longo da válvula
4	Falha no sensor de medição de vazão

A figura 5.18 ilustra o comportamento dessas variáveis para o cenário de falha 1, (agarramento da válvula) e a figura 5.19 ilustra o comportamento para o cenário de falha 2 (sedimentação na válvula ou em sua base). Para essas duas simulações, as falhas iniciam em  $t = 100$  segundos. O modelo utilizado nas simulações normaliza todas as leituras no intervalo unitário.

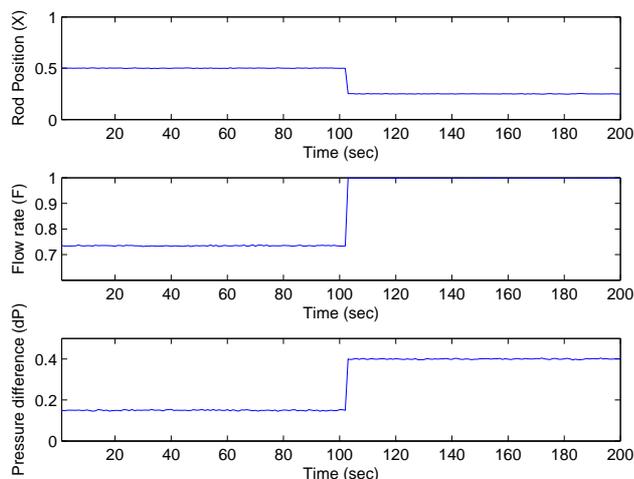


Figura 5.18: Cenário de falha 1

O seguinte experimento foi realizado para validar a metodologia proposta: o classificador iniciou com uma única regra, correspondendo à operação normal. Os parâmetros do antecedente da regra foram estimados utilizando-se 100 amostras das variáveis durante operação normal do sistema. O parâmetro  $\Sigma_{init}$  foi definido como a covariância dessas amostras iniciais. Em seguida, um fluxo de dados de aproximadamente 75 minutos contendo todos os modos de

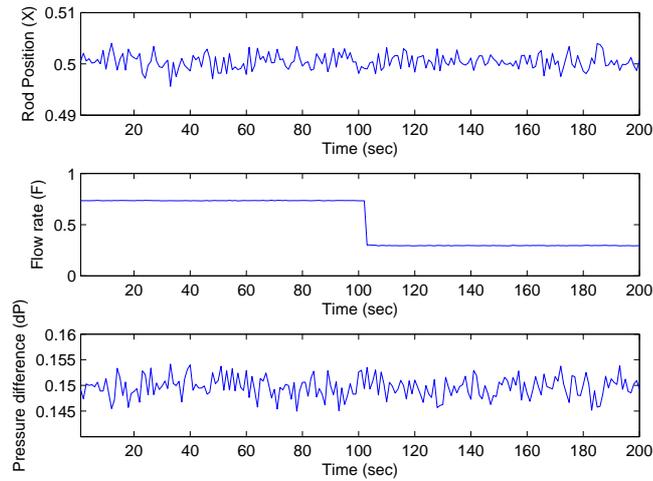


Figura 5.19: Cenário de falha 2

operação possíveis foi utilizado como entrada da metodologia. A figura 5.20 ilustra o índice dos modos de operação em função do tempo de simulação.

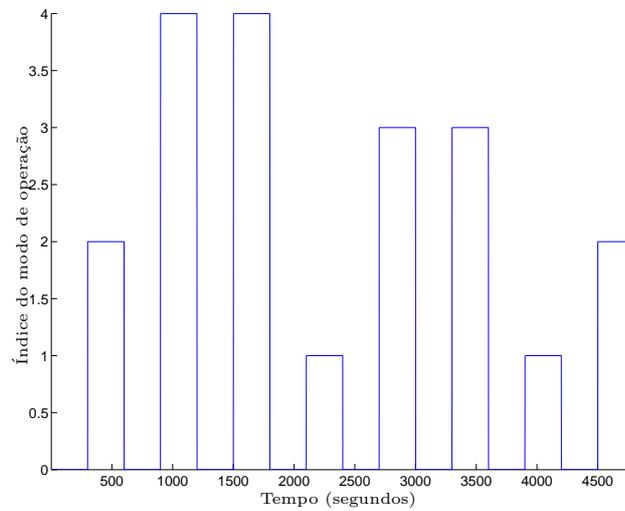


Figura 5.20: Fluxo de dados simulados para o problema do atuador

Os parâmetros do algoritmo de agrupamento foram definidos como  $w = 100$ ,  $\alpha = 0.01$ ,  $\beta = 0.01$  e o parâmetro do classificador como  $T_{\mu_y} = 0.01$ . A metodologia foi capaz de detectar todos os modos de falha e identificá-los na próxima ocorrência. De um total de 4500 amostras

simuladas, apenas 5 (aproximadamente 0.1%) necessitaram de intervenção do operador, isto é, necessitaram ser rotuladas pelo operador.

A figura 5.21 mostra os índices dos modos de operação estimados pelo classificador e os instantes temporais (marcados por um '\*') em que a intervenção do operador foi necessária. Analisando essa figura, nota-se que apenas uma intervenção foi requisitada para cada condição de falha. A intervenção em  $t \approx 3700$  foi identificada pelo operador como condição normal, portanto, pode ser interpretada como um falso alarme.

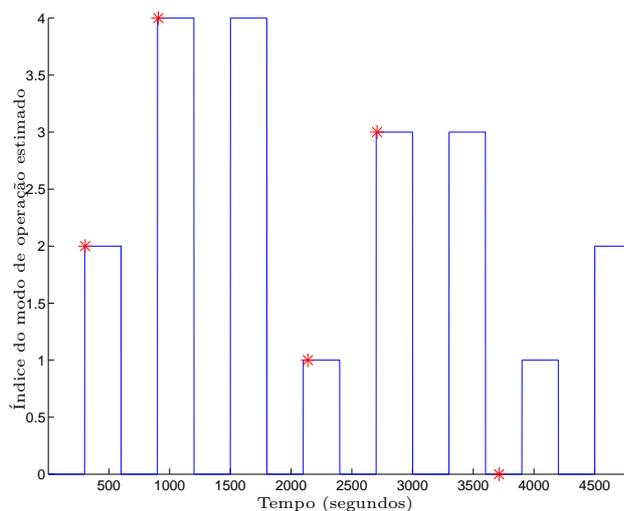


Figura 5.21: Saída do classificador para o problema do atuador

A tabela 5.12 descreve os valores de dois índices de desempenho para avaliar quantitativamente os resultados do experimento proposto. O tempo de atraso na detecção (TAD) em segundos, isto é, o intervalo temporal entre a primeira ocorrência de um modo de falha e a sua detecção e o número de regras finais (NRF) geradas para cada um dos modos de operação. Nota-se que algumas regras criadas tiveram seus consequentes definidos automaticamente. Isto é, o consequente é definido como o modo de operação estimado pelo operador. Por exemplo, o classificador iniciou com uma regra descrevendo o modo de operação normal e duas regras foram criadas, uma com consequente definido automaticamente pelo classificador e outra com consequente definido pelo operador (falso alarme). Embora o TAD relativo a falha 1 seja maior que das outras falhas, a segunda ocorrência dessa falha foi identificada em 7 segundos.

#### 5.4.2 Detecção e Diagnóstico de Falhas em um Motor de Indução

Nessa seção, a metodologia de detecção e diagnóstico adaptativo de falhas proposta foi aplicada no monitoramento de um motor de indução.

Tabela 5.12: Índices de desempenho para o problema do atuador

Modo de Operação	TAD (seg)	NRF
0	-	3
1	30	1
2	4	2
3	8	3
4	6	1

Motores de indução são máquinas elétricas frequentemente utilizadas por serem robustas, mecanicamente simples, adaptáveis a diversas condições de operação e facilmente controláveis. Motores são regularmente expostos a diferentes cargas e condições ambientais. Essas condições adversas, juntamente com seu processo natural de envelhecimento, pode levar a vários tipos de falhas. Assim, o monitoramento de motores é uma atividade crucial para a detecção de vários tipos de falhas em seu estágio inicial, afim de evitar os riscos causados por falhas severas (D'Angelo et al., 2011; Emara et al., 2003; Laurentys et al., 2010c; Nikranjbar et al., 2009).

Todos os experimentos descritos nessa seção utilizaram dados de um modelo dinâmico de um motor de indução. O modelo dinâmico utilizado é baseado em um modelo de transitório de sétima ordem, composto por equações diferenciais não lineares, e é capaz de simular a condição de operação normal do motor, assim como várias condições de falha. Baccarini et al. (2010) fornece uma descrição completa do modelo dinâmico utilizado.

A condição de operação normal e 6 condições de falha distintas foram utilizadas nos experimentos. A tabela 5.13 descreve os modos de operação (MO) simulados, e a figura 5.22 ilustra o modelo utilizado para simulação, composto por uma fonte de energia, um motor de indução e uma carga mecânica. Em todos os os experimentos realizados, foi adicionado um ruído Gaussiano de 1% nas variáveis medidas, a fim de simular ruído nos dados.

A figura 5.23 ilustra os valores de corrente (fase A), tensão (fase A), torque e velocidade de rotação, resultantes da simulação de um curto circuito na fase A do motor para  $t > 1$  segundo.

A técnica de extração de características utilizada foi o cálculo do valor eficaz para a corrente e tensão nas 3 fases, resultando em um vetor de 6 características, utilizado como entrada da metodologia proposta.

O seguinte experimento foi proposto com o intuito de validar a metodologia: inicialmente, para validar a detecção de novos modos de operação, um fluxo de dados contendo todos os 7 modos de operação foi apresentado ao método de detecção e diagnóstico adaptativo de falhas. O fluxo de dados foi gerado usando um tempo de integração de 0.2 segundos durando aproximadamente 70 segundos, contendo 15 segundos de operação normal, seguidos de todas as 6 condições de falha. Em seguida, condições de falhas aleatórias foram simuladas e apresentadas

Tabela 5.13: Modos de operação simulados para os experimentos do motor de indução

Índice	Descrição
0	Operação normal
1	Curto circuito na fase A
2	Curto circuito na fase B
3	Curto circuito na fase C
4	Curto circuito na fase A e B
5	Curto circuito na fase A e C
6	Curto circuito na fase B e C

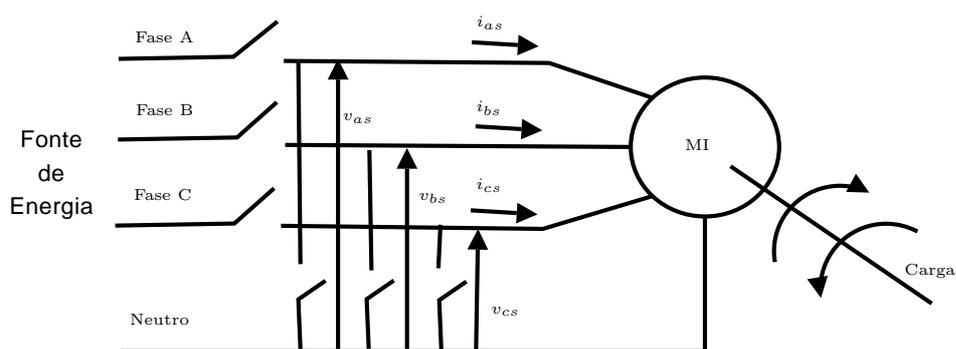


Figura 5.22: Diagrama do modelo utilizado

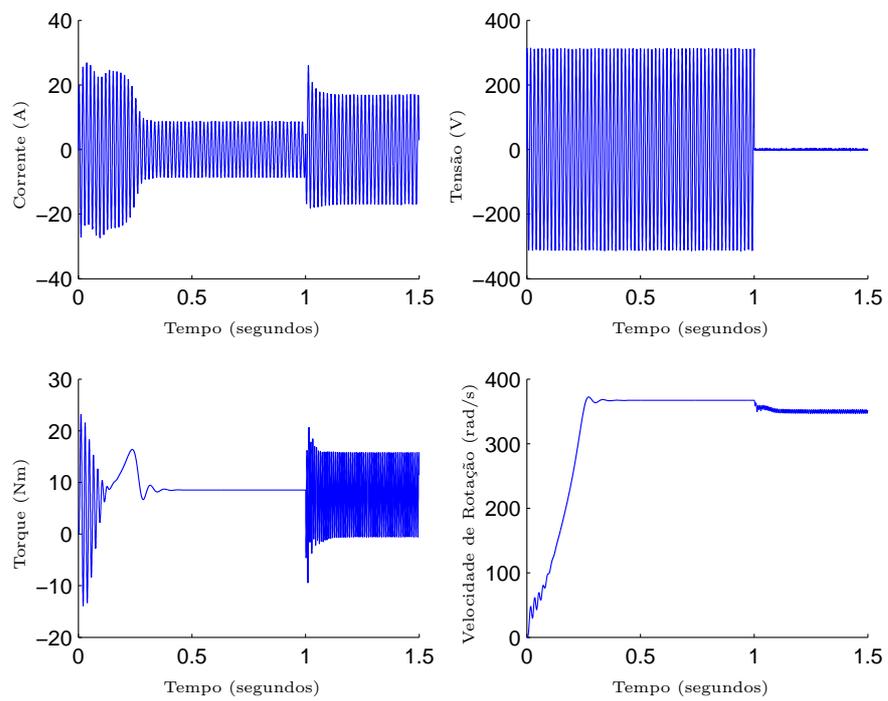


Figura 5.23: Curto circuito na fase A ( $t > 1$  segundo)

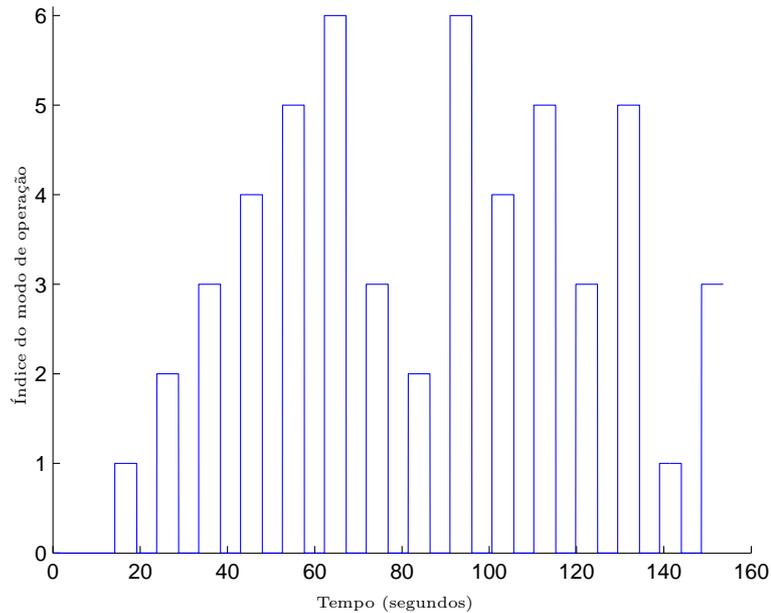


Figura 5.24: Sequência de modos de operação simulados para o problema do motor

ao método, com o objetivo de validar o diagnóstico adaptativo de falhas proposto, isto é, já que todos os modos de operação foram apresentados inicialmente ao classificador, este deve ser capaz de identificar esses modos em sua próxima ocorrência. A figura 5.24 ilustra a sequência de modos de operação simulados.

A metodologia proposta foi iniciada a partir de um única amostra, descrevendo o comportamento normal do motor após o transitório de partida. Os parâmetros do algoritmo de agrupamento foram ajustados para:  $\Sigma_{init} = I_6$ ,  $w = 100$ ,  $\alpha = 0.01$  e  $\beta = 0.01$  e o parâmetro do classificador nebuloso evolutivo foi ajustado para  $T_{\mu_y} = 0.01$ . Os dados contendo o transitório de partida do motor não foram utilizados.

A metodologia proposta foi capaz de detectar todos os modos de falha e identificá-los na próxima ocorrência. A figura 5.25 ilustra os índices dos modos de operação estimados pelo classificador e os instantes temporais (marcados por um '\*') em que foi necessária a intervenção do operador, ou seja, uma nova regra foi criada e não pode ser rotulada automaticamente.

Analisando a figura 5.24, é possível notar que, após todos os modos de falha terem sido apresentados ao sistema ( $t > 70$  segundos), apenas 3 novas regras foram criadas e tiveram que ser rotuladas pelo operador, duas dessas relacionadas à condição de operação normal ( $t \approx 140$  segundos e  $t \approx 150$  segundos) e uma associada à condição de falha 5 ( $t \approx 110$  segundos).

A tabela 5.14 descreve os valores de dois índices de desempenho utilizados para avaliar quantitativamente os resultados do experimento proposto. Analisando essa tabela, percebe-se que a grande maioria dos modos de falha estão associados a um TAD  $\approx 0.1$  segundos, com exceção do modo de falha 1 (TAD = 0.63 segundos). Porém, na segunda vez em que esse

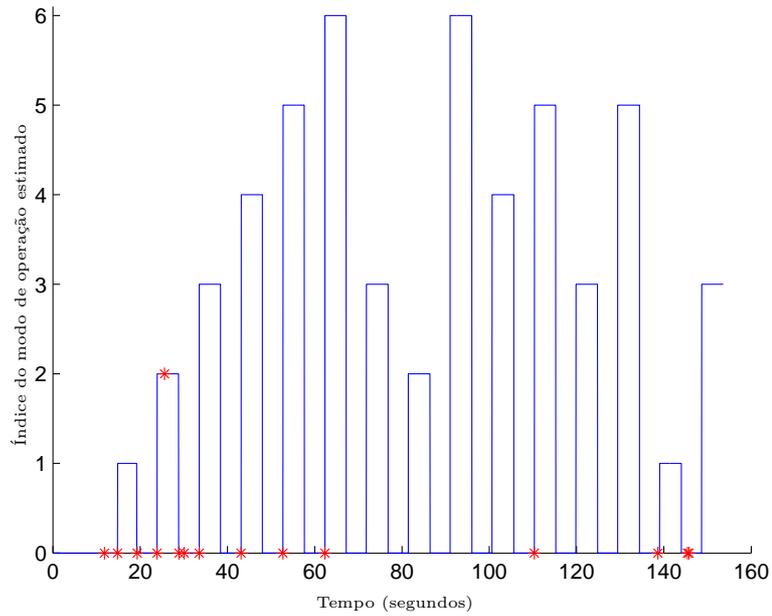


Figura 5.25: Saída do classificador para o problema do motor

modo de falha foi apresentado à metodologia, ele foi identificado após 0.02 segundos (passados) do início de sua ocorrência. Pode-se notar também que o classificador resultante é compacto, contendo 1 ou 2 regras para descrever cada modo de operação.

Tabela 5.14: Índices de desempenho para os dados do motor

OM	TAD (sec)	NRF
0	-	2
1	0.63	1
2	0.08	2
3	0.13	2
4	0.13	1
5	0.13	1
6	0.13	2

# Capítulo 6

## Conclusões e Perspectivas

O desenvolvimento de sistemas nebulosos com alto grau de flexibilidade e autonomia vem servindo de inspiração para vários estudos recentes na área de modelagem nebulosa. Esses sistemas, construídos a partir de um fluxo de dados, têm se mostrado como alternativas promissoras para resolver problemas em ambientes dinâmicos não estacionários. Tendo isso em vista, esse trabalho propôs novas metodologias de modelagem nebulosa evolutiva.

Inicialmente, esse trabalho propôs uma técnica de modelagem evolutiva nebulosa que utiliza um algoritmo de agrupamento recursivo não supervisionado baseado no conceito do aprendizado participativo. Com base nesse algoritmo de agrupamento, dois modelos evolutivos foram propostos, definidos como um conjunto adaptativo de regras nebulosas. Tais modelos utilizam funções de pertinência Gaussianas multivariáveis para definição dos antecedentes das regras, com o objetivo principal de preservar a informação referente à interação entre as variáveis de entradas, presente nos grupos utilizados para a definição das regras.

O primeiro modelo proposto, eMG, é definido como um modelo funcional evolutivo formado por um conjunto de regras do tipo Takagi-Sugeno de primeira ordem. Esse modelo tem como principais características a robustez, herdada do algoritmo de agrupamento utilizado, a precisão, verificada através de sua aplicação em problemas de previsão de séries temporais e identificação de sistemas dinâmicos e o baixo custo computacional.

Em seguida, uma metodologia de detecção e diagnóstico de falhas foi proposta baseada em um modelo linguístico evolutivo. O modelo é composto por um conjunto de regras linguísticas e pode ser visto como um classificador evolutivo que realiza diagnóstico adaptativo de falhas. A metodologia proposta é capaz de identificar novos modos de operação, realizar o diagnóstico de um sistema sem a necessidade de construção de modelos dinâmicos, atualizar informações referentes a modos de operação conhecidos (à medida que esses variem com o tempo), e incorporar informações sobre novos modos de operação detectados. Essa metodologia foi validada por um *benchmark* e no monitoramento da operação de um motor de indução, através de dados simulados.

Propôs-se também uma metodologia de modelagem nebulosa evolutiva baseada em árvores de regressão linear nebulosas construídas a partir de um fluxo de dados para problemas de

regressão. Essa metodologia possui como principal característica a capacidade de definir a topologia da árvore a partir de um algoritmo incremental, baseado na qualidade do modelo resultante. Dois modelos foram propostos, o eFT e o xFT. O xFT é uma variação do eFT, que também é capaz de realizar seleção de características durante o processo de aprendizagem. Os resultados experimentais obtidos sugerem que essa metodologia de modelagem é capaz de gerar modelos precisos, parcimoniosos e facilmente interpretáveis, uma vez que regras funcionais nebulosas podem ser facilmente extraídas de sua topologia.

Três modelos funcionais nebulosos evolutivos foram propostos nesse trabalho para resolver problemas de regressão. Os resultados experimentais obtidos sugerem que o desempenho dos modelos é superior aos de alguns métodos alternativos consagrados na literatura e similares entre si.

Sugere-se utilizar o modelo eMG para problemas em que a prioridade é o baixo custo computacional, como por exemplo, aplicações em tempo real, uma vez que o algoritmo de aprendizado desse modelo possui uma baixa complexidade temporal. Caso a prioridade seja a interpretabilidade, sugere-se utilizar o modelo eFT. Esse modelo possui um custo computacional superior ao eMG, porém um número compacto de regras nebulosas podem ser facilmente extraído de sua topologia. Finalmente, o terceiro modelo, xFT, deve ser utilizado em problemas em que se tem pouco conhecimento *a priori* sobre o sistema a ser modelado, de forma que o modelo é capaz de selecionar entradas e realizar a regressão de forma automatizada. Esse modelo possui o maior custo computacional dentre os modelos descritos e possui interpretabilidade similar à do eFT.

## 6.1 Propostas de Continuidade

Com relação à metodologia de detecção e diagnóstico adaptativo de falhas proposta, sugere-se a integração de metodologias adaptativas de extração/seleção de características no classificador, com o objetivo de aumentar sua eficiência. A metodologia proposta nesse trabalho realiza a seleção/extração de características antes do início processo de monitoramento. Conforme descrito anteriormente, é esperado que as características escolhidas como entradas da metodologia proposta sejam capazes de distinguir todos os modos de operação possíveis. Porém, isso apenas é assumido, uma vez que alguns modos de operação podem ser desconhecidos durante o processo de extração de características. Assim, é esperado que a incorporação de métodos de seleção/extração de características adaptativos no processo de aprendizagem do classificador seja capaz de aumentar sua eficiência discriminatória entre classes. Essa suposição deve então ser verificada, através da aplicação dessa metodologia problemas reais.

Com relação aos métodos de modelagem baseados em árvores de regressão linear nebulosas, sugere-se o estudo de metodologias para reduzir o custo computacional do seu processo de aprendizagem, afim de torná-los aptos à aplicações em tempo real. Uma sugestão para a redução do custo computacional seria utilizar menos pontos de corte candidatos com o valor de corte (o valor central das funções de pertinência presentes no nó interno) adaptativo, atualizado recursivamente a partir dos dados. Além disso, sugere-se a proposição de uma metodologia capaz de realizar a poda da árvore de forma incremental, capaz de reduzir a complexidade

do modelo, de forma análoga a metodologia de união de regras similares presente no modelo eMG.

Finalmente sugere-se a investigação de novas topologias para modelos nebulosos evolutivos. Em especial, sugere-se a definição de um algoritmo de aprendizado evolutivo para redes neurais nebulosas formadas por *unineurons* ponderados (Lemos et al., 2010a). Redes neurais nebulosas são redes neurais formadas por neurônios lógicos nebulosos (Pedrycz e Gomide, 2007). Essas redes possuem como principal característica a possibilidade de extração de conhecimento em forma de regras, a partir de sua topologia, bem como a possibilidade de inserir conhecimento a priori. Essas redes são formadas por uma colaboração sinérgica entre as teorias de conjuntos nebulosos e redes neurais, permitindo uma vasta gama de habilidades de aprendizado, fornecendo assim modelos que integram o tratamento de informação incerta provido pelos sistemas nebulosos e a habilidade de aprendizado proporcionada pelas redes neurais (Pedrycz e Gomide, 2007). Diversos modelos de redes neurais nebulosas já foram propostos na literatura para resolver problemas de identificação de sistemas (Hell et al., 2007), previsão de séries temporais (Ballini e Gomide, 2002) e classificação de padrões (Caminhas et al., 1999).

Uma rede neural nebulosa formada por neurônios baseados em uninormas ponderadas foi proposta por Lemos et al. (2010a). Recentemente foi provado que tais redes são aproximadores universais de funções, para certas escolhas de parâmetros e operadores (Lemos et al., 2011b). O algoritmo de aprendizagem inicial proposto para essa rede é em batelada. Espera-se então, propor uma metodologia evolutiva para a adaptação da topologia e parâmetros dessa rede a partir de um fluxo de dados.

# Referências Bibliográficas

- J. Abonyi, R. Babuska, and F. Szeifert. Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems Man and Cybernetics Part B (Cybernetics)*, 32(5):612–621, Oct 2002. ISSN 1083-4419. doi: 10.1109/TSMCB.2002.1033180.
- M. P. Allen. *Understanding Regression Analysis*. Springer, 1st edition, 1997.
- P. Angelov. Evolving Takagi-Sugeno fuzzy systems from streaming data, eTS+. In P. Angelov, D. Filev, and N. Kasabov, editors, *Evolving Intelligent Systems: Methodology and Applications*. Wiley-Interscience/IEEE Press, 2010.
- P. Angelov and R. Buswell. Evolving rule-based models: A tool for intelligent adaptation. In *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, volume 2, pages 1062–1067, July 2001.
- P. Angelov and D. Filev. An approach to Online identification of Takagi-Suigeno fuzzy models. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, 34(1):484–498, Feb 2004.
- P. Angelov and D. Filev. SimPLeTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models. In *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, pages 1068–1073, May 2005.
- P. Angelov and R. Guthke. A genetic-algorithm-based approach to optimization of bioprocesses described by fuzzy rules. *Bioprocess Engineering*, 16(5):299–303, 1997.
- P. Angelov and A. Kordon. Adaptive inferential sensors based on evolving fuzzy models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(2):529–539, April 2010. ISSN 1083-4419. doi: 10.1109/TSMCB.2009.2028315.
- P. Angelov and X. Zhou. Evolving fuzzy systems from data streams in real-time. In *2006 International Symposium on Evolving Fuzzy Systems*, pages 29–35, Sept. 2006.
- P. Angelov, D. Filev, and N. Kasabov. Guest editorial evolving fuzzy systems - preface to the special section. *IEEE Transactions on Fuzzy Systems*, 16(6):1390–1392, Dec. 2008a.

- P. Angelov, E. Lughofer, and X. Zhou. Evolving fuzzy classifiers using different model architectures. *Fuzzy Sets and Systems*, 159(23):3160 – 3182, 2008b. ISSN 0165-0114. doi: DOI:10.1016/j.fss.2008.06.019.
- P. Angelov, R. Ramezani, and X. Zhou. Autonomous novelty detection and object tracking in video streams using evolving clustering and Takagi-Sugeno type neuro-fuzzy system. In *IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence)*, pages 1456–1463, June 2008c.
- P. P. Angelov. *Evolving rule-based models: a tool for design of flexible adaptive systems*. Springer-Verlag, London, UK, 2002. ISBN 3-7908-1457-1.
- P. P. Angelov and X. Zhou. Evolving Fuzzy-Rule-Based Classifiers From Data Streams. *IEEE Transactions on Fuzzy Systems*, 16(6):1462–1475, DEC 2008.
- K. Astrom and B. Wittenmark. *Adaptive Systems*. Addison-Wesley, USA, 1st edition, 1988.
- R. Babuska. *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Boston, 1998.
- R. Babuska and H. Verbruggen. An overview of fuzzy modeling for control. *Control Engineering Practice*, 4(14):1593–1606, 1996.
- L. M. R. Baccharini, B. R. de Menezes, and W. M. Caminhas. Fault induction dynamic model, suitable for computer simulation: Simulation results and experimental validation. *Mechanical Systems and Signal Processing*, 24(1):300 – 311, 2010. ISSN 0888-3270. doi: DOI:10.1016/j.ymsp.2009.06.014.
- R. Ballini and F. Gomide. Learning in recurrent, hybrid neurofuzzy networks. In *IEEE International Conference on Fuzzy Systems*, pages 785–791, 2002.
- M. Bartys, R. Patton, M. Syfert, S. de las Heras, and J. Quevedo. Introduction to the damadics actuator fdi benchmark study. *Control Engineering Practice*, 14:577–596, 2006.
- C. D. Bocaniala and J. S. da Costa. Application of a novel fuzzy classifier to fault detection and isolation of the damadics benchmark problem. *Control Engineering Practice*, 14(6):653 – 669, 2006. A Benchmark Study of Fault Diagnosis for an Industrial Actuator.
- G. E. P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990. ISBN 0816211043.
- J. Breiman, L. Friedman, R. Olshen, and C. Stone. *Classification and regression trees*. Wadsworth Mathematics Series, San Diego, California, 1984.
- W. Caminhas, H. Tavares, F. Gomide, and W. Pedrycz. Fuzzy sets based neural networks: Structure, learning and applications. *Journal of Advanced Computational Intelligence*, 3(3): 151–157, 1999.

- F.-J. Chang and Y.-T. Chang. Adaptive neuro-fuzzy inference system for prediction of water level in reservoir. *Advances in Water Resources*, 29(1):1 – 10, 2006. ISSN 0309-1708.
- S. L. Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2(3), 1994.
- H. J. Cho and J. K. Park. An expert system for fault section diagnosis of power systems using fuzzy relations. *IEEE Transactions on Power Systems*, 12(1):342–347, Feb 1997.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2nd revised edition edition, September 2001.
- M. F. D’Angelo, R. M. Palhares, R. H. Takahashi, R. H. Loschi, L. M. Baccarini, and W. M. Caminhas. Incipient fault detection in induction machine stator-winding using a fuzzy-bayesian change point detection approach. *Applied Soft Computing*, 11(1):179 – 192, 2011. ISSN 1568-4946. doi: DOI:10.1016/j.asoc.2009.11.008.
- D. Dasgupta and S. Forrest. Novelty detection in time series data using ideas from immunology. In *Neural Information Processing Systems (NIPS) Conference*, 1996.
- F. X. Diebold and R. S. Mariano. Comparing predictive accuracy. *Journal of Business and Economics Statistics*, 13:253 – 263, 1995.
- A. Dobra and J. Gehrke. Secret: a scalable linear regression tree algorithm. In *KDD ’02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 481–487, New York, NY, USA, 2002. ACM. ISBN 1-58113-567-X. doi: <http://doi.acm.org/10.1145/775047.775117>.
- Y. Dong, Z. Sun, and H. Ha. A cosine similarity-based negative selection algorithm for time series novelty detection. *Mechanical Systems and Signal Processing*, 20(6):1461–1472, 2006.
- R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, November 2000.
- H. Emara, M. Ammar, A. Bahgat, and H. Dorrah. Stator fault estimation in induction motors using particle swarm optimization. In *IEMDC’03. IEEE International Electric Machines and Drives Conference, 2003*, volume 3, pages 1469–1475, June 2003.
- M. J. Er and S. Wu. A fast learning algorithm for parsimonious fuzzy neural systems. *Fuzzy Sets and Systems*, 126(3):337 – 351, 2002.
- D. Filev and F. Tseng. Novelty detection based machine health prognostics. In *2006 International Symposium on Evolving Fuzzy Systems*, pages 193–199, Sept. 2006.
- B. Fritzke. Growing cell structures: a self-organizing network for unsupervised and supervised learning. *Neural Networks*, 7:1441–1460, November 1994. ISSN 0893-6080. doi: 10.1016/0893-6080(94)90091-4.

- D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989. ISBN 0201157675.
- R. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, 1984.
- G. Gross and F. Galiana. Short-term load forecasting. *Proceedings of the IEEE*, 75(12):1558–1573, Dec. 1987. ISSN 0018-9219.
- I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003. ISSN 1532-4435.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag, Berlin, Germany, 2001.
- M. Hell. *Abordagem Neurofuzzy para Modelagem de Sistemas Dinamicos Nao Lineares*. PhD thesis, Universidade Estadual de Campinas, Campinas, Brazil, October 2008.
- M. Hell, P. Costa, and F. Gomide. Participatory learning in power transformers thermal modeling. *IEEE Transactions on Power Delivery*, 23(4):2058–2067, Oct. 2008.
- M. Hell, P. Costa, Jr., and F. Gomide. Recurrent neurofuzzy network in thermal modeling of power transformers. *IEEE Transactions on Power Delivery*, 22(2):904–910, APR 2007. ISSN 0885-8977. doi: {10.1109/TPWRD.2006.874613}.
- R. J. Hyndman. Time series data library. <http://robjhyndman.com/TSDL>, December 2010.
- R. Isermann. Process fault detection based on modeling and estimation methods—a survey. *Automatica*, 20(4):387 – 404, 1984.
- J. Jang. ANFIS - Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems Man and Cybernetics*, 23(3), 1993.
- J.-S. R. Jang and E. Sun, C.-T. and Mizutani. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice-Hall, 1st edition, 1997.
- C. Janikow. Fuzzy decision trees: Issues and methods. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, 28(1):1–14, FEB 1998.
- R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, (82 (Series D)):35–45, 1960.
- N. Kasabov. *Evolving Connectionist Systems: The Knowledge Engineering Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. ISBN 1846283450.
- N. Kasabov and D. Filev. Evolving intelligent systems: Methods, learning, & applications. In *2006 International Symposium on Evolving Fuzzy Systems*, pages 8–18, Sept. 2006.

- N. Kasabov and S. M. O. Lee. Evolving fuzzy neural networks for supervised/unsupervised on-line knowledge-based learning. *IEEE Transactions on Systems, Man and Cybernetics*, 31: 902–918, 2001.
- N. Kasabov and Q. Song. DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 10(2):144–154, APR 2002.
- N. K. Kasabov. *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. MIT Press, Cambridge, MA, USA, 1996.
- E. Kim, M. Park, S. Kim, , and M. Park. A Transformed Input-Domain Approach to Fuzzy Modeling . *IEEE Transactions on Fuzzy Systems*, 6(4):596–604, 1998.
- J. R. Koza. *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, MA, USA, 1992. ISBN 0-262-11170-5.
- V. Kreinovich, M. G., and H. Nguyen. Fuzzy rule-based modeling as universal approximation tool. In H. Nguyen and M. Sugeno, editors, *Fuzzy Systems: Modeling and Control*, pages 135–195. Kluwer Academic, Boston, MA, 1998.
- T. Y. Kwok and D. Y. Yeung. Constructive algorithms for structure learning in feedforward neural networks for regression problems. *IEEE Transactions on Neural Networks*, 8(3):630–645, MAY 1997. ISSN 1045-9227.
- C. Laurentys, R. Palhares, and W. Caminhas. A novel artificial immune system for fault behavior detection. *Expert Systems with Applications*, In Press, Corrected Proof:–, 2010a. ISSN 0957-4174. doi: DOI:10.1016/j.eswa.2010.12.019.
- C. Laurentys, R. Palhares, and W. Caminhas. Design of an artificial immune system based on danger model for fault detection. *Expert Systems with Applications*, 37(7):5145 – 5152, 2010b. ISSN 0957-4174. doi: DOI:10.1016/j.eswa.2009.12.079.
- C. Laurentys, G. Ronacher, R. Palhares, and W. Caminhas. Design of an artificial immune system for fault detection: A negative selection approach. *Expert Systems with Applications*, 37(7):5507 – 5513, 2010c. ISSN 0957-4174. doi: DOI:10.1016/j.eswa.2010.02.004.
- A. Lemos, W. Caminhas, and F. Gomide. New uninorm-based neuron model and fuzzy neural networks. In *2010 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, pages 1 –6, July 2010a. doi: 10.1109/NAFIPS.2010.5548195.
- A. Lemos, W. Caminhas, and F. Gomide. Evolving fuzzy linear regression trees. In *2010 IEEE International Conference on Fuzzy Systems (FUZZ)*, pages 1 –8, july 2010b. doi: 10.1109/FUZZY.2010.5583970.

- A. Lemos, W. Caminhas, and F. Gomide. Fuzzy multivariable gaussian evolving approach for fault detection and diagnosis. In E. Hullermeier, R. Kruse, and F. Hoffmann, editors, *Computational Intelligence for Knowledge-Based Systems Design*, volume 6178 of *Lecture Notes in Computer Science*, pages 360–369. Springer Berlin / Heidelberg, 2010c. doi: 10.1007/978-3-642-14049-5\_37.
- A. Lemos, F. Gomide, and W. Caminhas. Fuzzy evolving linear regression trees. *Evolving Systems*, In Press, Accepted Manuscript, 2011a. doi: 10.1007/s12530-011-9028-z.
- A. Lemos, V. Kreinovich, W. Caminhas, and F. Gomide. Universal approximation with uninorm-based fuzzy neural networks. In *2011 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, pages 1–6, March 2011b.
- A. Lemos, F. Gomide, and W. Caminhas. Multivariable gaussian evolving fuzzy modeling system. *IEEE Transactions on Fuzzy Systems*19(1):91–104, Jan. 2011c. ISSN 1063-6706. doi: 10.1109/TFUZZ.2010.2087381.
- G. Leng, T. McGinnity, and G. Prasad. An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network. *Fuzzy Sets and Systems*, 150(2):211–243, Mar 2005.
- Y. Li, L. qun Xu, J. Morphett, and R. Jacobs. On incremental and robust subspace learning. *Pattern Recognition*, 37:1509–1518, 2003.
- E. Lima, M. Hell, R. Ballini, and F. Gomide. Evolving fuzzy modeling using participatory learning. In P. Angelov, D. Filev, and N. Kasabov, editors, *Evolving Intelligent Systems: Methodology and Applications*. Wiley-Interscience/IEEE Press, 2009.
- R. Lippmann. A critical overview of neural network pattern classifiers. In *Proceedings of the 1991 IEEE Workshop Neural Networks for Signal Processing*, pages 266–275, 1991. ISBN 0-7803-0118-8. doi: 10.1109/NNSP.1991.239515.
- L. Ljung. *System Identification*. Prentice-Hall, 1999.
- D. G. Luenberger. Observing the state of a linear system. *IEEE Transactions on Military Electronics*, 8(2):74–80, apr. 1964.
- E. Lughofer. Extensions of vector quantization for incremental clustering. *Pattern Recognition*, 41(3):995 – 1011, 2008. Part Special issue: Feature Generation and Machine Learning for Robust Multimodal Biometrics.
- E. Lughofer. On dynamic soft dimension reduction in evolving fuzzy classifiers. In E. Hullermeier, R. Kruse, and F. Hoffmann, editors, *Computational Intelligence for Knowledge-Based Systems Design*, volume 6178 of *Lecture Notes in Computer Science*, pages 79–88. Springer, 2010. ISBN 978-3-642-14048-8.
- E. Lughofer. On-line incremental feature weighting in evolving fuzzy classifiers. *Fuzzy Sets and Systems*, 163(1):1 – 23, 2011. ISSN 0165-0114. doi: DOI:10.1016/j.fss.2010.08.012.

- E. Lughofer and C. Guardiola. On-line fault detection with data-driven evolving fuzzy models. *Control and intelligent systems*, 36(4):307–317, 2008.
- E. Lughofer, P. Angelov, and X. Zhou. Evolving single- and multi-model fuzzy classifiers with flexfis-class. In *IEEE International Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007*, pages 1–6, july 2007. doi: 10.1109/FUZZY.2007.4295393.
- E. D. Lughofer. FLEXFIS: A Robust Incremental Learning Approach for Evolving Takagi-Sugeno Fuzzy Models. *IEEE Transactions on Fuzzy Systems*, 16(6):1393–1410, DEC 2008a.
- E. D. Lughofer. Extensions of Vector Quantization for Incremental Clustering. *Pattern Recognition*, 41(3):995–1011, 2008b.
- J. J. Macias-Hernandez, P. Angelov, and X. Zhou. Soft sensor for predicting crude oil distillation side streams using takagi sugeno evolving fuzzy models. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 3305–3310, 2007.
- E. Mamdani and S. Assilian. Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7(1):1–13, 1975.
- M. Markou and S. Singh. Novelty detection: A review part 1: Statistical approaches. *Signal Processing*, 83:2499–2521, 2003a.
- M. Markou and S. Singh. Novelty detection: A review part 2: Neural networks based approaches. *Signal Processing*, 83:2481–2497, 2003b.
- N. Mehranbod, M. Soroush, and C. Panjapornpon. A method of sensor fault detection and identification. *Journal of Process Control*, 15(3):321–339, Apr 2005.
- L. F. Mendonca, J. M. C. Sousa, and J. M. G. S. da Costa. An architecture for fault detection and isolation based on fuzzy methods. *Expert Systems with Applications*, 36(2, Part 1): 1092–1104, Mar 2009.
- R. Miller. *Simultaneous statistical inference*. McGraw-Hill, Inc., New York, NY, USA, 1966.
- D. Montgomery. *Introduction to Statistical Quality Control*. Wiley, fourth edition, 2001.
- O. Nelles. *Nonlinear system identification*. Springer Verlag, Berlin, Germany, 2001.
- A. Nikranjbar, M. Ebrahimi, and A. S. Wood. Model-based fault diagnosis of induction motor eccentricity using particle swarm optimization. *Proceedings of the Institution of Mechanical Engineers Part C-Journal of Mechanical Engineering Science*, 223(3):607–615, MAR 2009.
- J. V. d. Oliveira and W. Pedrycz. *Advances in Fuzzy Clustering and its Applications*. John Wiley & Sons, Inc., New York, NY, USA, 2007.
- S. Ozawa, S. Pang, and N. Kasabov. Online feature extraction for evolving intelligent systems. In P. Angelov, D. Filev, and N. Kasabov, editors, *Evolving Intelligent Systems: Methodology and Applications*. Wiley-Interscience/IEEE Press, 2010.

- A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Inc., 2nd edition, 1984.
- W. Pedrycz and F. Gomide. *Fuzzy Systems Engineering: Toward Human-Centric Computing*. Wiley Interscience, NJ, USA, 2007.
- W. Pedrycz and K. Hirota. Uninorm-based logic neurons as adaptive and interpretable processing constructs. *Soft Computing*, 11(1):41–52, JAN 2007. ISSN 1432-7643. doi: {10.1007/s00500-006-0051-0}.
- D. Potts. Incremental learning of linear model trees. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 84, New York, NY, USA, 2004. ACM. ISBN 1-58113-828-5. doi: <http://doi.acm.org/10.1145/1015330.1015372>.
- F. Pouzols and A. Lendasse. Evolving fuzzy optimally pruned extreme learning machine for regression problems. *Evolving Systems*, 1:43–58, 2010. ISSN 1868-6478. doi: 10.1007/s12530-010-9005-y.
- R. Quinlan. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*, pages 236–243, 1992.
- V. Rajagopalan and A. Ray. Symbolic time series analysis via wavelet-based partitioning. *Signal Processing*, 86(11):3309 – 3320, 2006.
- J. d. J. Rubio. Sofmls: Online self-organizing fuzzy modified least-squares network. *IEEE Transactions on Fuzzy Systems*, 17(6):1296–1309, Dec 2009. ISSN 1063-6706. doi: 10.1109/TFUZZ.2009.2029569.
- B. Samanta. Gear fault detection using artificial neural networks and support vector machines with genetic algorithms. *Mechanical Systems and Signal Processing*, 18(3):625–644, MAY 2004.
- L. Silva. Aprendizagem participativa em agrupamento nebuloso de dados. Master’s thesis, Universidade Estadual de Campinas, Campinas, Brazil, 2003.
- L. Silva, F. Gomide, and R. Yager. Participatory learning in fuzzy clustering. In *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, pages 857–861, May 2005.
- T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, 1985.
- M. Timusk, M. Lipsett, and C. K. Mechefske. Fault detection using transient machine signals. *Mechanical Systems and Signal Processing*, 22(7):1724–1749, 2008.
- L. Torgo. Functional models for regression tree leaves. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 385–393, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1-55860-486-3.

- V. Venkatasubramanian, R. Rengaswamy, S. Kavuri, and K. Yin. A review of process fault detection and diagnosis Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 27(3):293–311, 2003a.
- V. Venkatasubramanian, R. Rengaswamy, S. Kavuri, and K. Yin. A review of process fault detection and diagnosis Part II: Quantitative model and search strategies. *Computers & Chemical Engineering*, 27(3):313–3265, 2003b.
- V. Venkatasubramanian, R. Rengaswamy, S. Kavuri, and K. Yin. A review of process fault detection and diagnosis Part III: Process history based methods. *Computers & Chemical Engineering*, 27(3):327–346, 2003c.
- D. Wang, X. Zeng, and J. Keane. A structure evolved learning method for mamdani fuzzy systems. In *AISB-EIS 10*, pages 1068–1073, 2010.
- L. Wang and J. Mendel. Generating Fuzzy Rules by Learning from Examples. *IEEE Transactions on Systems Man and Cybernetics*, 22(6):1414–1427, Nov-Dec 1992a.
- L. Wang and J. Mendel. Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least-Squares Learning. *IEEE Transactions on Neural Networks*, 3(5):807–814, Sep 1992b.
- W. Wang and J. Vrbanek. An evolving fuzzy predictor for industrial applications. *IEEE Transactions on Fuzzy Systems*, 16(6):1439–1449, Dec. 2008.
- J. R. Williamson. Gaussian ARTMAP: A neural network for past incremental learning of noisy multidimensional maps. *Neural Networks*, 9(5):881–897, JUL 1996. ISSN 0893-6080.
- M. Wong, L. Jack, and A. Nandi. Modified self-organising map for automated novelty detection applied to vibration signal monitoring. *Mechanical Systems and Signal Processing*, 20(3):593–610, 2006.
- R. Yager. A Model of Participatory Learning. *IEEE Transactions on Systems Man and Cybernetics*, 20(5):1229–1234, SEP-OCT 1990.
- R. Yager. Participatory learning: a paradigm for more human like learning. In *2004 IEEE International Conference on Fuzzy Systems, 2004*, volume 1, pages 79–84 vol.1, July 2004.
- R. Yager and D. Filev. Approximate Clustering via The Mountain Method. *IEEE Transactions on Systems Man and Cybernetics*, 24(8):1279–1284, AUG 1994.
- R. R. Yager. Extending the participatory learning paradigm to include source credibility. *Fuzzy Optimization and Decision Making*, 6(2):85–97, 2007.
- P. Young. *Recursive estimation and time-series analysis: an introduction*. Springer-Verlag New York, Inc., New York, NY, USA, 1984. ISBN 0-387-13677-0.
- Y. Yuan and M. Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, 69(2):125–139, 1995.

X. Zhou and P. Angelov. Autonomous visual self-localization in completely unknown environment using evolving fuzzy rule-based classifier. In *IEEE Symposium on Computational Intelligence in Security and Defense Applications, 2007. CISDA 2007.*, pages 131–138, April 2007. doi: 10.1109/CISDA.2007.368145.