Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Engenharia de Computação e Automação Industrial

Modelo de Previsão Baseado em Agrupamento e Base de Regras Nebulosas

Giselle Cristina Cardoso

Orientador: Prof. Dr. Fernando Antônio Campos Gomide

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como requisito parcial exigido para a obtenção do título de Mestre em Engenharia Elétrica.

Banca Examinadora: Prof. Dr. Fernando Antônio Campos Gomide
(FEEC/UNICAMP)

Profa. Dra. Sandra Sandri
(INPE)

Prof. Dr. José Raimundo Oliveira
(FEEC/UNICAMP)

Profa. Dra. Rosangela Ballini
(IE/UNICAMP)

Dissertação de Mestrado
Campinas – SP – Brasil
Maio – 2003

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

C179m

Cardoso, Giselle Cristina
Modelo de previsão baseado em agrupamento e base de regras nebulosas / Giselle Cristina Cardoso.-- Campinas, SP: [s.n.], 2003.

Orientador: Fernando Antônio Campos Gomide.
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Inteligência artificial – Processamento de dados. 2. Sistemas difusos. 3. Redes neurais (Computação). 4. Analise de séries temporais – Processamento de dados. 5. Jornais – Previsão. I. Gomide, Fernando Antônio Campos. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Resumo

Um problema enfrentado diariamente por muitas empresas jornalísticas é o de determinar a quantidade de jornais que devem ser impressos e distribuídos entre os numerosos pontos de vendas, visando minimizar perdas e maximizar vendas. A quantidade certa que deve ser reposta depende de vários fatores, especialmente da demanda de cada ponto de venda a qual, por sua vez, depende de sua localização. Atualmente, a previsão da demanda de cada ponto de venda é baseada em taxas de reposição observadas no passado e por um especialista da área. Este trabalho propõe o uso de *Knowledge Discovery in Databases* como uma técnica de previsão de reposição. O objetivo é prever a quantidade de jornais que devem ser repostos diariamente em cada uma das bancas de jornal. O modelo de previsão proposto utiliza agrupamento nebuloso para a exploração dos dados e regras nebulosas para a previsão. Os resultados experimentais obtidos com uma base de dados real mostram a eficácia do modelo, especialmente quando comparados com a metodologia atual e com os resultados proporcionados por métodos de previsão baseados em reposição e em redes neurais.

Abstract

A problem faced daily by most newspaper companies is how to determine the amount of newspaper to be printed and distributed among numerous selling points to minimize losses and maximize sales. The right amount that must be replaced depends on several features, especially the demand at each selling point, a function of its location. Currently, demand prediction uses replacement rates based on past data analysis and expert knowledge. This work proposes the use of knowledge discovery and predictive data mining techniques to predict the amount of newspaper to be delivered daily at each newsstand of a region. The prediction model uses fuzzy clustering for data exploration and fuzzy rules for prediction. Experimental results obtained with an actual data newspaper base show the effectiveness of the model, especially when compared with the current methodology, regression and a neural network-based predictor.

Agradecimentos

Agradeço a Deus, sem o qual não seria possível a conclusão de mais esta etapa de minha vida.

Ao Prof. Fernando Gomide pela orientação segura, pelo aprendizado propiciado e pela confiança que me passou ao longo do desenvolvimento deste trabalho.

Aos meus pais, irmãos e sobrinhos por terem me dado amor, carinho e incentivo em todos os momentos.

Aos amigos e companheiros Leila, Ivette, Rosana, Marina, Ivana, Igor, Rachel e Michel, pelas sugestões e pela amizade; e a todos do LCA, dos outros laboratórios da FEEC - Unicamp e os demais que passaram por lá e deixaram lembranças; meus sinceros agradecimentos.

Ao Pedro Neves Júnior pela motivação e contribuição.

Aos membros da banca: Profa. Dra. Rosangela Ballini, Prof. Dr. José Raimundo Oliveira e Profa. Dra. Sandra Sandri, agradeço a participação e contribuições.

Agradeço à Unicamp, particularmente a FEEC pela oportunidade de realizar esta pesquisa.

À CAPES pelo suporte financeiro.

A todos meus tios e primos, meu muito obrigado pela força, ajuda e carinho.

A todos meus amigos e colegas, que de maneira direta ou indireta contribuíram e me incentivaram nesta etapa.

Aos professores do Departamento de Ciência da Computação da PUC-MG, campus de Poços de Caldas, especialmente os professores: Iran, Márcio e Adriana, pelo apoio e incentivo de sempre.

Aos que não citei, mas ajudaram na conclusão deste trabalho, muito obrigado.

Dedico esta dissertação a Deus,
à Nossa Senhora
e a meus pais.

Índice

Lista de Figuras

Lista de Tabelas

Notação

X – conjunto finito de dados $\{x_1,...,x_n\}, x_k \in R^p$, $k = 1,...,n$

x_k – k-ésimo elemento de X, $k = 1, \ldots, n$

p – número de componentes de x_k

N – número de elementos de X

C – número de grupos

x_{ki} – i-ésimo componente de x_k, $i = 1, \ldots, p$

U – matriz de pertinência ($c \times n$)

u_{ik} – elemento da matriz U, $i = 1, \ldots, c$, $k = 1, \ldots, n$

V – conjunto dos centros dos c grupos $\{v_1, ..., v_c\} \subset R^p$

v_{ij} – elemento da matriz V, $i = 1, \ldots, c$, $j = 1, \ldots, p$ que representa o conjunto V

$j^* = \arg\max_j \{f_j\}$ – denota que o índice j^* corresponde aquele associado ao maior valor de f_j

RD_{ij} – desempenho nebuloso do j-ésimo atributo no i-ésimo grupo

SC_k – pontuação do k-ésimo atributo

det (A) – valor do determinante da matriz A

Lista de Abreviaturas

KDD	: *Knowledge Discovery in Databases*
FCM	: *Fuzzy C-Means*
E-FCM	: *Extended Fuzzy C-Means*
AID	: *Automatic Interaction Detection*
CHAID	: *Chi Square Automatic Interaction Detection*
ID3	: *Iterative Dichotomizer 3rd*
AR	: Algoritmo Autoregressivo

CAPÍTULO 1

INTRODUÇÃO

1.1 Motivação

Atualmente, a armazenagem de uma grande quantidade de dados necessária à processos de transações de informações vem criando bases de dados razoavelmente grandes que guardam uma grande quantidade de informações não acessíveis através de consultas feitas aos bancos de dados pelos métodos tradicionais. O *Knowledge Discovery in Databases* é uma maneira de se explorar essas bases de dados, com o objetivo de obter os padrões desconhecidos existentes.

O *Knowledge Discovery in Databases* tem sido muito utilizada na área financeira de acordo com Bose e Mahapatra (2001), sendo empregada para:

- prever o futuro das finanças, principalmente as relacionadas aos sistemas bancários, sendo esta a maior aplicação na categoria do mercado financeiro;
- prever a falência tanto de clientes como do próprio banco;
- estimar a confiança que se pode ter em um usuário de cartão de crédito;
- obter regras de negócios;
- aprovar e detectar fraudes em cartão de crédito;
- classificar os consumidores.

Esta tecnologia é muito utilizada no mercado financeiro pelo poder de investimento que essa área possui e também pela facilidade de criar bases de dados relativamente grandes.

No mercado de negócios essa tecnologia é utilizada para:

- procurar clientes alvos para determinados produtos e promoções, com o objetivo de enviar propagandas (Kaymak e Setnes, 2001);
- análise de vendas e de mercado (Bose e Mahapatra, 2001);
- análise do desempenho do produto no mercado (Bose e Mahapatra, 2001);
- análise de segmentação de mercado (Bose e Mahapatra, 2001).

Em telecomunicações, baseando-se em Bose e Mahapatra (2001), essa tecnologia é aplicada para:

- detectar fraudes;
- prever o comportamento de redes;
- analisar chamadas;
- detectar telefones clonados; entre outras.

A medicina é uma outra área que a tecnologia de extração de conhecimento de base de dados tem sido aplicada, principalmente para:

- traçar sintomas de pacientes em procedimentos cirúrgicos (Bose e Mahapatra 2001);
- extração de regras para diagnóstico de pacientes (Richards *et al*.,2001);
- prever doenças cardíacas (Chae *et al.*, 2001);
- monitorar problemas de hipertensão (Chae *et al.*, 2001).

Outras áreas, segundo Bose e Mahapatra (2001), em que o *Knowledge Discovery in Databases* tem sido muito aplicada, incluem:

- escalonamento;
- controle de qualidade de software;
- estimação do custo de software; entre outras.

Entre muitas áreas e aplicações a tecnologia de extração de conhecimento de banco de dados vem crescendo rapidamente, principalmente em aplicações destinadas à área de negócios organizacionais.

Em relação à distribuição de jornais, existem trabalhos, como o de Fleischfresser, 2001 e Ree e Yoon, 1996, que sugerem a otimização do serviço de entrega de jornais aos assinantes, reduzindo a distância total percorrida pelos entregadores de jornal, mas mantendo a qualidade e rapidez. O trabalho de Buer *et al.*, 1999, além de melhorar os serviços de entrega aos assinantes, avalia quantos veículos devem ser utilizados na entrega e quantos poderão ser reaproveitados para fazer outras entregas, de acordo com o tempo que cada um necessita para realizar a sua tarefa. Quanto a trabalhos relacionados à distribuição de jornais com o objetivo de aumentar as vendas e diminuir as perdas diárias nas bancas de jornal, não foi encontrado nenhum trabalho na literatura.

1.2 Objetivo

O objetivo principal deste trabalho é estudar o uso de técnicas de extração de conhecimento de base de dados em problemas de decisão. Para tal, considerou-se o problema de previsão de reposição diária de jornais em bancas com o propósito de maximizar vendas e minimizar perdas mediante a escolha de quantidades adequadas de reposição (Cardoso e Gomide, 2003a, 2003b).

Para atingir este objetivo, foram analisados vários métodos visando obter uma solução para o problema e avaliar a viabilidade prática das soluções obtidas. Em particular, explorou-se o método de Kaymak e Setnes (2001), pelo fato deste permitir a considerações de incertezas no contexto da teoria de sistemas nebulosos.

1.3 Organização

Este trabalho se divide em cinco capítulos. Neste capítulo apresenta-se a motivação para o desenvolvimento deste trabalho e o seu objetivo. O Capítulo 2 descreve

os fundamentos metodológicos necessários ao desenvolvimento deste trabalho. O Capítulo 3 discute metodologias de previsão que serão aplicadas para a solução do problema. O Capítulo 4 detalha o uso da metodologia de extração de conhecimento de base de dados de Kaymak e Setnes (2001) na solução do problema de previsão para reposição de jornais na metodologia de previsão de reposição. Neste capítulo, também, são feitas comparações com métodos clássicos de previsão com os resultados obtidos com a sua aplicação. Finalmente, no Capítulo 5 apresentam-se as conclusões e sugerem-se itens para trabalhos futuros.

CAPÍTULO 2

FUNDAMENTOS METODOLÓGICOS

2.1 Introdução

Este capítulo aborda as propriedades, fases e características da tecnologia de *Knowledge Discovery in Databases*, detalhando seu processo e uma de suas fases, a mineração de dados. A mineração de dados tem como objetivo procurar padrões em uma base de dados. A busca por padrões em bases de dados requer o uso de algoritmos para classificação, estimação e associação. A associação pode ser obtida através de regras ou agrupamento de dados. Entre os vários algoritmos usados na fase de mineração de dados incluem-se os baseados na teoria de conjuntos nebulosos. Estes algoritmos são particularmente apropriados para agrupamento de dados, destacando-se o *fuzzy c-means* (Bezdek, 1981, Klir e Yuan, 1993), *extended fuzzy c-means* (Kaymak e Setnes, 2000, Kaymak e Setnes, 2001) e o algoritmo de agrupamento participativo (Silva, 2003). Os algoritmos clássicos de classificação baseiam-se em árvores de decisão, destacando-se entre eles o *Automatic Interaction Detection* (Morgan e Sonquist, 1963, Neville, 1999, Cadiz 1994), o *Chi Square Automatic Interaction Detection* (Kass, 1975) e o *Iterative Dichotomizer 3rd* –ID3 (Quinlan, 1983, Mitchell, 1997, Neville, 1999).

Este capítulo também trata dos conceitos básicos de sistemas baseados em regras nebulosas. Devido à sua natureza lingüística, estes sistemas contribuem para a representação e interpretação de conhecimento extraído da base de dados e na utilização em problemas de tomada de decisão.

2.2 Knowledge Discovery in Databases

Nas últimas décadas, a tecnologia permitiu a armazenagem de grandes quantidades de dados tanto em sistemas de informação tradicionais (transações bancárias, registros de compras, etc.), quanto em sistemas inovadores (internet, integração de informações de diversos sistemas, etc.). Paralelamente, as bases de dados vêm atingindo grandes proporções e complexidade, criando um desafio considerável para a aquisição, representação e compreensão do conhecimento nelas contido.

Apesar das novas técnicas que vem sendo propostas para o estudo de base de dados nos moldes tradicionais, na prática estas novas tecnologias têm se mostrado limitadas quando se trata de sistemas complexos. A cada dia que passa, a quantidade de informação ultrapassa a capacidade de análise proporcionada por métodos tradicionais (planilhas, consultas e gráficos). Esses métodos podem gerar relatórios a partir dos dados, mas não conseguem analisá-los para se obter o conhecimento neles contidos. A necessidade de explicitar o conhecimento contido em uma base de dados induziu a pesquisa e o desenvolvimento de novas técnicas e ferramentas que permitissem a extração de conhecimento destas bases de dados. Em 1989, surgiu um novo ramo da computação que se tornou conhecido como Knowledge Discovery in Databases – KDD. O KDD tem como objetivo otimizar e automatizar o processo de descrição das tendências e dos padrões contidos em bases de dados e enfatizar o alto nível das aplicações dos métodos de mineração de dados (Sade, 1996).

A metodologia de KDD utiliza uma nova geração de hardware e software e baseia-se em métodos de análise estatística, visualização de dados, árvores de decisão e inteligência computacional, dentre outros, sendo estes os principais para explorar bases de dados e descobrir relações e padrões nelas existentes. Em aplicações, o KDD pode ser vista como uma forma de selecionar, explorar e modelar conjuntos de dados para detectar, por exemplo, padrões de consumo e proporcionar uma ferramenta de auxílio na elaboração de estratégias de marketing, estratégias de racionalização no uso de recursos naturais, etc.

A tecnologia KDD é definida como um processo que automatiza a identificação e o reconhecimento de padrões em base de dados. O KDD constitui hoje uma área de pesquisa e desenvolvimento cuja expansão tornou-se mais pronunciada nos últimos anos. Segundo Frawley *et al.* (1991), Mannila (1996), Fayyad *et al.* (1996) e Lee *et al.* (2001), sua principal característica é a extração não-trivial de informações a partir de uma base de dados**.** Essas são informações que, implicitamente contidas na base de dados, são difíceis de serem detectadas somente via métodos clássicos de análise, mas são informações potencialmente úteis para tomada de decisão.

2.2.1 O processo de KDD

Devido as suas características, o processo de KDD depende das metodologias e técnicas de análise de dados e envolve diversas fases ou etapas. Segundo Mannila (1996, 1997), Amaral (2001), Han e Kamber (2001), o núcleo do processo é muitas vezes confundido com o próprio processo de KDD, a chamada mineração de dados, ou *data mining*. A mineração de dados também é referida na literatura como processamento de padrões de dados, arqueologia de dados, descoberta de conhecimento em bases de dados, descoberta de informação e colheita de informação (Han e Kamber, 2001). Esse processo é composto por um conjunto de atividades que compartilham o conhecimento descoberto a partir de bases de dados, começando com o entendimento do domínio da aplicação e dos objetivos finais a serem atingidos.

Como descreve Amaral (2001), o processo de KDD envolve duas fases principais, sendo a primeira a fase de preparação de dados e a segunda a fase de mineração de dados. Cada uma destas duas fases constitui-se, por sua vez, de subfases conforme mostra a Figura 2.1 (Amaral, 2001, Han e Kamber, 2001 e Kaymak e Setnes, 2001).

A primeira fase, chamada preparação de dados, realiza a maior parte do trabalho, sendo por isso considerada muito importante, pois é nesta fase que os dados necessários para a solução do problema são selecionados. Esta fase inicia com um agrupamento organizado de uma massa de dados, selecionando aqueles que são relevantes e, portanto, alvo da exploração. A seguir é feita a limpeza dos dados através de um pré-processamento, visando adequá-los aos algoritmos considerados. Isso se faz através da

integração de dados heterogêneos, eliminação de dados incompletos, repetição de tuplas, problemas de tipagem, etc. Essa etapa consome grande parte do esforço necessário para todo o processo, devido principalmente às dificuldades causadas pela integração de bases de dados heterogêneas. Os dados pré-processados devem também passar por uma transformação que os coloque e armazene em uma forma adequada, visando facilitar o uso das técnicas de mineração de dados.

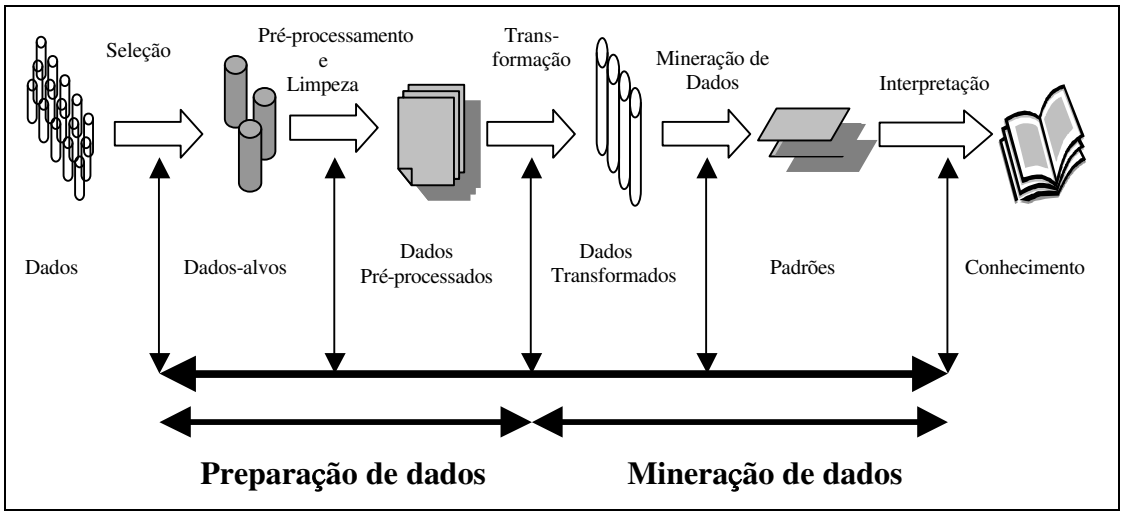


Figura 2.1: Fases e subfases do processo de KDD.

Prosseguindo o processo, inicia-se a fase de mineração de dados propriamente dita pela escolha dos algoritmos a serem aplicados. Essa escolha depende fundamentalmente dos objetivos do processo de KDD, entre eles classificação, regressão, previsão de séries temporais, detecção de desvios, segmentação de base de dados, agrupamento, regras associativas, sumarização, visualização e mineração de textos. De modo geral, na fase de mineração de dados as ferramentas computacionais utilizam algoritmos e metodologias especializadas para a busca por padrões. Essa busca pode ser efetuada ou automaticamente pelo sistema, ou iterativamente via um analista responsável pela geração de hipóteses. Atualmente, existem inúmeras ferramentas capazes de realizar e/ou auxiliar esta busca, muitas delas baseadas em metodologias e algoritmos de redes neurais, indução de árvores de decisão, sistemas baseados em regras e programas estatísticos. Estas metodologias e algoritmos se manifestam tanto isoladamente quanto em

combinação, traduzindo-se em ferramentas híbridas de KDD. Em geral, o processo de busca é iterativo, de forma que os analistas revêem o resultado, formam um novo conjunto de questões para refinar a busca em um dado aspecto das descobertas e realimentam o sistema com novos parâmetros. Ao final do processo, o sistema de mineração de dados gera um relatório das descobertas, que passa então a ser interpretado por analistas de mineração. Somente após esta interpretação obtém-se o conhecimento.

Observa-se que o processo de KDD requer etapas de pré e pós-processamento de dados, etapas estas necessárias tanto para assegurar o melhor aproveitamento dos dados tendo em vista a aplicação, quanto à consistência dos resultados. Como vimos, atividades de pré-processamento incluem a seleção apropriada de subconjuntos de dados, por razões de desempenho e de relevância para a aplicação, assim como complexas transformações de dados que servem de ponte para a separação entre os dados e seu significado real. O pós-processamento envolve a seleção e a análise de resultados, bem como a aplicação de técnicas de visualização para auxiliar o entendimento do conhecimento gerado pela mineração de dados.

2.2.2 Mineração de dados

Muitas das metodologias, algoritmos e técnicas utilizadas em mineração de dados se originaram na pesquisa em inteligência artificial da década de 80 e princípios da década de 90. Entretanto, de acordo com Mannila (1996, 1997), essas técnicas somente passaram a ser enfatizadas e utilizadas em sistemas de banco de dados por aumentar o valor da informação.

Mannila (1996) afirma que, em cada aplicação da técnica de mineração de dados, existe um conjunto de métodos e algoritmos que são os candidatos potenciais para a extração de relações relevantes implícitas em base de dados. Entre eles incluem-se métodos e algoritmos de análise de seqüências, agrupamento de dados, classificação, estimativas, regras de associação e, mais recentemente, técnicas que utilizam a teoria de conjuntos nebulosos e algoritmos genéticos (Han e Kamber, 2001). Cada um destes candidatos pode ser utilizado nos diferentes tipos de problemas relacionados com a aplicação em mente.

A mineração de dados, segundo Indurkhya e Weiss (1998), divide-se em dois tipos. O primeiro, predição, utiliza dados históricos e respostas conhecidas. O segundo tipo, extração de conhecimento, é complementar à predição. A Tabela 2.1 mostra alguns métodos e algoritmos de acordo com o tipo de mineração de dados.

Tabela 2-1: Métodos e algoritmos de mineração de dados.

Predição	**Extração de conhecimento**
Classificação	Agrupamento
Regressão	Visualização
Séries Temporais	Regras de Associação
	Detecção de desvio
	Sumarização

Antes de se iniciar o processo de mineração de dados, deve-se decidir pelo tipo de algoritmo que será usado na aplicação. Se decidir pelo uso de um algoritmo de agrupamento de dados, ele será uma das primeiras etapas do processo de mineração de dados, já que sua função é identificar grupos de registros relacionados. Estes grupos serão usados em futuras explorações, como será explicado na seção 2.3.

Uma outra importante técnica, além do agrupamento, é a classificação de dados. A classificação consiste na utilização de um conjunto de exemplos pré-classificados para desenvolver um modelo capaz de classificar uma população maior de dados. Em geral, algoritmos de classificação incluem árvores de decisão ou redes neurais. A classificação através de árvores de decisão é abordada na seção 2.4.

Se os padrões encontrados pelo algoritmo de mineração de dados são eficientes, ele poderá ser usado de forma preditiva para classificar novos dados de acordo com os grupos definidos. Por exemplo, um classificador pode ser treinado a identificar empréstimos de risco a partir das informações cadastrais de milhares de interessados, e usado como suporte a decisão no momento de se conceder um empréstimo a alguém.

Existe, porém, uma diferença significativa entre a mineração de dados e os outros mecanismos de análise de dados, justamente na maneira como cada um deles exploram as relações existentes entre os dados que estão sendo analisados. Os diversos mecanismos

de análise dispõem de métodos baseados na verificação, isto é, o usuário constrói hipóteses sobre relações específicas e as verifica com o auxílio do próprio sistema. Esse modelo torna-se dependente da intuição e habilidade do analista em propor hipóteses interessantes, em manipular a complexidade do espaço de atributos, e em refinar a análise baseando-se em resultados de consultas à base de dados potencialmente complexas. No processo de mineração de dados, ele mesmo é responsável pela geração de hipóteses, garantindo maior rapidez, precisão e completeza aos resultados.

Como vemos, a mineração de dados é uma metodologia para encontrar uma descrição lógica ou matemática, eventualmente de natureza complexa, de padrões e regularidades em um determinado conjunto de dados.

Na seção seguinte, considera-se as características e os principais algoritmos nebulosos de agrupamento usados na mineração de dados.

2.3 Agrupamento de dados

Algoritmos de agrupamento de dados são uns dos mais relevantes em tarefas de mineração de dados. Sua função é identificar uma estrutura de grupos de dados correlacionados. A Figura 2.2, ilustra um agrupamento de dados feito em um conjunto de dados de clientes relativos a débitos pessoais, tentando identificar nesses dados eventuais clientes negligentes. Após o agrupamento de dados observam-se três grupos que determinaram o perfil de cada grupo de clientes. Estes grupos, posteriormente, serão fundamentais em futuras explorações onde serão usados como ponto de partida para análise.

O objetivo de algoritmos de agrupamento é identificar duas ou mais coleções de dados que formam uma estrutura de grupo e que possam ser associadas a classes. Em sistemas cujos dados contém um número grande de atributos, o algoritmo de agrupamento, ao particionar a base de dados em um conjunto de grupos, também proporciona um mecanismo de sumarização e compactação de informação.

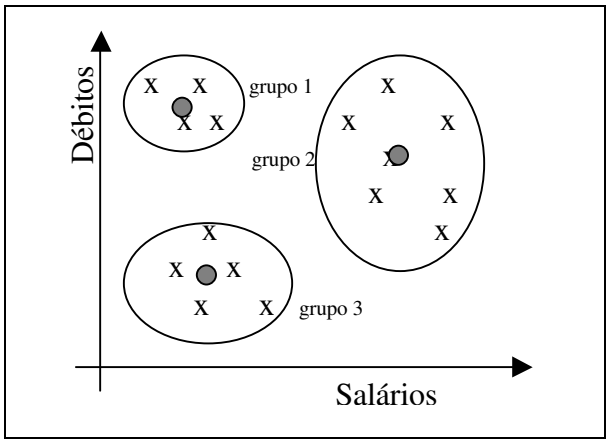


Figura 2.2: Agrupamento de dados.

Agrupamento de dados é um conceito comum em estatística mas, pelo fato de poder também ser generalizado a dados não necessariamente numéricos, ele também é de grande relevância em mineração de dados.

Os resultados obtidos em uma operação de agrupamento de dados podem ser usados tanto para verificar características de uma base de dados, como dados de entrada para outras técnicas, como por exemplo, a classificação, já que uma classe pode ser considerada associada a um grupo de mais fácil manuseio por parte de algoritmos de classificação.

Existem, também, algoritmos de agrupamento utilizados para organizar dados de acordo com o grau com que cada um desses dados é compatível com os demais elementos que formam os grupos. Estes algoritmos baseiam-se na teoria de conjuntos nebulosos e constituem generalizações dos algoritmos clássicos. Dentre muitos algoritmos de agrupamento de dados nebulosos, podemos citar o *fuzzy c-means* (Bezdek, 1981, Klir e Yuan, 1993), o *extended fuzzy c-means* (Kaymak e Setnes, 2000, Kaymak e Setnes, 2001) e o agrupamento participativo (Silva, 2003).

Atualmente, algoritmos de agrupamento de dados estão sendo utilizados em várias áreas do conhecimento, principalmente em economia, biologia, medicina, geografia, classificação de documentos, entre outras (Han e Kamber, 2001). Um exemplo clássico é o de classificação demográfica, que serve de início para a determinação das

características de um grupo social, visando desde hábitos de compras até utilização de meios de transporte e de comunicação.

2.3.1 Algoritmo *fuzzy c-means*

O algoritmo de agrupamento de dados nebuloso *fuzzy c-means* – FCM, proposto por Bezdek (1981), é um dos mais atrativos para este propósito, devido principalmente à sua simplicidade e eficiência. Este algoritmo é baseado na otimização iterativa de uma função objetivo que traduz um critério de partição ponderado por graus de pertinência dos dados aos respectivos grupos.

O algoritmo FCM (Bezdek, 1981), apresentado na Figura 2.3, utiliza os seguintes conceitos e notações. Seja $X = \{x_1, x_2, ..., x_n\}$ um conjunto finito de dados, genericamente chamados de pontos de R^p, onde R^p é um espaço Euclidiano p-dimensional; c o número de grupos, onde $2 \le c \le n$; U a matriz de pertinência $c \times n$, onde u_{ik}, com $1 \le i \le c$ e $1 \le k \le n$, denota o grau de pertinência do ponto x_k ao grupo i. Uma c-partição de X é definida por:

$$M_{fc} = \left\{ U \left| u_{ik} \in [0,1], \forall i,k; \sum_{i=1}^{c} u_{ik} = 1, \forall k; 0 < \sum_{k=1}^{n} u_{ik} < n, \forall i \right. \right\}. \quad (2.1)$$

O algoritmo FCM procura agrupar os dados através de um procedimento que tem como objetivo determinar o mínimo da seguinte função:

$$J_m(U,V) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m d_{ik}^2, U \in M_{fc}, 1 < m < \infty \quad (2.2)$$

onde $d_{ik} = \|x_k - v_i\|_A$ é a distância entre x_k e v_i, $\|.\|$ é uma norma induzida por um produto interno, como por exemplo $\|x\|_A^2 = x^T A x$, onde A é uma matriz $p \times p$ definida positiva; $V = \{v_1, v_2, ..., v_c\}$ é o conjunto de centros do grupo, representado por uma matriz $c \times p$, $v_i \in R^p$, $1 \le i \le c$; o ponto v_i é chamado de centro do i-ésimo grupo; e m é o fator que define o grau de nebulosidade da partição nebulosa do sistema.

Dado X, escolher o número de grupos $1 < c < n$, o parâmetro $m > 1$, o critério de parada $\varepsilon > 0$ e o número máximo de iterações $lmax$.

1. Inicializar $U^{(0)}$ e o contador de iterações l=1.

2. Calcular os c centros dos grupos $(v_1^{(l)}, v_2^{(l)}, ..., v_c^{(l)})$ usando $U^{(l)}$, com a equação (2.3):

$$v_i^{(l)} = \frac{\sum_{k=1}^{n} (u_{ik}^{l-1})^m x_k}{\sum_{k=1}^{n} (u_{ik}^{l-1})^m}, \; i = 1,2,...,c \; . \qquad (2.3)$$

3. Atualizar $U^{(l)}$ com o seguinte procedimento:

Para $1 \le k \le n$

Se $\left\| x_k - v_i^{(l)} \right\|^2 > 0$

$$u_{ik}^{l} = \left[\sum_{j=1}^{c} \left(\frac{\left\| x_k - v_i^{(t)} \right\|^2}{\left\| x_k - v_j^{(t)} \right\|^2} \right)^{\frac{1}{m-1}} \right]^{-1}, i = 1,2,...,c \; . \qquad (2.4)$$

Se $\left\| x_k - v_i^{(l)} \right\|^2 = 0$

$$u_{ik}^{l} = 1, \quad l \le i \le c \qquad (2.5)$$

4. Calcular $\Delta = \left\| U^{(l)} - U^{(l-1)} \right\| = max_{i,j} \left| u_{ij}^{(l)} - u_{ij}^{(l-1)} \right|$ (2.6)

Se $\Delta > \varepsilon$ ou l < $lmax$,

$l = l + 1$ e voltar ao passo 2.

Senão parar.

Figura 2.3: Algoritmo FCM.

2.3.2 Algoritmo *extended fuzzy c-means*

O algoritmo *extended fuzzy c-means* – E-FCM é um algoritmo de agrupamento nebuloso desenvolvido por Kaymak e Setnes (1998, 2000). A sua principal característica é a criação de grupos e a união de grupos semelhantes, formando novos grupos (Kaymak e Setnes, 2000) até que um número razoável de grupos seja encontrado.

Quando o E-FCM é inicializado, cria-se aleatoriamente um número suficientemente grande de grupos de acordo com n (número de pontos), começando a

seguir o agrupamento. O algoritmo, então, determina iterativamente se dois grupos possuem semelhanças. A similaridade entre dois grupos *A* e *B* é definida, como:

$$S(A,B) = max\left(\frac{|A \cap B|}{|A|}, \frac{|A \cap B|}{|B|}\right), \tag{2.7}$$

onde $|\,.\,|$ denota a cardinalidade de um conjunto nebuloso, e $\cap$ representa a interseção (Kaymak e Setnes, 2001, Klir e Yuan, 1993). A similaridade $S(A,B)$ é obtida pelo máximo entre o grau de inclusão de *A* em *B* e da inclusão de *B* em *A*.

O algoritmo E-FCM, apresentado na Figura 2.4, utiliza alguns parâmetros, como $\alpha \in [0,1]$ o limiar de similaridade o qual determinará se dois grupos devem se unir ou não. De acordo com Kaymak e Setnes (2000), em vários problemas pode-se usar um limiar adaptativo, sendo que o seu valor irá depender do número de grupos existentes. Este limiar pode ser estimado por $\alpha = 1/(c-1)$, onde *c* é o número de grupos existentes naquele determinado momento em que o limiar está sendo calculado.

Os conceitos e notações utilizados pelo algoritmo E-FCM são os mesmos apresentados para o algoritmo FCM, na seção 2.3.1, exceto o parâmetro β que controla o do número de centros, *P*, que é uma matriz de covariância ($p \times p$) e *r* que é o raio em relação ao centro *i*.

2.3.3 Algoritmo de agrupamento participativo

Este algoritmo de agrupamento de dados, introduzido por Silva (2003), foi inspirado nos conceitos de aprendizagem participativa proposto por Yager (1990), o qual permite representar o aprendizado com a participação das crenças de um sistema.

Na abordagem de agrupamento de dados, um dos problemas fundamentais é a estimação do número de grupos. Dentre os algoritmos de agrupamentos, um algoritmo capaz de encontrar um número razoável de grupos em um conjunto de dados é chamado de algoritmo não supervisionado.

Dado X, escolher o número inicial de grupos $1 < c^{(0)} < n$, o parâmetro $m > 1$, o critério de parada $\varepsilon > 0$ e o número máximo de iterações $lmax$.
Inicializar $U^{(0)}$ aleatoriamente, $S_{i^*j^*}^{(0)} = 1$ e $\beta^{(0)} = 1$.

Repetir para $l = 1,2,...$

1. Calcular os centros dos grupos, com a equação (2.3).
2. Calcular a covariância e o raio:

$$P_i = \frac{\sum_{k=1}^{n} (u_{ik}^{(l-1)})^m (x_k - v_i^{(l)})(x_k - v_i^{(l)})^T}{\sum_{k=1}^{N} (u_{ik}^{(l-1)})^m}, \quad 1 \le i \le c^{(l-1)}. \tag{2.8}$$

$$r_i = \frac{\beta^{(l-1)} \sqrt{[det(P_i)]^{1/n}}}{c^{(l-1)}}, \quad 1 \le i \le c^{(l-1)} \quad . \tag{2.9}$$

3. Calcular a distância:

$$d_{ik} = max(0, \sqrt{(x_k - v_i^{(l)})^T (x_k - v_i^{(l)})} - r_i), \quad 1 \le i \le c^{(l-1)}, \quad 1 \le k \le n. \tag{2.10}$$

Atualizar a matriz de pertinência:

Para $1 \le k \le n$, $\phi_k = \{i \mid d_{ik} = 0\}$

Se $\phi_k = 0$, calcule $u_{ik}^{(l)}$ com a equação (2.11):

$$u_{ik}^{l} = \frac{1}{\sum_{j=1}^{c} (d_{ik} / d_{jk})^{2/(m-1)}}, 1 \le i \le c^{(l-1)}. \tag{2.11}$$

Senão $u_{ik}^{l} = \begin{cases} 0 & se\ d_{ik} > 0 \\ 1/|\phi_k| & se\ d_{ik} = 0 \end{cases}, 1 \le i \le c^{(l-1)}.$ (2.12)

4. Selecionar os pares de grupos mais similares:

$$S_{ij}^{(l)} = \frac{\sum_{k=1}^{n} min(u_{ik}^{(l)}, u_{jk}^{(l)})}{\sum_{k=1}^{N} u_{ik}^{(l)}}, \quad 1 \le i, j \le c^{(l-1)}, \tag{2.13}$$

$$(i^*, j^*) = \arg\max_{\substack{(i,j) \\ i \ne j}} (S_{ij}^{(l)}).$$

5. Unir os grupos mais similares:

Se $|S_{i^*j^*}^{(l)} - S_{i^*j^*}^{(l-1)}| < \varepsilon$

$$\alpha^{(l)} = 1/(c^{(l-1)} - 1) \tag{2.14}$$

Se $S_{i^*j^*}^{(l)} > \alpha^{(l)}$

$$u_{i^*k}^{(l)} = (u_{i^*k}^{(l)} + u_{j^*k}^{(l)}), \quad 1 \le k \le n, \tag{2.15}$$

remover a linha j^* de U,
$c^{(l)} = c^{(l-1)} - 1$

Senão aumentar o número de centros. (2.16)

$$\beta^{(l)} = min(c^{(l-1)}, \beta^{(l)} + 1).$$

Até que $\|U^{(l)} - U^{(l-1)}\| \le \varepsilon$ ou $l \ge lmax$.

Figura 2.4: Algoritmo E-FCM.

O algoritmo de agrupamento participativo ou simplesmente algoritmo participativo, é um algoritmo não supervisionado. O algoritmo de agrupamento participativo, apresentado na Figura 2.5, utiliza os conceitos e notações utilizadas pelo algoritmo FCM, apresentado na seção 2.3.1. Neste algoritmo o grau de pertinência do ponto *i* com respeito ao *k*-ésimo grupo, é calculado através da seguinte equação:

$$u_{ik}^{(l)} = \frac{1}{\sum_{j}^{c} (d_{ik} / d_{jk})^{1/m-1}} . \qquad (2.17)$$

O algoritmo, também utiliza alguns parâmetros, como $\alpha \in [0,1]$, a taxa de aprendizagem, *a* que é o índice de alerta do algoritmo, $\beta \in [0,1]$ determina o grau de conservadorismo na aprendizagem, e τ que é um limiar que define o quanto um ponto deve estar distante dos grupos existentes, para que um novo grupo seja criado. Quando criado, este ponto torna-se o centro do novo grupo.

Na próxima seção serão abordados conceitos básicos sobre árvores de decisão, outra técnica de classificação que é útil em mineração de dados.

2.4 Árvores de decisão

As árvores de decisão surgiram em 1963 de uma análise chamada *Automatic Interaction Detection*, desenvolvido na Universidade de Michigan (Neville, 1999). Porém, estes algoritmos ficaram mais conhecidos quando Ross Quinlan, da Universidade de Sidney, desenvolveu um algoritmo chamado ID3 (Mitchell, 1997).

De acordo com Song e Yoon (2000) e Neville (1999), as árvores de decisão são um conjunto de regras que dividem um conjunto de dados em vários grupos, levando em consideração a relação existente entre as variáveis. Atualmente, é um método muito usado para inferência indutiva.

Dado X, e os parâmetros $\alpha \in [0,1]$, $\beta \in [0,1]$, $\tau \in [0,1]$, $\varepsilon > 0$ e $m > 0$.

1. Inicializar $U^{(0)}$, o número máximo de iterações ($lmax$), c = 2, l=1 e $a_{ki}^{0} = 0$.

2. Repetir

Para $1 \le k \le n$

Para $1 \le i \le c$

Calcular a matriz de covariância e a distância d_{ki}:

$$F_i = \frac{\sum_{j=0}^{n} [u_{ji}^{(l-1)}]^m (x_j - v_i^{(l)})(x_j - v_i^{(l)})^T}{\sum_{j=0}^{n} [u_{ji}^{(l-1)}]^m}, \qquad (2.18)$$

$$d_{ki} = (x_k - v_i^{(l)})^T \left\{ \left[\left(\det(F_i)^{1/n+1} \right) F_i^{-1} \right] \right\} (x_k - v_i^{(l)}) . \qquad (2.19)$$

Calcular $u_{ki}^{(l+1)}$ com a equação (2.17).

Calcular o grau de compatibilidade:

$$\rho_{ki}^{(l)} = 1 - d_{ki} \qquad (2.20)$$

Para $i = 1,..., c$

Calcular o índice de alerta:

$$a_{ki}^{(l)} = a_{ki}^{(l-1)} + \beta\left(\left(1 - \rho_{ki}^{(l)}\right) - a_{ki}^{(l-1)}\right) \qquad (2.21)$$

Se $a_{ki}^{(l)} \le \tau$

Ajustar v_s mais próximo de x_k:

$$v_s = v_s + a\rho_{ks}^{1-a_{ks}^{(l)}} (x_k - v_s) \qquad (2.22)$$

Onde: $s = \arg\max_i \{\rho_{ki}\}$

Senão, criar um novo centro

Atualizar o número de centros

Calcular o índice de compatibilidade entre os centros:

Para $i = 1, \ldots, c-1$

Para $j = i+1, \ldots, c$

$$\rho_{v_i}{}^{l} = 1 - \sum_{h=1}^{p} \left\| v_{ih}{}^{l} - v_{jh}{}^{l} \right\|^2 \qquad (2.23)$$

$$\lambda_{v_i} = \beta(1 - \rho_{v_i}) \qquad (2.24)$$

Se $\lambda_{v_i} \le 0.95\tau$

Eliminar v_i.

Determinar o erro:

$$erro = \left\| V^{(l)} - V^{(l-1)} \right\| = max_{ij} \left\| v_{ih}{}^{(l)} - v_{ih}{}^{(l-1)} \right\| \qquad (2.25)$$

$l = l + 1$

Até (($l \ge lmax$) ou (erro < ε))

5. Recalcular os índices de compatibilidade ρ.

Figura 2.5: Algoritmo de agrupamento participativo.

Segundo Han e Kamber (2001), uma árvore da decisão é um fluxograma com a estrutura de uma árvore, onde o primeiro nó da árvore é a raiz, nó este que contém todos os dados a serem analisados. Cada nó interno da árvore representa um determinado teste que deverá ser feito em um atributo da base de dados selecionada, gerando os nós filhos. Os nós filhos representam o resultado do teste realizado no nó interno. Este procedimento é repetido até que não existam atributos a serem testados. Neste caso, os nós passam a ser chamados de folhas e representam os grupos.

Um modelo típico de árvore de decisão é apresentado na Figura 2.6, onde se classificam potenciais compradores de microcomputadores. Os nós internos dessa árvore são representados por retângulos e têm como função testar os atributos da base de dados utilizada. Os círculos são as folhas das árvores e representam os grupos formados. Neste exemplo, pessoas com atributos cujos valores permitem atingir folhas que contém "Sim" são classificados como prováveis compradores de microcomputadores de acordo co a empresa.

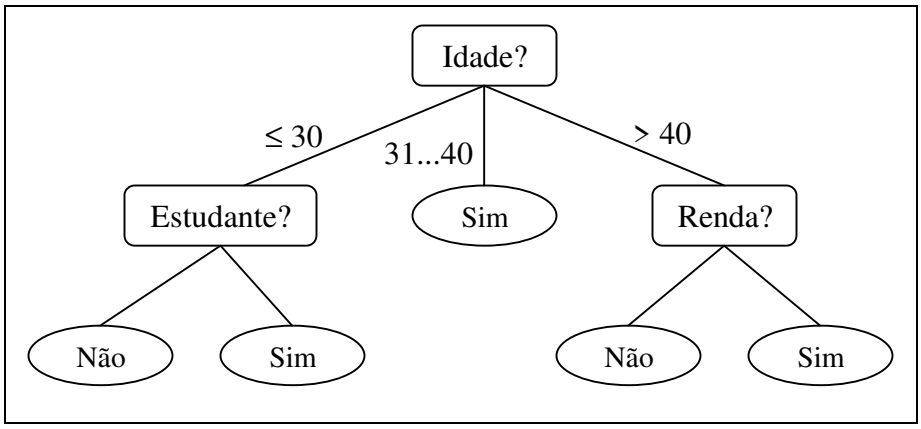


Figura 2.6: Árvore de decisão.

As árvores de decisão podem ser usadas em classificação, predição e regressão, sendo que na mineração de dados elas são especialmente utilizadas para fazer classificação.

Os diversos métodos existentes de árvores de decisão se diferem quanto aos tipos de atributos que são utilizados. Entre os métodos mais representativos estão: *Automatic*

Interaction Detection (Morgan e Sonquist, 1963, Neville, 1999, Cadiz 1994), *Chi Square Automatic Interaction Detection* (Kass, 1975) e o ID3 (Mitchell, 1997, Neville, 1999).

2.4.1 *Automatic Interaction Detection*

O método de *Automatic Interaction Detection* – AID, foi proposto por Morgan e Sonquist (1963). É um método estatístico desenvolvido para ser utilizado em bases de dados de grande porte (Kass, 1975). De acordo com Neville (1999), este método forneceu a primeira abordagem baseada em árvore de decisão para a comunidade estatística, sendo para esta comunidade a forma mais simples de análise através de árvores de decisão.

Supõe-se que as bases de dados usadas pelo AID são compostas por uma variável dependente contínua e por variáveis independentes qualitativas ou categorizadas. A variável somente é considerada uma variável independente quando ela altera o comportamento da variável dependente.

Durante a sua execução, o algoritmo de AID divide um nó da árvore somente em dois outros nós. O conjunto de variáveis é dividido em dois subgrupos, através da maximização da soma dos quadrados entre subconjuntos, em um procedimento em cascata. Este algoritmo inicia com um único grupo de dados. A seguir este grupo é dividido em dois subgrupos, o subgrupo que contém a variável dependente e o subgrupo que contém as variáveis independentes. Então, cada variável independente é testada para fazer a nova divisão desse grupo, examinando-se todas as possibilidades de dividir o grupo em dois. Para cada possível divisão, calcula-se a soma dos quadrados dos valores dos elementos de cada grupo, dividindo-a pela média dos valores dos elementos do grupo da variável dependente. Escolhe-se então a divisão que apresenta a maior soma para representar a contribuição da variável independente. O algoritmo deve continuar desta forma até que não se consiga dividir todos os subgrupos encontrados.

O AID tem duas finalidades básicas quando gera uma árvore de decisão. A primeira é fazer uma análise preliminar de dados; a segunda é formar grupos similares entre si (Cadiz ,1994).

2.4.2 *Chi Square Automatic Interaction Detection*

Após anos de uso, foram encontradas duas deficiências no AID. Primeiro, este método não responde corretamente quando encontra uma variável irregular na base de dados, influenciando o processo de divisão em subgrupos. Quando o AID é utilizado, a possibilidade de encontrar a soma máxima dos quadrados da variável independente é diretamente proporcional ao número de categorias. Segundo, este método também é influenciado quando as variáveis independentes pertencem a mais de uma categoria. Para solucionar essas limitações, Kass (1975) adicionou uma etapa para testar a significância do processo de divisão, permitindo assim a criação de vários subgrupos.

Mais especificamente, Kass (1975), propõe o *Chi Square Automatic Interaction Detection* – CHAID, uma técnica derivada do AID e expandida especialmente para os casos onde as variáveis independentes podem pertencer a mais de uma categoria.

Portanto, ao se proceder de acordo com a técnica CHAID os dados são divididos, a cada passo, em vários grupos e não em somente dois subgrupos, através do teste chi-quadrado, que consiste na maximização da significância estatística do chi-quadrado. As categorias das variáveis independentes são agrupadas somente se elas mostrarem padrões de comportamento semelhantes em relação à variável dependente. Além disto, para cada uma das categorias das variáveis independentes selecionadas, a técnica escolhe a próxima variável que melhor apresenta a categoria da variável anterior. Ao final, os resultados da análise são mostrados em forma de uma árvore, onde as variáveis independentes aparecem de acordo com a capacidade de prever níveis específicos de outras variáveis independentes.

Um problema com esta técnica consiste no teste do chi-quadrado, o qual se mostra inadequado para um estudo exaustivo de uma base de dados, pois é incapaz de extrair determinadas informações. No entanto, é considerado um teste simples para a descoberta de associações mais evidentes.

2.4.3 ID3

Em 1983, na Universidade de Sydney, Austrália, Ross Quinlan desenvolveu o algoritmo chamado ID3 – *Iterative Dichotomizer 3rd*, que serviu como o primeiro programa baseado em árvore de decisão nas áreas de inteligência artificial e aprendizagem de máquina (Quinlan, 1983, Neville, 1999). Sua principal característica é gerar regras ordenadas pela sua importância, gerando um modelo de árvore de decisão dos fatos que afetam os itens de saída.

O ID3 constrói a sua árvore procurando por um atributo que deve ser utilizado como raiz. Para determinar este atributo, todos os atributos deverão ser avaliados via um teste estatístico que determina qual é a melhor classificação. Um descendente da raiz é então criado para cada valor possível do atributo que foi colocado na raiz da árvore. Nós descendentes dão origem a outros, distribuindo os atributos entre os nós até que se finalize a árvore.

A principal decisão do ID3 corresponde a selecionar qual é o melhor atributo para classificar cada nó da árvore. Para selecionar estes atributos em cada nó utiliza-se um teste baseado no "ganho de informação". Este teste tem como objetivo determinar qual nó proporciona o maior ganho de informação na separação dos atributos na classificação. O ganho de informação é definido por (Mitchell, 1997):

$$Ganho(W,E) \equiv Entropia(W) - \sum_{v \in Valores(E)} \frac{|W_v|}{|W|} Entropia(W_v), \qquad (2.26)$$

onde $Entropia(W) \equiv \sum_{h=1}^{b} -q_h \log_2 q_h$. $Valores(E)$ é definido como o conjunto dos possíveis valores do atributos, W é um conjunto finito de dados, W_v é o subconjunto de W para cada valor do atributo E que tem o valor v, q_h é a proporção que W pertence ao grupo h e b é o número de grupos existentes.

2.5 Sistemas baseado em regras nebulosas

Zadeh (1965), introduziu a teoria de conjuntos nebulosos, com o objetivo de modelar a imprecisão presente na linguagem natural. Segundo Zadeh (1965), deve-se considerar a teoria de conjuntos nebulosos não como uma simples teoria, mas como uma metodologia para generalizar uma teoria específica. Desde sua criação, a teoria dos conjuntos nebulosos e a lógica nebulosa correspondente, vêm contribuindo em todas as áreas de pesquisas.

Como citado em Pedrycz e Gomide (1998), um conjunto nebuloso é definido como uma coleção de objetos com valores de pertinência entre 0 (exclusão completa) e 1 (pertinência completa). Estes valores expressam o grau com que o objeto é compatível com as propriedades ou características da referida coleção.

Como Zadeh (1965) definiu, um conjunto nebuloso A é caracterizado por uma função de pertinência que mapeia os elementos de um domínio, espaço, ou universo de discurso U no intervalo unitário [0,1]. Ou seja,

$$A: U \rightarrow [0,1].$$

Um conjunto nebuloso A em U pode ser representado como um conjunto de pares ordenados de elementos genéricos $x \in U$ e os respectivos graus de pertinência:

$$A = \{(A(x)/x) \mid x \in U\}, \tag{2.27}$$

sendo A(x) o grau de pertinência de x em A (Pedrycz e Gomide, 1998). Em princípio, uma função da forma $A:U \rightarrow [0,1]$ descreve uma função de pertinência associada a um conjunto nebuloso A.

As operações padrão com conjuntos nebulosos são: interseção, união e complemento, e compreende generalizações das respectivas operações com conjuntos clássicos. Os operadores padrão de interseção, união e complemento são definidos, respectivamente como:

$$(A \cap B)(x) = \min (A(x), B(x)) = A(x) \wedge B(x), \tag{2.28}$$

$$(A \cup B)(x) = \text{Max } (A(x), B(x)) = A(x) \vee B(x), \tag{2.29}$$

$$A(x) = 1 - A(x), \qquad (2.30)$$

onde, A e B são conjuntos nebulosos definidos em um universo de discurso U e $(A \cap B)(x)$ e $(A \cup B)(x)$ denotam a função de pertinência dos conjuntos resultantes da interseção e da união de A e B, respectivamente.

A teoria de conjuntos nebulosos tem vários sub-ramos, entre eles, a aritmética nebulosa, a programação matemática nebulosa, a teoria de grafos nebulosos e a lógica nebulosa.

Um conceito importante na teoria de conjuntos nebulosos é o de variável lingüística. Variáveis lingüísticas não possuem números como valores, mas termos ou sentenças de uma linguagem natural ou artificial.

Uma variável lingüística é definida por uma quintupla (X,T(X),U,G,M), onde X é o nome da variável, T(X) é um conjunto de nomes dos valores lingüísticos de X, U é o universo de discurso dos conjuntos nebulosos que caracterizam os termos de T(X), G é uma regra sintática, que usualmente tem a forma de uma gramática, para gerar os nomes dos valores lingüísticos e M é uma função que associa um conjunto nebuloso de U a cada elemento de T(X).

2.5.1 Base de conhecimento e inferência nebulosa

Uma base de conhecimento nebulosa consiste em uma base de dados e uma base de regras. Na base de dados ficam armazenadas as características, definições do universo de discurso e as definições das funções de pertinência referentes aos termos lingüísticos. A base de regras é formada por sentenças condicionais que possuem a seguinte estrutura:

$$\text{Se } < \text{antecedente} > \text{ então } < \text{conseqüente} > ,$$

como por exemplo:

$$\text{Se } X \text{ é } A_i \text{ e } Y \text{ é } B_i \text{ então } Z \text{ é } G_i,\ i = 1,2,...,n. \qquad (2.31)$$

onde A_i, B_i e G_i são os conjuntos nebulosos nos universos de discurso X, Y, Z respectivamente; (X é A_i e Y é B_i)**,** é considerado antecedente e Z é G_i é considerado

conseqüente. Basicamente, uma base de conhecimento nebulosa é uma relação entre o antecedente e o conseqüente.

Existem vários modelos para determinar o resultado de uma inferência, a partir dos dados A(x), B(x) e uma base de conhecimento, entre eles o de Mamdani e Takagi-Sugeno.

O modelo de Mamdani utliza os seguintes passos:

Passo 1: $m_{Ai} = \max [A(x) \wedge A_i(x)]$ (2.32)

$m_{Bi} = \max [B(y) \wedge B_i(y)]$ (2.33)

Passo 2: $m_i = \min[m_{Ai}, m_{Bi}], i = 1,2,...,n$ (2.34)

Passo 3: $G'_i(z) = m_i \wedge G_i(z), i = 1,2,...,n$ (2.35)

Passo 4: $G(z) = \bigcup_{i=1}^{n} G'_i(z) = \max[G'_i(z), i =1,2,...,n], \forall z \in Z.$ (2.36)

No modelo Takagi-Sugeno, o conseqüente é uma função, por exemplo na seguinte forma:

$$G_i = g_0^i + g_1^i x_1 + g_2^i y \tag{2.37}$$

onde g_k^i são parâmetros associado ao conseqüente. Neste caso o resultado da inferência será:

$$G = \frac{\sum_{i=1}^{n} m_i G_i}{\sum_{i=1}^{n} m_i} \tag{2.38}$$

Para obter um valor numérico de saída, o valor nebuloso obtido pelo método de Mamdani deve ser transformado através do processo de defuzzificação. Entre os métodos de defuzzificação podemos citar os principais como o centróide e a média dos máximos. No método centróide o valor de saída é o centro de gravidade da função de pertinência resultado da inferência, isto é, G(z).

Em geral, uma base de conhecimento nebulosa necessita ser completa e consistente, garantindo que exista sempre uma regra a ser disparada para qualquer entrada e procurando evitar contradições entre as regras.

2.5.2 Otimização e decisão nebulosa

Bellman e Zadeh (1970) introduziram os princípios de otimização nebulosa, no qual objetivos e restrições são representados por conjuntos nebulosos. Neste caso, uma coleção de N funções objetivas nebulosas F_k, k = 1,....,N, e M restrições C_l, l = 1,...,M, definidas em um universo de discurso U, são dados. Uma decisão nebulosa é definida pela seguinte função de pertinência, onde x ∈ U:

$$D(x) = F_1(x) \wedge ... \wedge F_k(x) \wedge C_1(x) \wedge ... \wedge C_l(x), \quad k = 1,...,N \text{ e } l = 1,..., M. \tag{2.39}$$

Bellman e Zadeh (1970) sugerem a seguinte solução ótima x^*:

$$x^* = \arg(\max_x D(x)) , \tag{2.40}$$

como ilustrado na Figura 2.7.

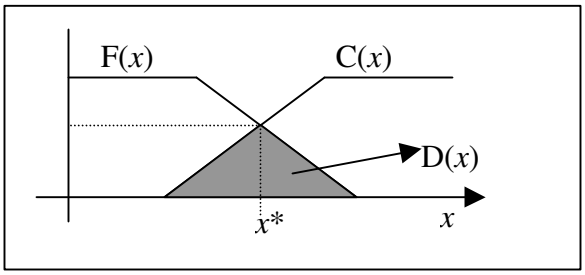


Figura 2.7: Otimização e decisão nebulosa.

2.6 Resumo

Este capítulo abordou algumas propriedades, características e fases do KDD. Também foram apresentados vários algoritmos de agrupamento de dados existentes, tais como, o FCM, o E-FCM e o aprendizado participativo, utilizados na fase de mineração de dados do KDD. Entre os algoritmos de classificação também relevantes nesta fase estão as árvores de decisão, podendo-se citar o AID, CHAID e o ID3, utilizados na classificação, predição e regressão.

Neste capítulo, também foram sumarizados os conceitos de conjuntos nebulosos, base de conhecimento, inferência nebulosa, otimização e decisão nebulosa. Estas metodologias serão utilizadas no modelo de previsão sugerido por Kaymak e Setnes (2001), na seção 3.2 e na aplicação em reposição de jornais apresentada no Capítulo 4.

CAPÍTULO 3

METODOLOGIAS DE PREVISÃO

3.1 Introdução

Este capítulo descreve a metodologia de classificação, seleção e previsão utilizada neste trabalho. Esta metodologia utiliza agrupamento nebuloso e sistemas baseados em regras nebulosas para criar um modelo de previsão. Inclui-se também uma breve recapitulação de conceitos básicos sobre séries temporais e redes neurais estáticas multicamadas como alternativas metodológicas em previsão.

3.2 Modelo nebuloso de previsão

Kaymak e Setnes (2001), propõem uma metodologia estruturada de KDD baseada em técnicas de agrupamento e de decisão no contexto da teoria de conjuntos nebulosos. O propósito principal da metodologia é fornecer modelos de classificação e previsão de clientes alvos em bases de dados de grande porte. Uma aplicação, cujo objetivo é determinar clientes que sejam potencialmente promissores na maximização do desempenho de uma campanha de *marketing* oferecendo um determinado produto financeiro também é apresentada pelos autores.

Na metodologia de Kaymak e Setnes (2001), a estrutura de KDD é análoga àquela abordada na seção 2.2.1 e ilustrada na Figura 2.1, onde primeiramente há a seleção dos dados alvos. Nessa fase, foram selecionados 16.525 clientes de uma base de dados de uma financeira holandesa. Nesse conjunto de dados, cada registro de cliente é composto por 170 atributos, sendo 61 deles com valores binários, 54 contínuos e 55 categorizados,

representando as características de cada um dos clientes. Os valores binários indicam a posse ou não de um produto da financeira, os contínuos são valores monetários associados a contas e posses de produtos na financeira e os categorizados representam características dos clientes, como faixa etária, classificação a partir de parâmetros sócio-econômicos, entre outras.

A seguir vem a fase de preparação do conjunto de dados, visto que em qualquer base de dados usualmente há registros incompletos. Os valores binários de interesse da financeira e da aplicação devem ser completados e não podem se ausentar na base de dados. Se alguns dos registros relativos a estes valores estiverem incompletos, eles devem ser complementados com 0 (zero), para indicar que o cliente não possui o produto correspondente.

Para obter um conjunto de dados completos, freqüentemente, os valores contínuos e categorizados ausentes no conjunto de dados são substituídos por um número que corresponde a um valor desconhecido. Enquanto essa substituição é aceita por algumas aproximações estatísticas, outros métodos, como o agrupamento, requerem um conjunto de dados onde estes estejam dentro de um determinado domínio. Para solucionar este problema, existem duas maneiras: tanto os atributos, quanto os registros que possuem valores incompletos são removidos do conjunto de dados selecionado. Essas duas maneiras são contraditórias, pois é necessário conservar tanto os atributos, como os registros para a exploração. De um lado, alguns preferem conservar a maioria possível de registros no conjunto de dados, de modo que permaneça um número suficiente de exemplos para avaliar grupos interessantes. Por outro lado, outros preferem conservar no conjunto de dados a maioria possível de atributos, com o objetivo de capturar todas as relações relevantes existente no conjunto de dados.

Embora não seja aplicado no caso do modelo de previsão proposto no Capítulo 4, resume-se a seguir a proposta de Kaymak e Setnes (2001) para contornar este dilema. Assume-se um limiar $\eta \in [0,1]$, definido como a proporção entre o número de valores incompletos que cada atributo do conjunto de dados possui pelo número total de valores do atributo. Esse limiar é utilizado para determinar os atributos que devem ser removidos. Ou seja, os atributos que possuem uma proporção de valores incompletos maior que η,

são removidos. Após esta etapa, os registros que ainda contém valores incompletos são simplesmente removidos do conjunto de dados.

O melhor limiar η^* pode ser determinado utilizando o método de decisão nebulosa. Um conjunto nebuloso da maioria de registros que devem ser mantidos, pode ser definido a partir da porcentagem permitida de valores incompletos em cada atributo; uma pequena porcentagem de valores incompletos terá uma pertinência alta no conjunto nebuloso. Adicionalmente, um conjunto nebuloso da maioria de atributos permitidos, pode ser definido a partir da porcentagem permitida de valores incompletos em cada atributo. Uma alta porcentagem permitida de valores terá, tipicamente, uma alta pertinência nesse conjunto. Uma vez definidos esses conjuntos nebulosos, o melhor limiar é encontrado utilizando o método de decisão nebulosa proposto por Bellman e Zadeh (1970) (seção 2.5.2), onde a decisão corresponde à aquela que fornece o maior grau de pertinência que satisfaça as duas condições. Os conjuntos nebulosos para a maioria de registros mantidos e maioria de atributos permitidos são derivados dos dados, considerando o número de valores incompletos em cada atributo e a diminuição resultante no número dos atributos e registros. Portanto, o limiar η^* é então determinado como o ponto de máxima pertinência da interseção entre os dois conjuntos nebulosos. Determinada a porcentagem permitida de informação incompleta nos atributos, todos os atributos com uma porcentagem alta de valores incompletos são removidos. Após a remoção dos atributos, registros que ainda possuem valores incompletos devem ser removidos do conjunto de dados, como mostra o Exemplo 3.1, ilustrado na Figura 3.1.

Exemplo 3.1: Considere uma base de dados inicial representada por uma matriz 6×4, como ilustrada na Figura 3.1. Removendo todos os registros dessa matriz, ou seja, os clientes, C_k, k = 1,2,...,6, que possuem um ou mais atributos incompletos, a matriz ficará somente com um registro. Por outro lado se remover todos os atributos, F_j, j = 1,2,3,4, que contém um ou mais valores incompletos, um único atributo permanecerá nesse conjunto de dados. Então, para manter a maioria de registros e atributos na base de dados, deve-se calcular a proporção do número de atributos incompletos pelo número total de atributos para cada registro, obtendo a função de pertinência da maioria de registros mantidos. Também se calcula a proporção do número de registros incompletos

pelo número de registros completos para cada um dos atributos, obtendo a função de pertinência da maioria de atributos permitidos. Esses conjuntos nebulosos são ilustrados no gráfico da Figura 3.1. A melhor decisão é obtida no ponto máximo de interseção dessas duas funções nebulosas, onde determina que qualquer atributo que possuir uma proporção maior entre o número de registros incompletos e o número total de registros, deve ser eliminado. Após eliminar os atributos, qualquer registro que ainda possuir algum atributo incompleto deve ser eliminado. Neste exemplo, serão mantidos na base de dados dois atributos e quatro registros.

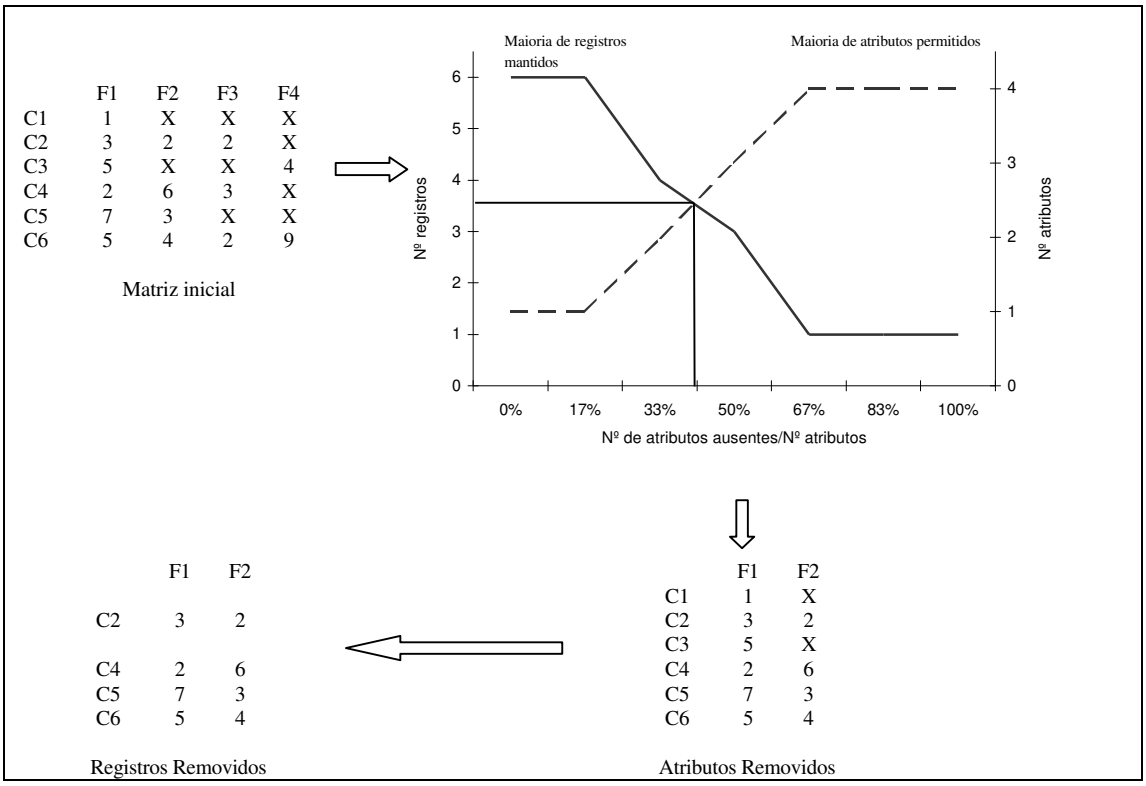


Figura 3.1: Exemplo 3.1.

No conjunto de dados selecionado por Kaymak e Setnes (2001) para a exploração, quando se reduzem os atributos e registros incompletos através do método de decisão nebulosa remove-se 22 atributos e 1039 registros.

Após preparação adequada, o conjunto de dados está pronto para ser explorado, iniciando a segunda fase, chamada mineração de dados. A exploração é feita através de

técnicas de agrupamento nebuloso e, por essa razão, os dados devem estar dentro de um domínio apropriado. Para isso os valores binários e categorizados necessitam de ser eliminados do conjunto de dados a ser explorado. Valores binários indicam a posse de produtos por clientes e na base em questão já estão implicitamente representados no conjunto de dados pelos valores contínuos. Isto pode ser interpretado como o resultado de uma projeção do conjunto de dados no espaço dos valores contínuos. Os valores categorizados na base utilizada já são representados pelos produtos que cada cliente possui, sendo assim desnecessário usá-los. Após a redução do conjunto de dados, pela eliminação dos valores binários e categorizados, restam 49 atributos. Este número ainda é considerado grande, sendo necessário novas reduções. Para determinar quais atributos são importantes para um determinado problema, tendo em vista reduções adicionais de atributos, constrói-se gráficos ganho, um gráfico para cada atributo, com o objetivo de analisar o impacto que o atributo causa no desempenho do modelo.

A construção e análise de gráficos de ganho baseia-se em técnicas de agrupamento, conforme detalhado a seguir.

A classificação dos clientes no caso da aplicação de Kaymak e Setnes (2001) é reduzida através das características contidas no conjunto de dados $\{x_k, y_k\}$, k = 1,2,...,N, onde $x_k = \lfloor x_{k1}, x_{k2}, ..., x_{kp} \rfloor$, é um vetor p-dimensional de atributos que descreve o k-ésimo registro e $y_k \in \{0,1\}$ é a resposta correspondente a este registro. Os valores dos p atributos são estudados separadamente para verificar o quão bons eles são na discriminação de dois grupos, um que responde e um que não responde ao atributo. Os valores x_{kj} de cada atributo X_j, j=1,2,...,p são divididos em c_j grupos através de agrupamento nebuloso de dados e o desempenho nebuloso – RD – de cada grupo é calculado de acordo com:

$$RD_{ij} = \frac{\sum_{k=1}^{N} \mu_{ij}(x_{kj}) y_k}{\sum_{k=1}^{N} \mu_{ij}(x_{kj})}, \quad 1 \le i \le c_j, \quad 1 \le j \le p, \tag{3.1}$$

onde $\mu_{ij}(x_{kj})$ denota o grau de pertinência de x_{kj} *ao grupo* i e utiliza os conceitos e notações utilizadas no algoritmo FCM, na seção 2.3.1.

O desempenho de cada grupo é utilizado para calcular a pontuação – SC_{jk} – de cada registro de acordo com o atributo *j*, conforme a expressão:

$$SC_{jk} = \sum_{i=1}^{c_j} \mu_{ij}(x_{kj}) RD_{ij} \ , \ \ 1 \le k \le N \tag{3.2}$$

Quanto maior a pontuação SC_{jk}, maior a chance de uma resposta positiva do *k*-ésimo registro, de acordo com o atributo *j*. A expressão 3.2, fornece, então, um modelo para ordenar os registros (exemplo: clientes) de acordo com o atributo *j*.

O modelo criado para cada atributo pode ser avaliado utilizando-se um gráfico de ganho. Para se construir um gráfico de ganho para o *j*-ésimo atributo os registros devem primeiramente ser organizados em ordem decrescente dos valores de suas respectivas pontuações (SC_{jk}). O gráfico de ganho é então obtido supondo um conjunto alvo composto por 1 até *N* registros. Nestes conjuntos, os registros são inseridos seguindo a ordem crescente k^*. Calcula-se também a soma acumulada das correspondentes respostas y_{k^*}. O valor do gráfico de ganho é dado por:

$$g_j\left(\frac{k^*}{N}\right) = \frac{\sum_{i=1}^{k^*} y_i}{\sum_{i=1}^{N} y_i}, \quad k^* = 1,2,...,N \tag{3.3}$$

Ao gráfico de ganho correspondente ao *j*-ésimo atributo se associa uma pontuação SX_j que reflete a eficiência deste atributo na discriminação. O valor de SX_j é dado por:

$$SX_j = \sum_{k^*=1}^{N} \left(1 - \frac{k^*}{N}\right) g_j\left(\frac{k^*}{N}\right) \tag{3.4}$$

Utilizando o gráfico de ganho, a modelagem prossegue de uma maneira iterativa que inclui a seleção de atributos para l = 1,2,...,D iterações, onde D é a profundidade da árvore de decisão resultante, ou seja, o número de atributos que serão usados pelo modelo. Os gráficos de ganho produzidos independentemente para cada atributo são comparados de acordo com as suas respectivas pontuações SX_j. A cada iteração l do

processo de modelagem, o atributo com a maior pontuação é selecionado. Isto é, seleciona-se o atributo $j^l = \arg\max_j SX_j$. Os c_{j^l}'s centros do grupo, do atributo selecionado e os valores correspondentes do desempenho RD_{ij^l}, são parte do modelo final. O conjunto de registros, é então dividido em n' grupos $K^{j'}$, $j' = 1,...,n'$, tal que:

$$K^{j'} = \left\{ k \middle| SC_{j'k} = \max_j SC_{jk} \right\}. \tag{3.5}$$

Cada grupo $K^{j'}$ contém os registros que são melhor caracterizados ou classificados pelo atributo j'. Após determinar $K^{j'}$, determina-se os $K^{(l)}$ registros que são melhor caracterizados pelo atributo $j^{(l)}$, isto é, pelo atributo com o gráfico de ganho mais favorável. A seguir, removem-se os $K^{(l)}$ clientes do conjunto de dados analisado. Isto é, removem-se

$$K^{(l)} = \left| K^{j^l} \right| \tag{3.6}$$

clientes do conjunto de dados.

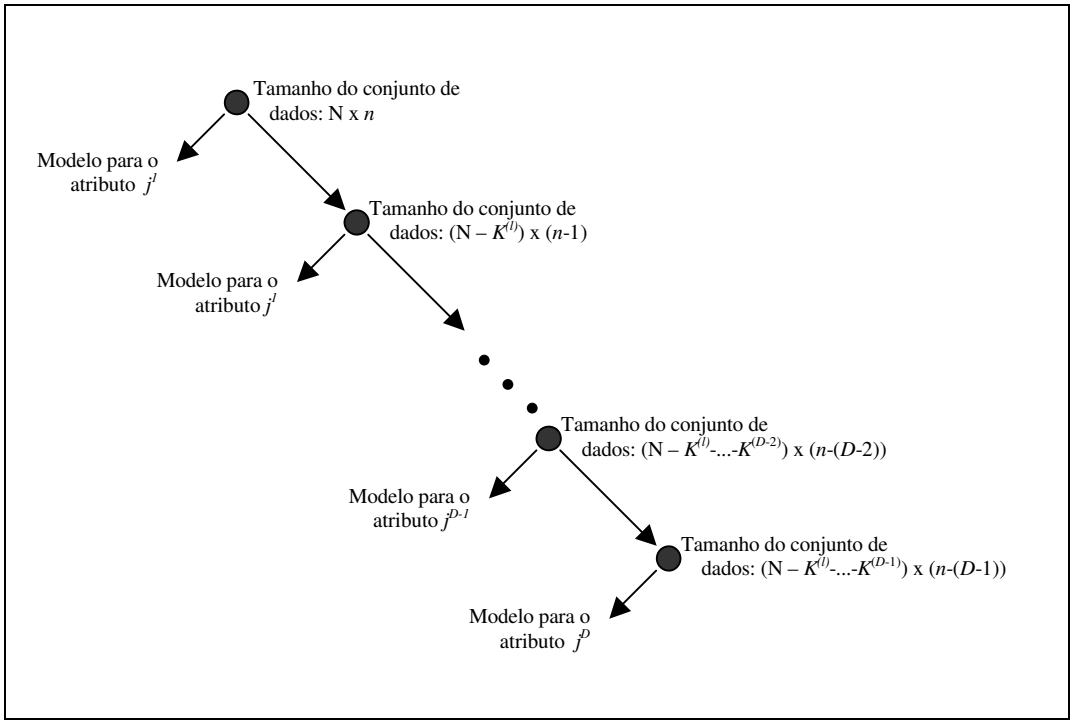


Figura 3.2: Redução do conjunto de dados.

Portanto a dimensão do conjunto de dados é reduzida pela remoção do atributo j^l. Esse processo repete-se na próxima iteração (l + 1), mas considerando-se o conjunto de dados reduzido, conforme ilustrado na Figura 3.2.

Através da redução iterativa do conjunto de dados, busca-se pela estrutura mais importante que pode ser descrita por um único atributo. No passo l, os valores de um atributo podem ser divididos em poucos, relativamente distintos grupos, algum deles com um valor de RD grande. Quando os registros que possuem uma maior pontuação no modelo baseado neste atributo forem removidos, outras estruturas no conjunto de dados podem ser reveladas pelo agrupamento dos valores dos atributos restantes. Isto torna possível capturar determinadas estruturas no conjunto de dados onde normalmente se deveria considerar iterações entre os atributos, conforme ilustra o Exemplo 3.2. O número de iterações é igual à profundidade da árvore de decisão resultante. A profundidade pode ser especificada pelo analista, indicando o número de atributos desejáveis no modelo. No entanto, não é necessário pré-determinar o valor da profundidade D. O processo de modelagem pode ser finalizado quando, por exemplo, não houver mais dados com resposta positiva no conjunto de dados reduzido.

Exemplo 3.2: A Figura 3.3 ilustra um problema com os atributos X_1 e X_2, de acordo com os pontos mostrados no espaço dos atributos. Os dados representados pelos pontos pretos representam os dados alvos, ou seja os dados que possuem aquele determinado produto. No passo 1a os valores correspondentes aos atributos são agrupados independentemente, obtendo-se dois grupos para X_1: A_1 e B_1 com os desempenhos pontuados em 2/11 e 5/10, respectivamente. Três grupos são obtidos para X_2: A_2, B_2 e C_2, com os desempenhos pontuados em 3/9, 1/3 e 3/9, respectivamente. No passo 1b, a avaliação dos gráficos de ganho dos dois modelos resultantes $f(X_1)$ e $f(X_2)$, baseados em X_1 e X_2, respectivamente, revela que X_1 é a melhor variável. O modelo $f(X_1)$ é armazenado, assim como os respectivos centros de grupo e o valor de desempenho nebuloso (RD) correspondente. O processo de modelagem é repetido para o conjunto de dados reduzido, passo 2. No conjunto de dados reduzido, tanto o atributo X_1 quanto os dados que obtém maior pontuação no modelo $f(X_1)$ do que no $f(X_2)$ devem ser removidos.

Isto é, os pontos do grupo marcado com i pontuam 1/2 no modelo $f(X_1)$ e 1/3 no modelo $f(X_2)$. Observe que, por estarmos considerando grupos rígidos (não nebulosos) neste exemplo, por simplicidade, a pontuação corresponde ao valor de RD. Conseqüentemente, assim como no caso anterior, os grupos marcados com ii e iii pontuam 2/11 e 1/3 nos modelos $f(X_1)$ e $f(X_2)$, respectivamente. Logo os pontos do grupo i são removidos e os dados dos grupos ii e iii constituem o conjunto de dados reduzido, para o qual dois grupos são identificados no passo 2.

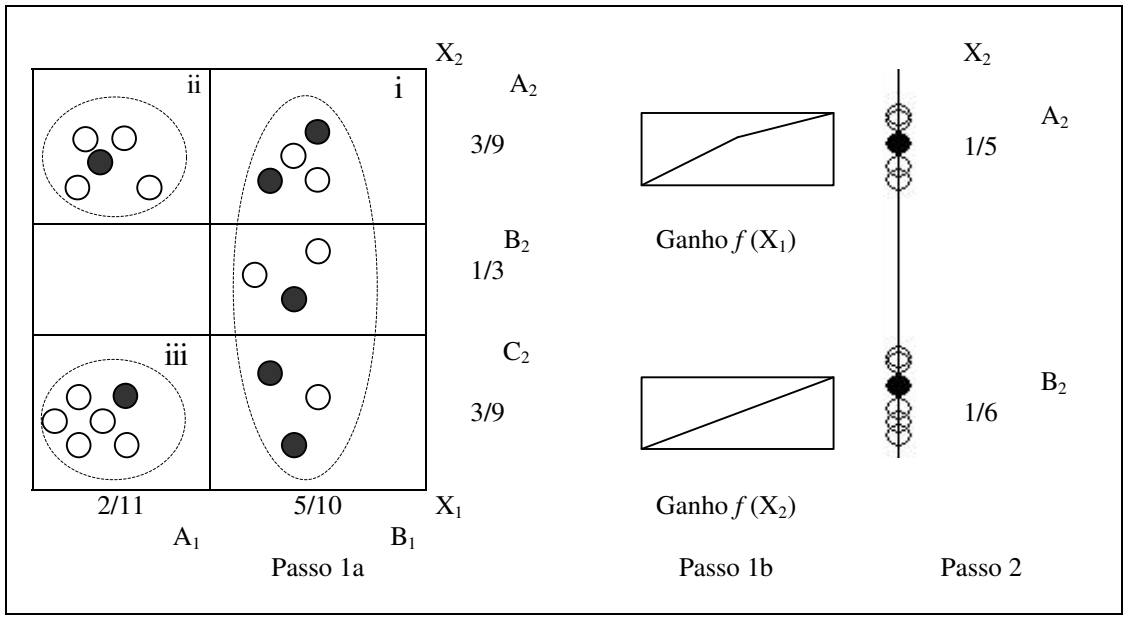


Figura 3.3: Exemplo 3.2 (Kaymak and Setnes, 2001).

Na Figura 3.4, observa-se os gráficos de ganho de três atributos do modelo referente à financeira holandesa considerada no exemplo de aplicação de Kaymak e Setnes (2001). Os gráficos de ganho estão ordenados de acordo com a respectiva pontuação SX_j de cada atributo j. O processo iterativo de modelagem com a redução do conjunto de dados é ilustrado na Figura 3.4.

A seleção do modelo final é feita após a criação dos modelos para os D atributos mais significativos. Os modelos individuais são combinados em um modelo final. A pontuação e a seleção dos clientes são obtidas calculando a média dos resultados obtidos por cada atributo separadamente. A pontuação de cada registro x_k é dada por:

$$SC_k = \frac{1}{D}\sum_{j^l=1}^{D} SC_{kj^l}\ ,\ \ k = 1,2,...,N. \tag{3.7}$$

Neste ponto, todos os clientes do conjunto de dados estão devidamente pontuados de acordo com o correspondente valor de SC_k. Os registros com maior pontuação serão selecionados como alvos.

A decisão quanto ao valor do limiar de corte (valor abaixo do qual os registros são considerados como tendo baixa pontuação) depende do analista e dos objetivos da aplicação.

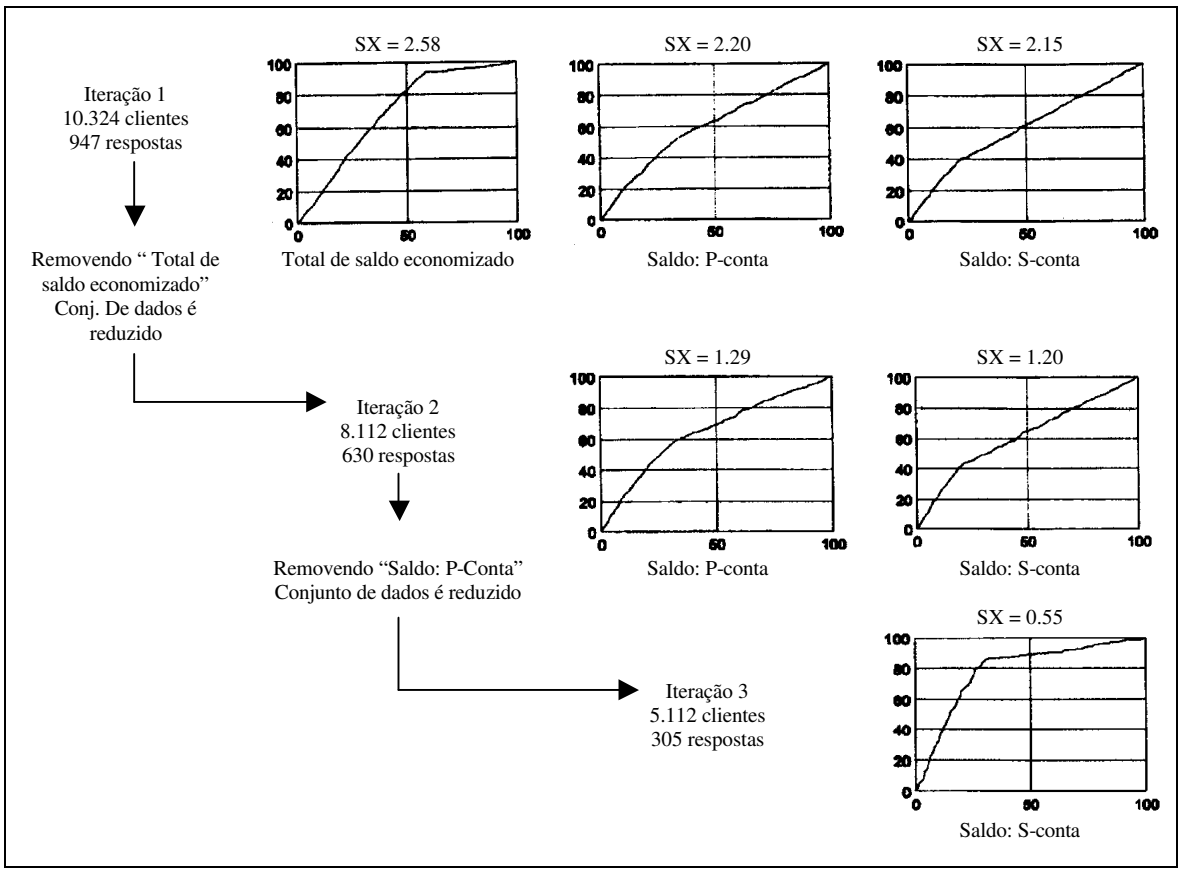


Figura 3.4: Redução da base de dados durante o processo iterativo de modelagem.

Para permitir maior transparência e maior facilidade na inspeção e validação do modelo nebuloso, uma versão na forma de um sistema baseado em regras pode ser extraída a partir dos parâmetros do modelo obtido pelo processo de modelagem iterativo

descrito acima. O modelo, na forma de um sistema baseado em regras, também é conveniente quando é necessário complementá-lo com conhecimento especialista, além de tornar as regras acessíveis para outros propósitos (e.g., transparência). Após o agrupamento de dados, uma base de conhecimento nebulosa pode ser extraída do modelo, fornecendo regras que podem ser utilizadas para a tomada de decisão. Isto é, para cada atributo X_j, $j = 1,...,D$, utilizado no modelo, os respectivos centros c_j compõem o antecedente, e o desempenho (RD) associado a esse grupo determina o conseqüente da regra. Os c_j grupos podem ser representados por conjuntos nebulosos, A_{ij}, com $i = 1,...,c_j$. Uma maneira de determinar os A_{ij}'s é associar cada grupo com um conjunto nebuloso trapezoidal, onde o centro do grupo corresponde ao núcleo do conjunto nebuloso. A base de regras nebulosas contém um total de $\sum_{j=1}^{D} c_j$ regras. As regras do modelo têm a seguinte estrutura:

$$R_i: \text{Se atributo } X_j \text{ é } A_{ij} \text{ então resposta Y é } D_i, \tag{3.8}$$

onde D_i representa o desempenho – RD – correspondente ao grupo com centro *i*.

Utilizando o modelo de Takagi-Sugeno, as regras criadas pelo exemplo de aplicação de Kaymak e Setnes (2001) estão ilustradas na Figura 3.5. Os antecedentes das regras estão ilustrados nas Figuras 3.6, 3.7 e 3.8, respectivamente. Utilizando essa base de regras nebulosa criada e as características que cada um dos clientes possuem, de acordo com a sua respectiva pontuação, é possível, então, determinar os clientes que são considerados alvos para a aquisição de um determinado produto.

R_1	Se total de saldo economizado é baixo	então resposta y = 0.0463
R_2	Se total de saldo economizado é moderado	então resposta y = 0.1559
R_3	Se total de saldo economizado é alto	então resposta y = 0.1631
R_4	Se total de saldo economizado é muito alto	então resposta y = 0.1542
R_5	Se saldo: p-conta é baixo	então resposta y = 0.0697
R_6	Se saldo: p-conta é moderado	então resposta y = 0.1338
R_7	Se saldo: p-conta é alto	então resposta y = 0.1839
R_8	Se saldo: s-conta é baixo	então resposta y = 0.0733
R_9	Se saldo: s-conta é moderado	então resposta y = 0.1608
R_{10}	Se saldo: s-conta é alto	então resposta y = 0.1803

Figura 3.5: Regras extraídas do modelo.

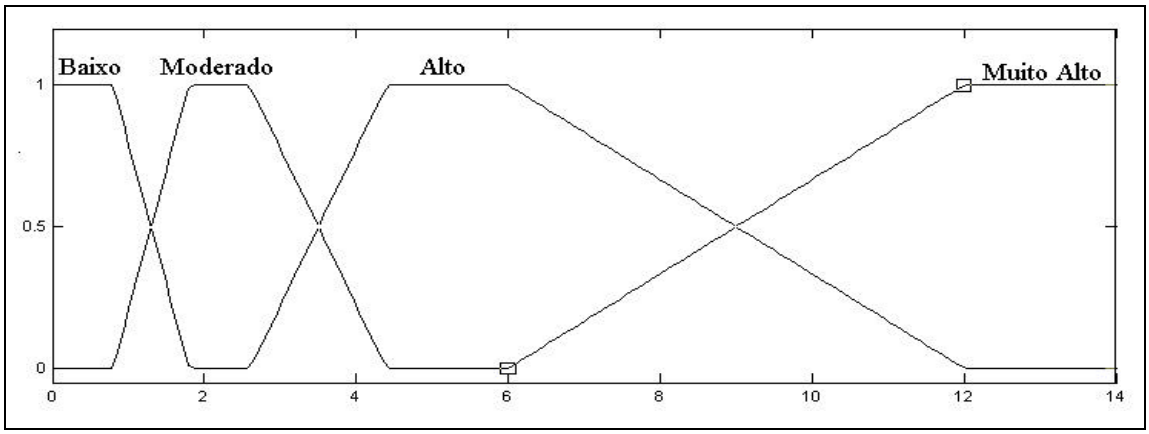


Figura 3.6: Total de saldo economizado.

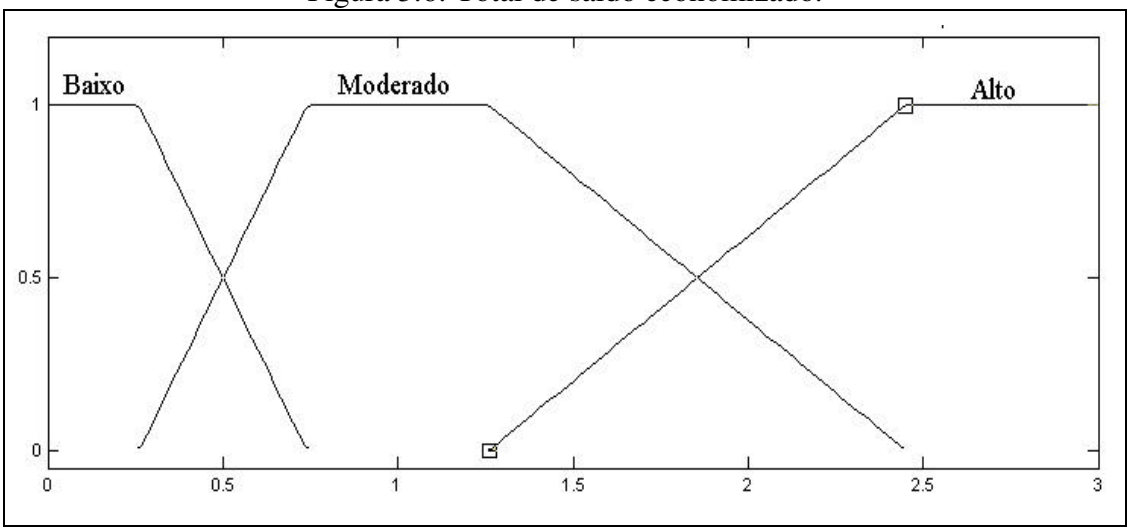


Figura 3.7: Saldo da p-conta.

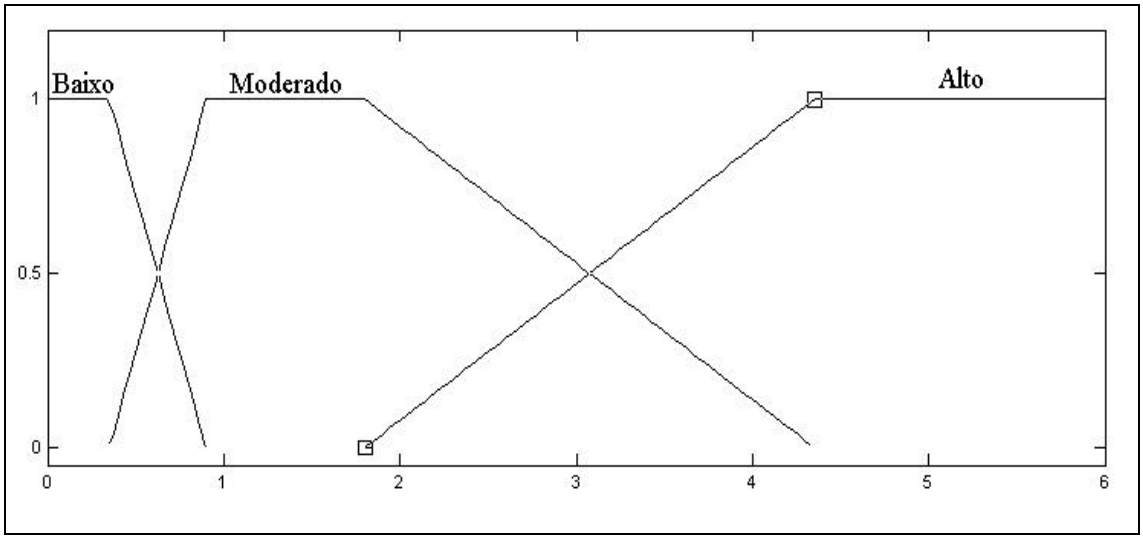


Figura 3.8: Saldo da s-conta.

Kaymak e Setnes (2001), compararam o desempenho do modelo proposto com o modelo correspondente obtido pelo CHAID. O método CHAID na prática mostra problemas associados com a aplicação de técnicas estatísticas, especialmente quando alguns dados são inválidos, impondo uma limitação no seu desempenho. O agrupamento nebuloso é adaptativo. Ao contrário da aproximação estatística onde um cliente pertence a um grupo com um determinado número de pontos, no método proposto a cada cliente corresponde uma contagem particular, ajudando assim a capturar características individuais. O número das regras e atributos selecionados pelo modelo CHAID a partir dos dados é maior do que o modelo nebuloso, mas o modelo nebuloso fornece melhores resultados.

3.3 Séries temporais

Uma série temporal pode ser definida como qualquer conjunto de observações ordenadas no tempo (Morettin e Toloi, 1987; Abelém, 1994). As características de fenômenos físicos, biológicos, econômicos, entre outros da natureza, podem ser estudados através da análise de séries temporais. As séries temporais têm a previsão como uma de suas principais áreas de aplicação.

O objetivo da análise de séries temporais é obter propriedades estatísticas a fim de caracterizar seu comportamento e identificar um modelo adequado para uma determinada aplicação (Ballini, 2000). Essa análise possui dois caminhos: a análise no domínio do tempo e a análise no domínio da freqüência. A análise no domínio do tempo concentra-se em descrever a magnitude de eventos que ocorrem em determinados instantes e na relação entre as observações em diferentes instantes de tempo (Ballini, 2000). A análise no domínio da freqüência, analisa a freqüência de certos eventos que ocorrem em determinado período de tempo (Ballini, 2000). As duas formas de análise de séries temporais se complementam, pois cada uma captura os diferentes aspectos existentes em uma série.

A análise clássica de séries temporais é feita através da decomposição da série em quatro componentes: tendência, sazonal, cíclica e aleatória (Ballini, 2000). Os componentes de tendência são, freqüentemente, aqueles que produzem mudanças graduais em longo prazo. Os componentes sazonais de uma série são oscilações de subida e de queda que sempre ocorrem em um determinado período do ano, do mês, da semana, ou do dia. Os componentes cíclicos são aqueles que provocam oscilações de subida e de queda nas séries, de forma suave e repetitiva, ao longo do componente tendência. A diferença essencial entre os componentes sazonais e cíclicos é que o primeiro possui movimentos previsíveis, ocorrendo em intervalos regulares de tempo, enquanto que os movimentos cíclicos tendem a ser irregulares. O quarto componente, aleatório, representa movimentos ascendentes e descendentes da série após a ocorrência de um efeito de tendência, um efeito cíclico, ou de um efeito sazonal, sendo que este componente aparece como flutuações de período curto, com deslocamento inexplicável, e geralmente, correspondem a eventos imprevisíveis.

Segundo Mueller (1996), o objetivo dessa decomposição consiste em remover cada um dos componentes, permitindo que o comportamento da série temporal seja melhor compreendido e, conseqüentemente, contribua para prever valores futuros mais apropriados.

Uma série temporal pode ser representada por $X = \{x_1, x_2, ..., x_t\}$, onde cada observação x_t está associada a um instante de tempo distinto, existindo uma relação de dependência temporal entre essas observações.

Os métodos de previsão de séries temporais podem ser classificados como simples e avançados. Segundo Morettin e Toloi (1981), os métodos simples consistem em identificar o padrão básico presente nos dados históricos e, então, usar esse padrão para prever valores futuros. Os métodos avançados são definidos por Mueller (1996) como aqueles que fornecem uma previsão de valores futuros pela combinação de valores reais passados e/ou dos erros ocorridos.

Entre os métodos avançados de previsão de séries temporais existe o modelo Autoregressivo (AR) que é dado por:

$$x_t = \phi_1 x_{t-1} + \phi_2 \ x_{t-2} + \ldots + \phi_p \ x_{t-q} + e_t \ , \qquad (3.9)$$

onde x_t corresponde à observação da série temporal no tempo t, ϕ_p corresponde ao parâmetro do modelo AR de ordem q.

O parâmetro e_t representa o erro devido a eventos aleatórios que não podem ser explicados pelo modelo. O erro, e_t, também chamado de ruído branco, possui média zero e variância constante para cada valor de t. Caso as observações da série temporal possam ser representadas por (3.9), a ordem do modelo a ser determinada e os parâmetros estimados, é possível prever os valores futuros da série em análise.

A maioria dos métodos de previsão de séries temporais se baseia na suposição de que observações passadas contêm todas as informações sobre o padrão de comportamento da série temporal e esse padrão é recorrente no tempo. Mueller (1996) diz que o propósito dos métodos de previsão consiste em distinguir o padrão de qualquer ruído que possa estar contido nas observações e então usar esse padrão para prever os valores futuros da série temporal. Assim, pela identificação desse ruído, a previsão para períodos de tempo subseqüentes ao observado pode ser desenvolvida.

3.4 Redes Neurais Artificiais

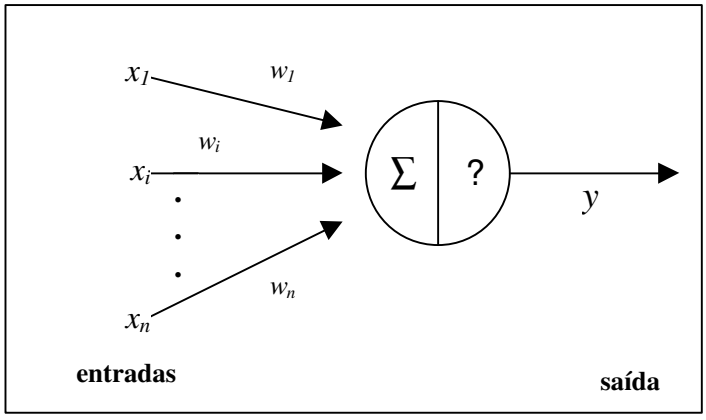


Figura 3.9: Neurônio artificial.

As redes neurais artificiais são utilizadas não só na construção de modelos de previsão, mas também de classificação e controle. Essas redes são compostas por elementos computacionais simples. Um desses elementos, chamado de neurônio, possui n entradas, $x_1, x_2, ..., x_n$, e uma saída y, sendo que cada uma das entradas x_i possui um peso w_i associado, que determina o quanto a i-ésima entrada contribui para a saída y, como ilustra a Figura 3.9.

O neurônio artificial calcula sua saída através da soma ponderada de suas entradas e por uma função de ativação não linear $f(x)$.

As redes neurais são construídas conectando a saída de um neurônio a entrada de um ou mais neurônios. As conexões de entrada são assinaladas a uma camada de neurônios, chamada camada de entrada, e as saídas finais são atribuídas à outra camada, denominada camada de saída. Essas duas camadas se conectam através de uma ou mais camadas intermediárias. A Figura 3.10 ilustra a arquitetura de uma rede neural multicamada, com uma única camada intermediária.

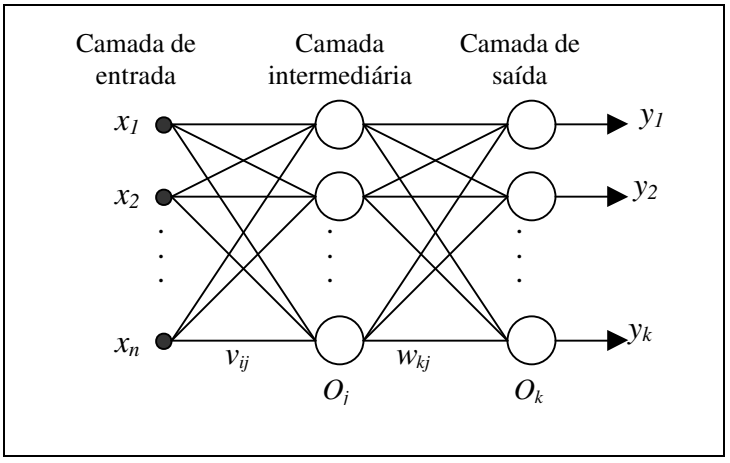


Figura 3.10: Rede neural multicamada.

Os principais passos para a criação de uma rede neural são:

- identificar as entradas e saídas;
- escolher um modelo de rede;

- escolher uma topologia apropriada, definindo seus níveis intermediários;
- treinar a rede com um conjunto de dados representativos;
- testar a rede em um conjunto independente do conjunto de treinamento;
- retreinar a rede se necessário;
- aplicar o modelo gerado ao problema real e avaliar seus resultados.

Essencial para uma rede neural é o seu treinamento, onde é feito o ajuste dos pesos das conexões até que a rede produza saídas correspondentes aos padrões de entradas. Em previsão, a idéia é usar um conjunto de treinamento para ajustar os pesos das conexões até que a rede tenha um comportamento preditivo correto.

Um algoritmo comum para treinar redes neurais é o *backpropagation*, resumido na próxima seção.

3.4.1 Algoritmo de *Backpropagation*

O algoritmo de *backpropagation* é um método de treinamento de redes neurais, baseado no conceito de gradiente e tem como objetivo minimizar o erro quadrático médio.

Aplicações que utilizam este treinamento geralmente tratam de problemas que envolvem o mapeamento de um conjunto de entradas a um conjunto específico de saídas desejadas. Sendo assim, este método é um exemplo de treinamento supervisionado. O principal objetivo é treinar a rede para que ela seja capaz de responder corretamente às entradas que são utilizadas para o treinamento. Após o treinamento espera-se que a rede seja capaz de fornecer resultados semelhantes para entradas semelhantes.

O treinamento da rede neural de *backpropagation* envolve três fases:

- alimentação da entrada da rede com dados de treinamento;
- cálculo e *backpropagation* do erro;
- ajuste de pesos.

Após o término do treinamento da rede, somente a fase de alimentação da rede é executada, obtendo assim, a saída.

Na execução do algoritmo de *backpropagation*, os pesos entre as camadas de entrada, intermediária e de saída, são inicializados aleatoriamente, com valores pequenos.

Após a inicialização dos pesos, na fase de alimentação da rede, a entrada x é enviada para as unidades da camada intermediária. Em seguida, cada unidade intermediária calcula sua ativação z_in_j, de acordo com (3.10):

$$z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \text{ , } i,j = 1,2,...,n. \tag{3.10}$$

e determina a correspondente saída. O sinal de saída, enviado para a camada seguinte, é calculado via função de ativação (3.11):

$$z_j = f(z\_in_j) . \tag{3.11}$$

A função de ativação é em geral, uma função contínua, diferenciável e monotonicamente crescente. Estas são as características necessárias para que a rede seja treinada com o algoritmo de *backpropagation*. Os sinais de saída gerados pela função de ativação são sempre limitados, em geral entre (-1, 1) ou (0,1).

Cada unidade da camada de saída O_k calcula sua ativação y_in_k para proporcionar uma resposta à uma determinada entrada. As unidades da camada de saída também calculam as suas saídas através de uma função de ativação. Portanto, a ativação y_in_k, é calculada conforme (3.12):

$$y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk} \text{ , } k = 1,2,...,n. \tag{3.12}$$

O sinal, enviado para a camada de saída, é calculado através da função de ativação:

$$y_k = f(y\_in_k) \tag{3.13}$$

Durante o treinamento, cada unidade de saída compara sua saída com o valor desejado t_k para determinar o erro associado àquele padrão. Baseado no erro, um fator δ_k é calculado e utilizado para distribuir o erro às unidades da camada intermediária. O termo δ_k é calculado conforme:

$$\delta_k = (t_k - y_k) f'(y\_in_k) \tag{3.14}$$

Este fator é também utilizado para fazer o ajuste de pesos entre a camada de saída e a camada intermediária; $f'(y\_in_k)$ é a derivada da função de ativação.

Para fazer o ajuste dos pesos, são calculados os termos de correção conforme:

$$\Delta w_{jk} = \alpha \delta_k z_j \tag{3.15}$$
$$\Delta w_{0k} = \alpha \delta_k \tag{3.16}$$

Ainda no treinamento, é calculado o fator δ_j para a camada intermediária. Este termo é calculado da seguinte forma:

$$\delta_j = \delta\_in_j f'(z\_in_j), \tag{3.17}$$

onde o valor de δ_in_j é dados por:

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk} \tag{3.18}$$

O fator δ_j é utilizado para fazer o ajuste de pesos entre a camada intermediária e a camada de entrada. Para fazer o ajuste de pesos, são calculados os termos de correção:

$$\Delta v_{ij} = \alpha \delta_j x_i \tag{3.19}$$
$$\Delta v_{0j} = \alpha \delta_j \tag{3.20}$$

onde α é a taxa de aprendizagem.

O ajuste dos pesos w_{jk}, entre a camada intermediária e a camada de saída, é baseado no fator δ_k e na ativação z_j da camada intermediária. O ajuste dos pesos v_{ij}, entre a camada de entrada e a camada intermediária, é baseado no fator δ_j e a entrada x_i.

Os novos pesos são calculados da seguinte forma:

$$w_{jk}(novo) = w_{jk}(velho) + \Delta w_{jk} \quad (3.21)$$

$$v_{ij}(novo) = v_{ij}(velho) + \Delta v_{ij} \quad (3.22)$$

Segundo Freeman e Skapura (1992), os padrões de treinamento são apresentados sucessivamente às unidades da rede neural até que um erro aceitável seja alcançado ou enquanto um número máximo de iterações não for satisfeito. O último conjunto de pesos observado entre as conexões é mantido para testar a habilidade da rede em mapear a função de entrada saída e a conseqüente validação do modelo neural.

3.5 Resumo

Neste capítulo foi apresentado o método de Kaymak e Setnes (2001), como uma técnica de KDD, que consiste nas seguintes fases:

1. Preparação dos dados: selecionar dados de uma determinada base de dados, completar registros incompletos e eliminar registros e atributos incompletos utilizando o método de aproximação nebulosa;
2. Mineração de dados: agrupar os dados com um algoritmo de agrupamento nebuloso, calcular o desempenho de cada atributo, a pontuação de cada um dos registros, eliminar atributos redundantes e extrair uma base de conhecimento nebulosa para a tomada de decisão.

Esse modelo será utilizado no desenvolvimento de uma aplicação de previsão de reposição de jornais entre postos de vendas, que será apresentado no próximo capítulo.

Modelos de séries temporais e redes neurais, úteis na realização de previsões, serão utilizadas na seção 4.4 para a comparação entre os resultados por eles obtidos e os resultados da metodologia desenvolvida no capítulo 4.

CAPÍTULO 4

METODOLOGIA DE PREVISÃO DE REPOSIÇÃO

4.1 Introdução

Este capítulo trata do problema de reposição de jornais (Cardoso e Gomide, 2003a, 2003b), apresentando uma metodologia para determinar a quantidade de jornais a serem distribuídos em uma região, baseada no método de Kaymak e Setnes (2001). O objetivo é obter um sistema de apoio à tomada de decisão na previsão da reposição a ser feita nas bancas de jornal. Na seqüência, apresenta-se os resultados obtidos com o sistema e comparações realizadas com métodos alternativos baseados em redes neurais e séries temporais.

4.2 Definição do problema

Atualmente, a imprensa é uma poderosa indústria, considerada uma das maiores dos meios de comunicação em massa, sendo constituída de publicações periódicas que divulgam notícias, informações, comentários e imagens referentes aos acontecimentos econômicos, esportivos, sociais, entre outros, do mundo, os quais são de interesse dos indivíduos e das comunidades.

Segundo Fleischfresser (2001), o jornalismo é uma atividade complexa, que abrange desde a simples coleta de notícia até a sua difusão organizada, através de empresas editoras, cuja força e prestígio se baseiam na circulação, representada pelo número de exemplares vendidos e pelo volume de anúncios.

A confecção de um jornal pode ser dividida, de acordo com Fleischfresser (2001), em cinco etapas:

a. Redação: Trabalho dos profissionais que colhem e/ou redigem notícias, escrevem reportagens e editoriais, corrigem ou reescrevem, ilustram e diagramam as matérias, revêem os originais compostos.

b. Fotolitagem: Processo de geração de filme com as reportagens e imagens, e da preparação das chapas para a impressão.

c. Impressão: É a etapa em que as chapas para impressão são encaixadas e são definidas as cores; as rotativas imprimem cortam e dobram os rolos de papel, deixando pronto o jornal.

d. Expedição: É o processo de agrupamento dos diversos cadernos do jornal, da colocação dos encartes e da embalagem.

e. Distribuição: É a etapa de entrega dos jornais aos assinantes e às bancas.

Esse trabalho objetiva desenvolver um sistema para auxiliar a fase de impressão, expedição e distribuição de jornais às bancas. Para estas fases, ele deverá fornecer a quantidade total de jornal a ser impresso, a fim de que a expedição e distribuição tenham a quantidade certa de jornais que deverá ser embalada e distribuída para cada banca, ou seja a quantidade que provavelmente será comercializada naquele dia nas bancas, levando em consideração a possibilidade de se minimizar as perdas e maximizar as vendas.

Para a elaboração deste trabalho, a base de dados utilizada foi fornecida por uma empresa jornalística. Esta base de dados é composta por 650 (seiscentos e cinqüenta) bancas, onde cada uma possui a data, a reposição feita e a quantidade de jornais que foi vendida em cada dia, no período de janeiro de 1998 a agosto de 2000, como ilustra a Tabela 4-1.

Os dados contidos nesta base de dados não possuem um comportamento padrão explícito, sendo difícil prever, a partir de sua inspeção, o valor que deve ser reposto em cada banca a cada dia. Exemplos de dados e as correspondentes médias e médias móveis, são mostrados, para uma das bancas, nas Figuras 4.1, 4.2, 4.3 e 4.4. Estas figuras ilustram o comportamento das vendas para a banca considerada. Esses dados foram utilizados para

fazer a previsão da melhor reposição para as bancas, segundo a metodologia apresentada no capítulo anterior.

Tabela 4-1: Base de dados utilizada no problema

	Atributos			
	Banca	**Data**	**Reposição**	**Venda**
Registros	20	01/01/1998	19	6
	20	02/01/1998	15	14
	20	03/01/1998	17	10
	...	...	...	...
	20	30/08/2000	20	11
	23	01/01/1998	7	7
	23	02/01/1998	5	3
	23	03/01/1998	8	7
	...	...	...	...
	23	30/08/2000	5	5
	207	01/01/1998	0	0
	207	02/01/1998	8	8
	207	03/01/1998	12	9
	...	...	...	...
	207	30/08/2000	2	1
	...	...	...	...

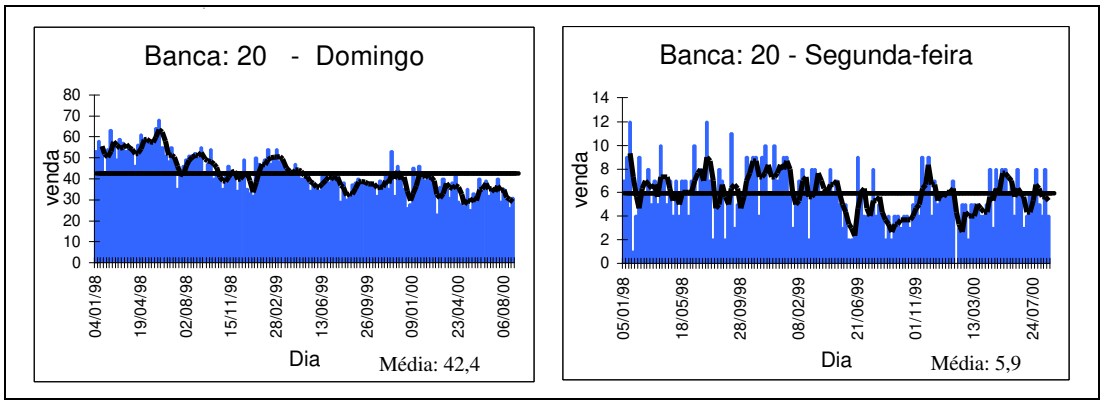


Figura 4.1: Dados de uma das bancas do banco de dados.

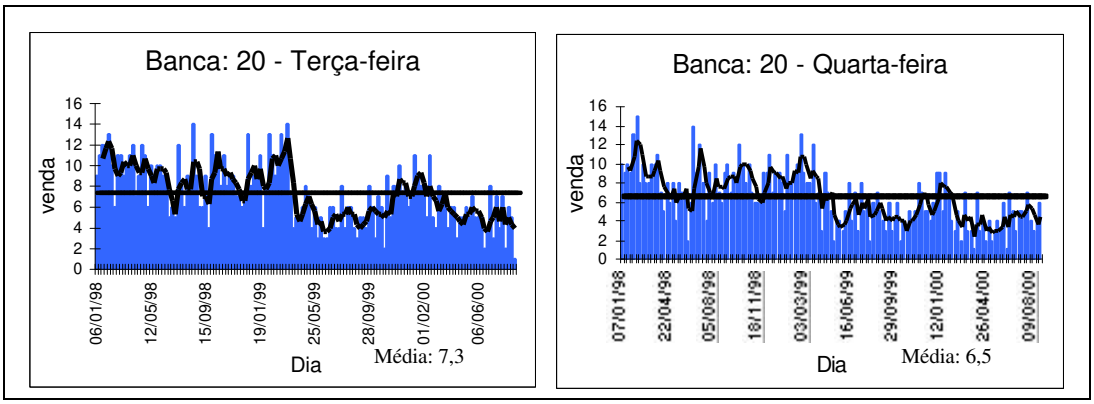


Figura 4.2: Dados de uma das bancas do banco de dados.

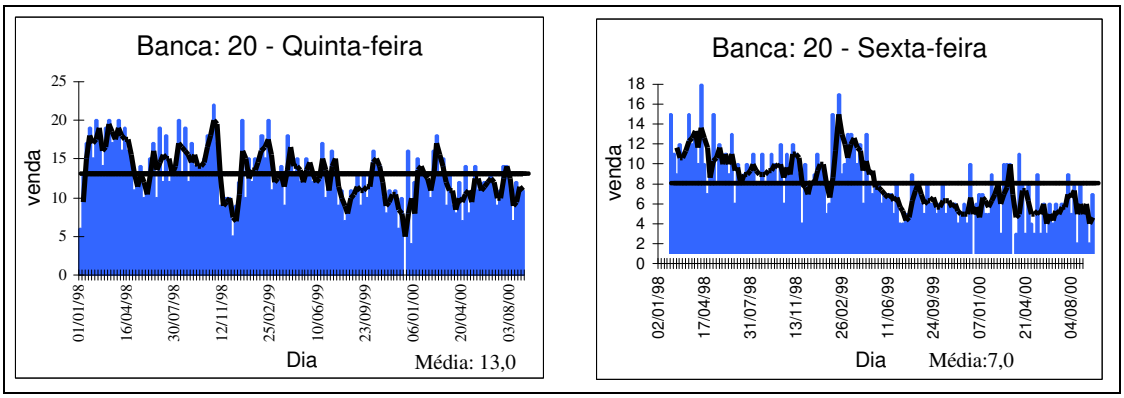


Figura 4.3: Dados de uma das bancas do banco de dados.

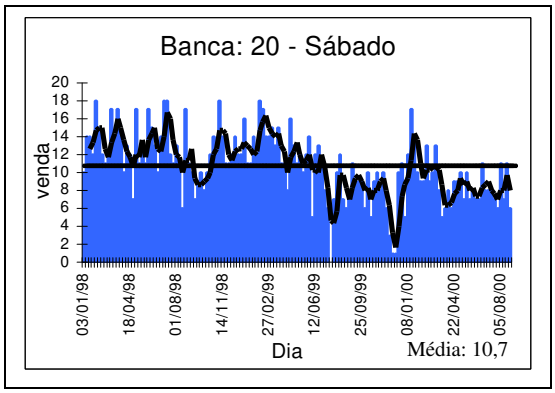


Figura 4.4: Dados de uma das bancas do banco de dados.

O método de Kaymak e Setnes (2001), será aplicado com o objetivo de propor um método que determine melhor a quantidade de jornais a ser reposto em cada banca de jornal diariamente.

4.3 Previsão baseada em agrupamento e regras nebulosos

Conforme a seção anterior, esse trabalho propõe uma aplicação de agrupamento de dados e regras nebulosas para o cálculo do desempenho de cada uma das bancas do banco de dados, com a finalidade de determinar a reposição para essas bancas, segundo o método de Kaymak e Setnes (2001) (Cardoso e Gomide, 2003a, 2003b). Para tal, foi desenvolvido um sistema de KDD análogo ao apresentado na seção 2.2.1.

Como vimos, o KDD inicia-se com a fase de preparação dos dados, pois usualmente em base de dados existem registros e atributos com valores errados e incompletos. Na base de dados utilizada, registros com valores inadequados de *reposição* e *venda*, foram completados com 0 (zero), pois estes registros quando não estão completos, significa que naquele determinado dia não houve *reposição*, *venda* ou ambos. Bancas que não possuíam a maioria dos valores dos atributos, foram excluídas. O conjunto de atributos escolhido para esta aplicação foi {*reposição*, *venda*} de cada uma das 600 bancas que restaram no conjunto de dados para o período de tempo armazenado no banco de dados. Não sendo necessário eliminar outros registros e atributos, o conjunto de dados, é considerado adequadamente preparado para a segunda fase.

A segunda fase, mineração de dados, inicia-se com o agrupamento nebuloso, aplicado a cada uma das bancas separadamente. Nessa fase, primeiramente foi utilizado o algoritmo E-FCM a fim de se classificar cada uma das bancas de acordo com a *reposição* e a *venda*. O algoritmo foi aplicado com o objetivo de determinar o número de grupos existente em cada banca. Como vimos, a principal característica do E-FCM é determinar o número adequado de grupos, conforme detalhado na seção 2.3.2. O algoritmo E-FCM foi implementado de acordo com a Figura 2.4. Nesta aplicação, X contém os valores da *reposição* e *venda* de uma banca; c, valor inicializado aleatoriamente de acordo com o número de pontos existentes em X; $m = 2$; $\varepsilon = 0{,}001$ e $lmax = 30$.

Além do algoritmo E-FCM, foi também implementado o algoritmo de agrupamento participativo apresentado na seção 2.3.3 e ilustrado na Figura 2.5. Neste caso, os parâmetros utilizados foram os mesmos para o algoritmo E-FCM, exceto: α = 0,01; β = 0,9 e τ = 0,05.

O algoritmo participativo foi implementado e mostrou-se ser computacionalmente mais eficiente no tempo de processamento em relação ao número de pontos em *X* que o algoritmo E-FCM. A menos do desempenho computacional, tanto o algoritmo participativo quanto o E-FCM sugeriram dois grupos. Admitindo-se a priori que o número de grupos é conhecido, o que é raro na prática, pode-se utilizar algoritmos computacionalmente mais eficientes, como por exemplo o FCM (seção 2.3.1). O apêndice A, mostra a comparação da complexidade desses algoritmos em termos do tempo de processamento em relação ao número de pontos em *X* e o número de grupos existentes. No caso do problema de reposição, supondo a presença de dois grupos, o FCM proporcionou resultados de agrupamento semelhante aos anteriores, mas de forma mais rápida, conforme esperado.

A Figura 4.5 mostra exemplos de resultados proporcionados pelo algoritmo FCM, utilizando os parâmetros utilizados pelo E-FCM, exceto c = 2. Estes resultados serão utilizados posteriormente para obter a base de regras de previsão. Nos resultados apresentados na Figura 4.5, exceto na Banca 107, os dados se dispõem de modo que a *reposição* seja sempre maior ou igual à *venda*, obtendo uma tendência linear para o limite superior entre os pontos de *reposição* versus *venda*.

Para confirmar, o resultado do agrupamento utilizou-se o algoritmo AID (seção 2.4.1), o qual classifica os dados através de árvore de decisão e por ser o algoritmo que se obteve acesso, via a ferramenta estatística Systat 10.2, desenvolvida pela Systat Software Inc. (disponível na Internet no endereço: http:\www.systat.com) e também pela característica do AID de ser eficaz quando se possuem poucas variáveis no conjunto de dados. Comparando os resultados dos algoritmos de agrupamento nebulosos, o AID também obteve dois grupos. Por possuir dois atributos no conjunto de dados, o AID obteve o mesmo agrupamento. Caso contrário, provavelmente o AID encontraria um número maior de grupos. Os agrupamentos obtidos pelo AID para algumas bancas do conjunto de dados são ilustrados na Figura 4.6, onde a *venda*, considerada a variável

dependente, pode ser dividida em dois sub-grupos de acordo com a condição imposta pela variável independente, neste caso, a *reposição*.

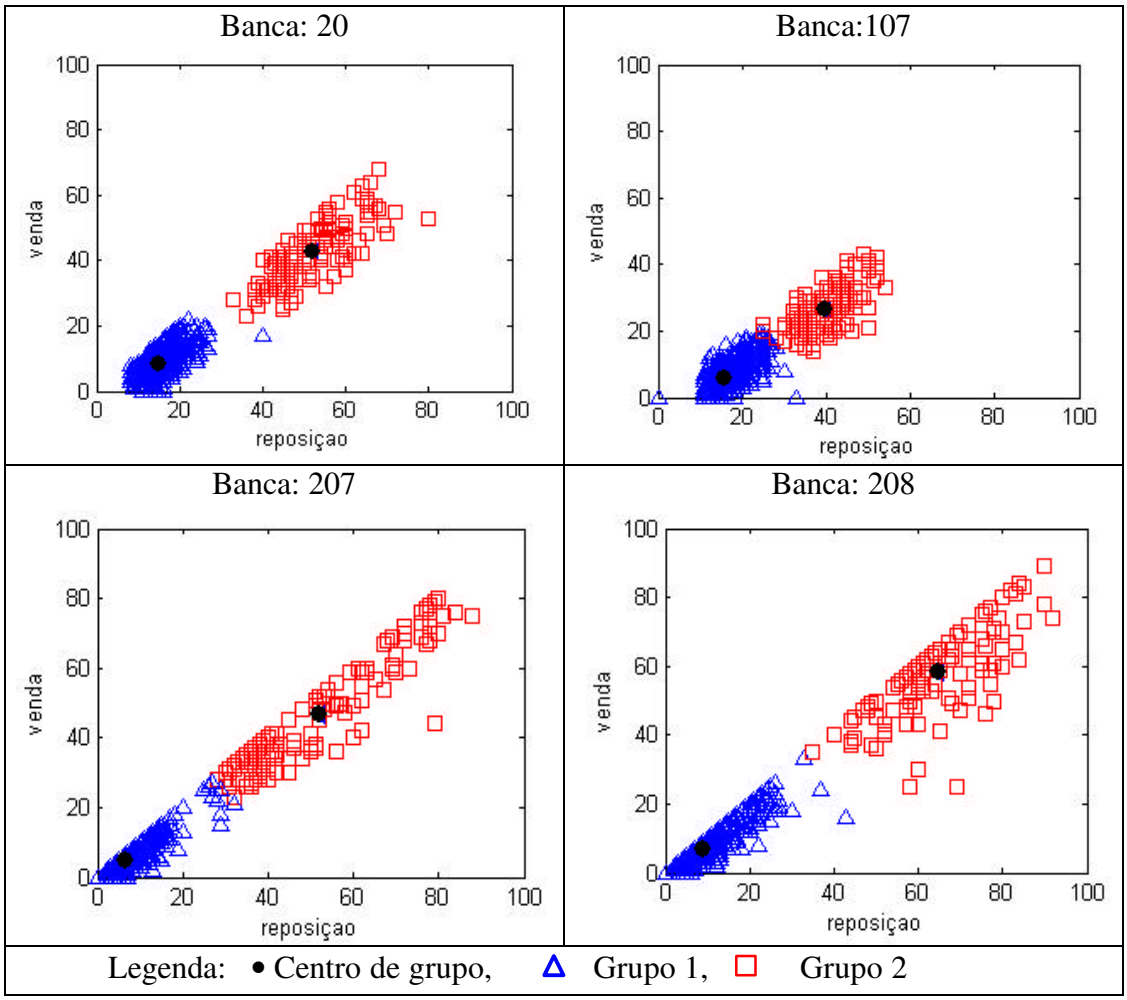


Figura 4.5: Agrupamento de dados obtidos pelo algoritmo FCM.

Como o conjunto de dados utilizado nesta aplicação possui somente dois atributos, *venda* e *reposição*, o AID fornece um resultado próximo dos resultados obtidos pelos algoritmos de agrupamento, exceto pela classificação dos registros que contém algum atributo incompleto que este algoritmo sempre classifica como sendo pertencente ao conjunto sempre mais a esquerda, ou seja, no primeiro nó da árvore à esquerda. Como o AID não introduziu características diferentes dos outros métodos de exploração, ele não será abordado no restante do trabalho.

A análise dos pontos que compõem cada grupo, como por exemplo, os da Figura 4.5 indicam que o grupo 1(Δ) representa os dias de segunda-feira a sábado enquanto que o grupo 2 (□) representa domingos ou feriados.

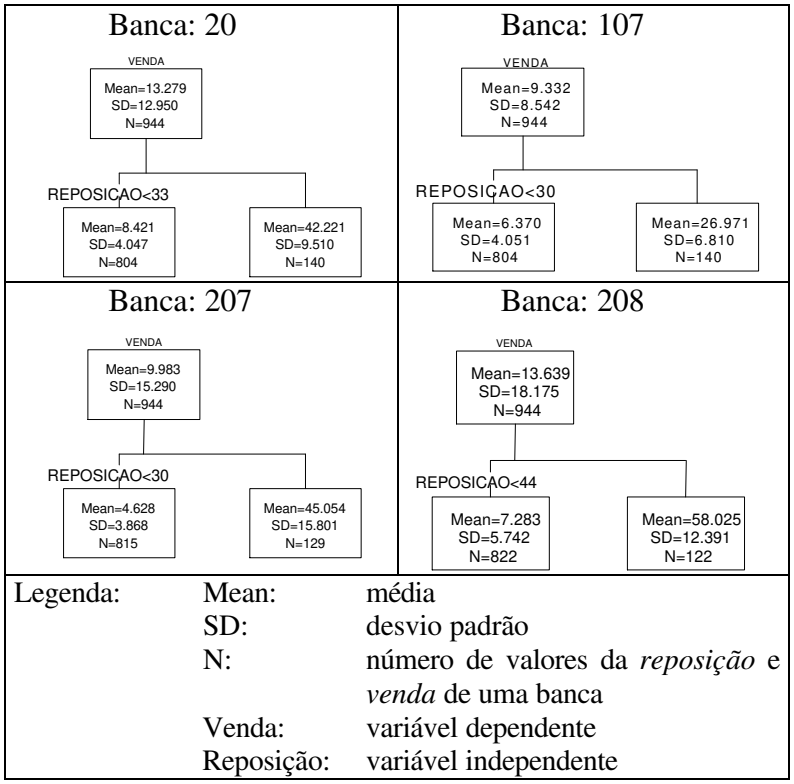


Figura 4.6: Agrupamento dos dados através do AID.

Uma vez encontrado os grupos pelo algoritmo FCM, calcula-se o desempenho – RD – para cada um dos grupos encontrados de acordo com a expressão (4.1).

$$RD_{ij}^{l} = \frac{\sum_{l=1}^{L}\sum_{k=1}^{N}\mu_{ij}(x_{lk}^{j})\,y_{lk}}{\sum_{k=1}^{N}\mu_{ij}(x_{lk}^{j})},\ 1 \le i \le c_j, 1 \le j \le J, \qquad (4.1)$$

onde x_{lk}^{j} é o *l*-ésimo atributo do *k*-ésimo dia, $l = 1,\ldots, L$, $k = 1,\ldots,N$, e $\mu_{ij}(x_{lk}^{j})$ é o grau de pertinência de x_{lk}^{j} no *i*-ésimo grupo da banca *j*,

$$y_{lk} = \begin{cases} 1, & se\ x_{lk}^{j} \in G_i^j = \{x_{lk}^{j} \mid \mu_{ij}(x_{lk}^{j}) \ge \mu_{ij}(x_{rk}^{j}),\ r = 1,\ldots, j, r \ne l\} \\ 0, & caso\ contrário \end{cases} \qquad (4.2)$$

que representa a resposta correspondente ao grupo i, sendo c_j o número de grupos correspondente à banca j. No caso desta aplicação, assumimos $L = 2$ (*reposição* e *venda*), $N = 944$ (número de dias) e $J = 600$ (número de bancas). A Figura 4.7 ilustra o desempenho de cada um dos grupos de acordo com os grupos sugeridos pelo algoritmo FCM.

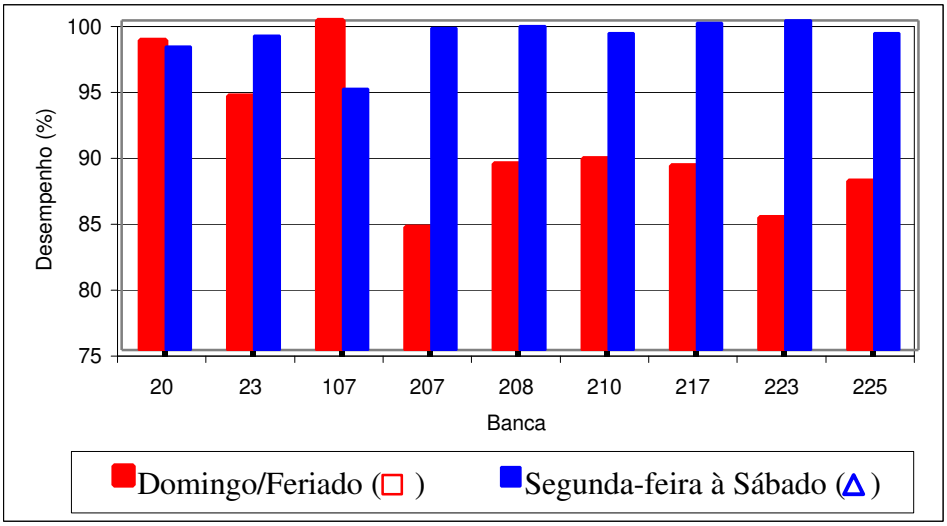


Figura 4.7: Desempenho dos grupos de cada banca.

O valor do desempenho RD de cada um dos grupos é utilizado, então, para calcular a pontuação SC,

$$SC_{jk} = \sum_{i=1}^{c_j} \mu_{ij}(x_{lk}^{j}) RD_{ij}\,, \tag{4.3}$$

para cada dia k e cada uma das bancas j. A pontuação computa o comportamento que cada dia obteve em relação ao desempenho do grupo, no qual o dia possui a maior pertinência. As Figuras 4.8 a 4.11 ilustram o gráfico de ganho, de acordo com a pontuação – SC – em ordem crescente dos valores para cada banca, relativos aos grupos da Figura 4.5. O primeiro gráfico de cada figura corresponde ao grupo 2 (□) e o segundo ao grupo 1 (Δ). Como o conjunto de dados utilizado possui poucos atributos, não é necessário calcular *SX* para cada atributo, pois os atributos que estão neste conjunto de dados de exploração não podem ser eliminados.

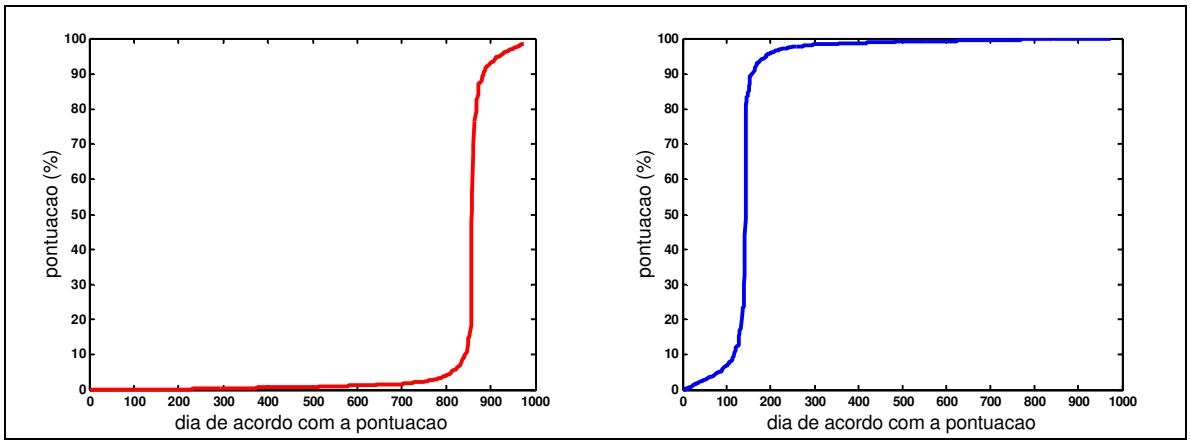


Figura 4.8: Gráfico de ganho da banca 20.

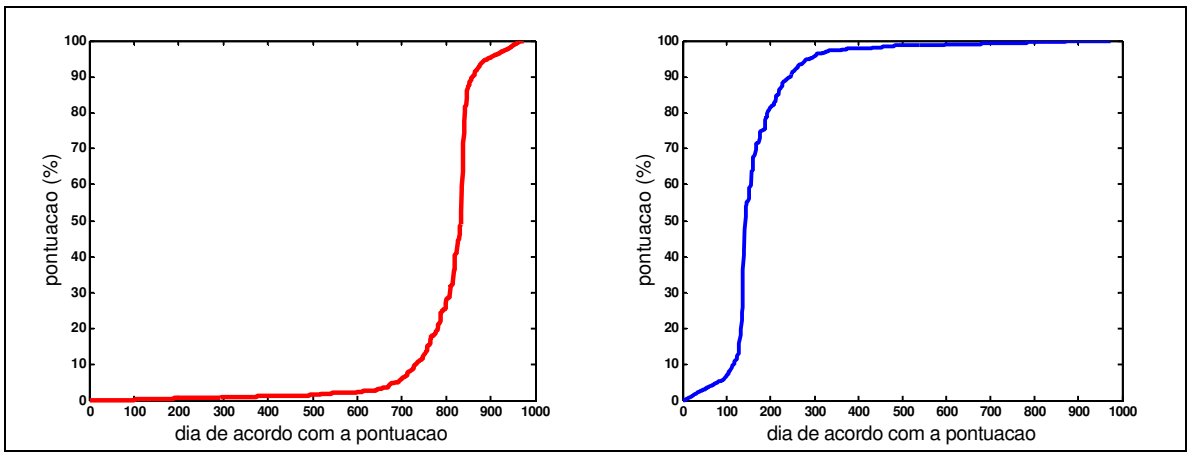


Figura 4.9: Gráfico de ganho da banca 107.

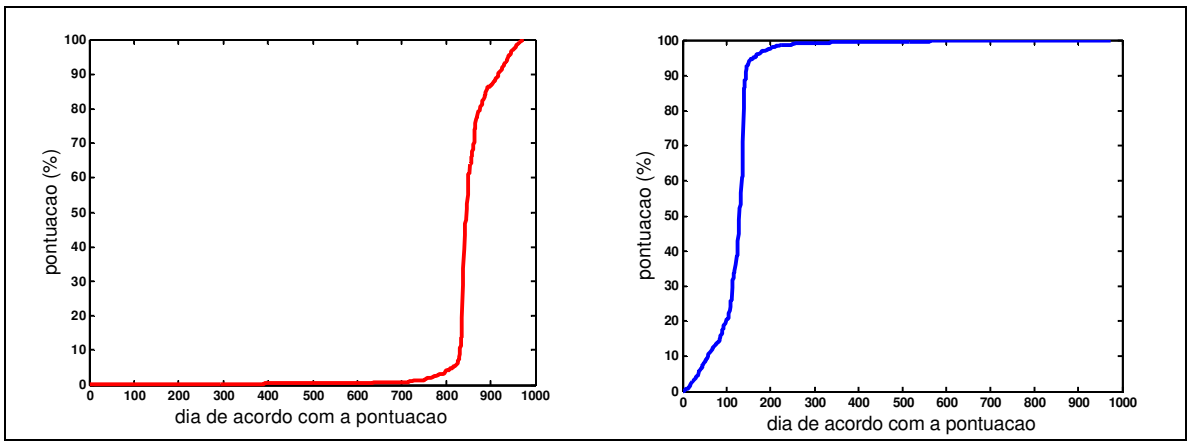


Figura 4.10: Gráfico de ganho da banca 207.

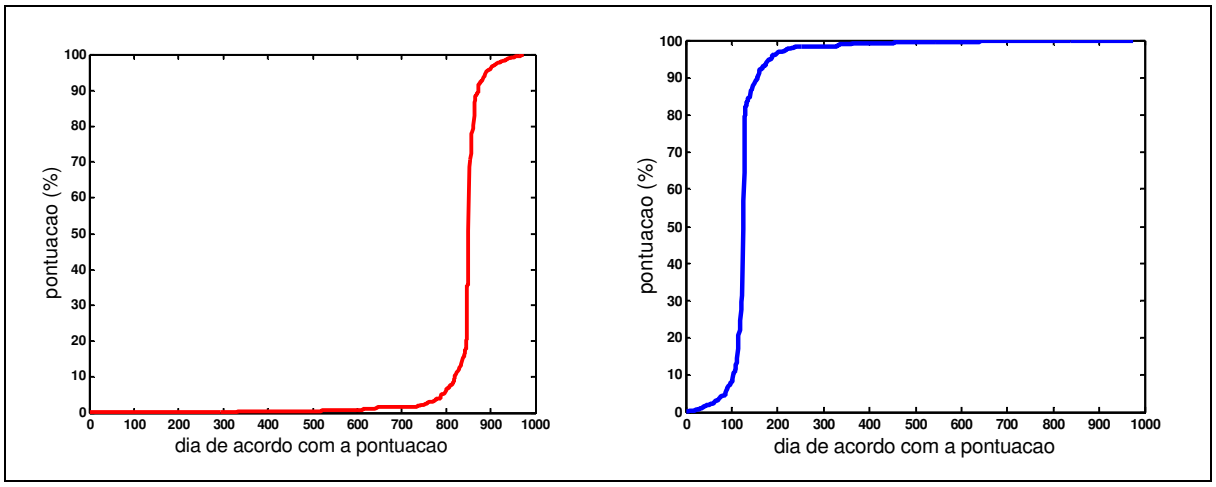


Figura 4.11: Gráfico de ganho da banca 208.

Uma vez calculados os valores de RD e SC, o próximo passo é determinar uma base de conhecimento e inferência nebulosa de acordo com o agrupamento obtido. Neste ponto, diferentemente de Kaymak e Setnes (2001), utiliza-se aqui o modelo de Mamdani (seção 2.5.1) na construção de uma base de conhecimento e inferência nebulosa criada e processada no Tool Box Fuzzy do Matlab 6.1, da MathWorks, Inc.. O modelo de Mamdani é utilizado pelo fato de considerarmos o conhecimento de um especialista na construção da base de conhecimento e inferência nebulosa. A base de conhecimento e inferência nebulosa considera três características: a *notícia*, o *dia* e a *pontuação*, respectivamente, como ilustra a Figura 4.12.

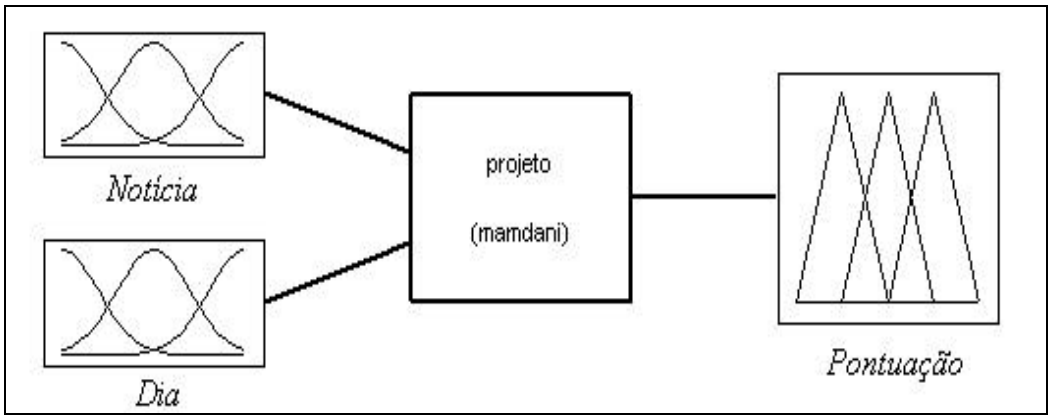


Figura 4.12: Base de conhecimento e inferência nebulosa.

A característica *notícia*, representada pela variável lingüística *Notícia*, segundo especialistas, envolve uma avaliação do especialista do impacto no leitor, em uma escala [0, 100], das manchetes, figuras, fotos e outras características da primeira página do jornal naquele determinado dia. Portanto, a variável lingüística *Notícia* pode ser definida em um intervalo de 0% a 100%, para modelar o impacto da primeira página. Por exemplo, quando o conteúdo da primeira página possuir algo muito importante, a notícia é avaliada de acordo com o valor 100%. A variável *Notícia* possui dois valores, *Normal* e *Especial*, como ilustra a Figura 4.13.

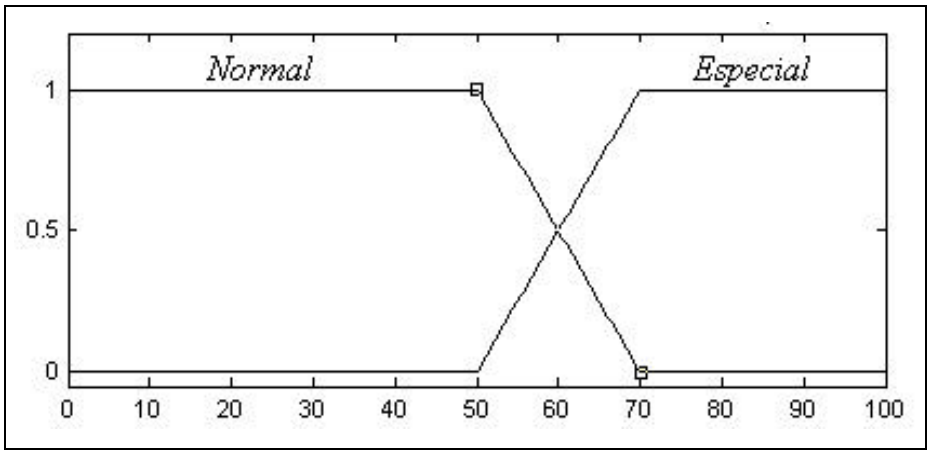


Figura 4.13: Variável Lingüística *Notícia.*

A natureza do *dia* é representada, por sua vez, pela variável lingüística *Dia*, que possui, conforme sugere o agrupamento, dois valores *Domingo-Feriado* e *Segunda-feira-a-Sábado*. Analogamente ao caso anterior, a variável *Dia* pode ser representada utilizando o intervalo de 0% a 100%, intervalo este que contém a avaliação das características daquele determinado dia. Por exemplo, se em um domingo houver eleições presidenciais, então esse dia não é considerado plenamente um domingo típico, pois muitas pessoas estarão trabalhando, podendo avaliá-lo, segundo o especialista, como um dia 50%. A variável lingüística *Dia* é ilustrada na Figura 4.14.

Como nos casos anteriores, a variável lingüística *Pontuação*, possui três valores, *Baixa*, *Média* e A*lta*, de acordo com o determinado pelo especialista. Essa variável

lingüística também tem, como universo, o intervalo de 0% a 100%, como mostra a Figura 4.15.

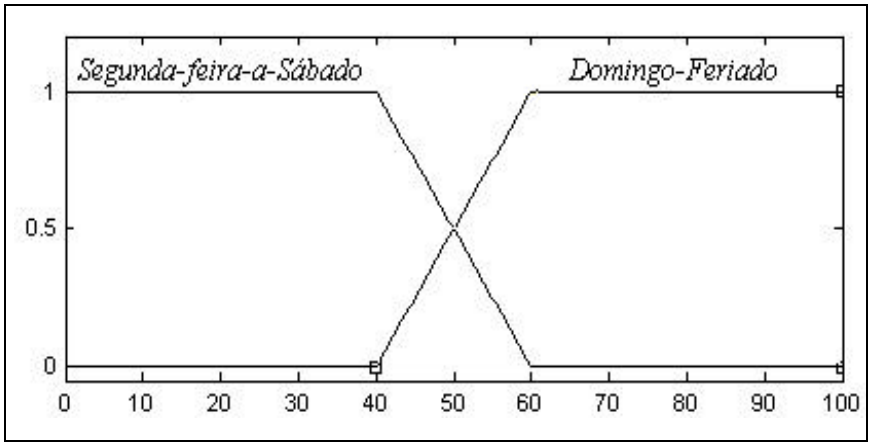


Figura 4.14: Variável Lingüística *Dia.*

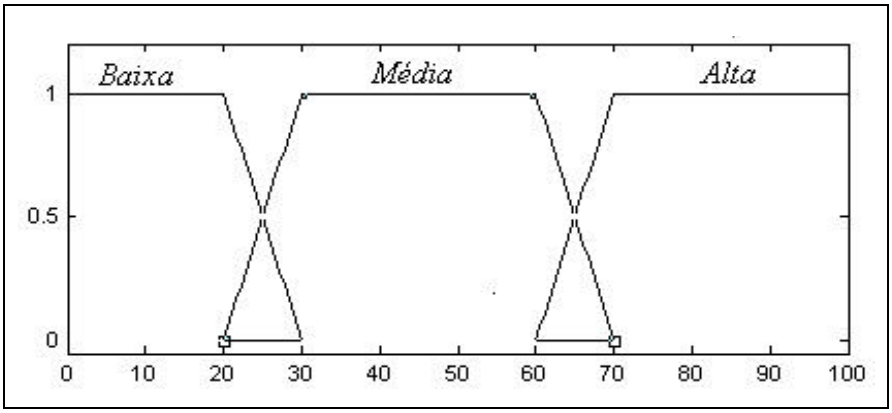


Figura 4.15: Variável Lingüística *Pontuação.*

Com as variáveis lingüísticas definidas, as regras são criadas, considerando-se o agrupamento de dados. Exemplos de regras criadas para a base de conhecimento e inferência nebulosa, seguem abaixo:

- Se *Notícia* é *Normal* e *Dia* é *Segunda-feira-a-Sábado* então *Pontuação* é *Baixa*.
- Se *Notícia* é *Normal* e *Dia* é *Domingo-Feriado* então *Pontuação* é *Média*.
- Se *Notícia* é *Especial* e *Dia* é *Segunda-feira-a-Sábado* então *Pontuação* é *Média*.
- Se *Notícia* é *Especial* e *Dia* é *Domingo-Feriado* então *Pontuação* é *Alta*.

O universo de discurso das variáveis lingüísticas, a base de conhecimento nebulosa, o modelo de Mamdani (seção 2.5.1) e defuzzificação via centróide (seção 2.5.1) gera uma superfície de decisão, conforme ilustra a Figura 4.16.

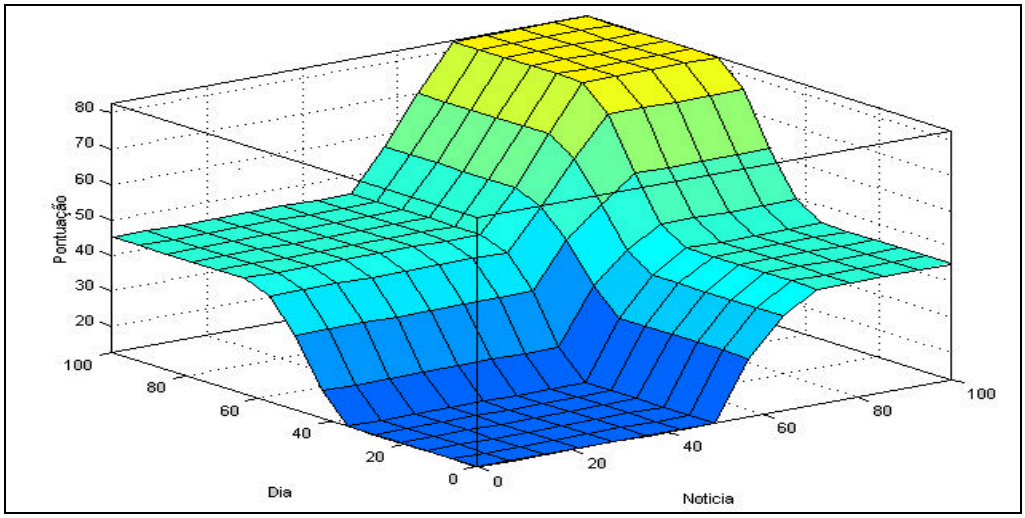
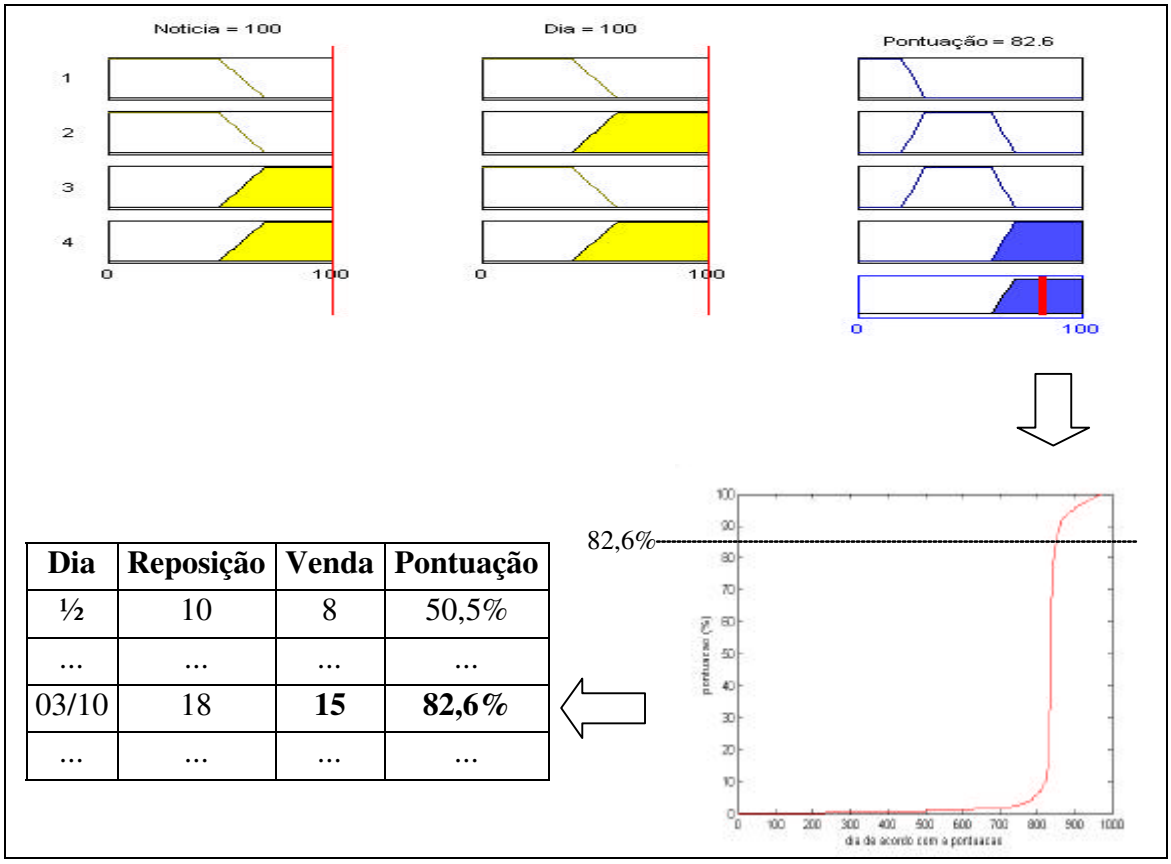


Figura 4.16: Superfície de decisão.

O sistema, utilizando a base de regras nebulosa gerada e as variáveis lingüísticas, *Notícia* e *Dia,* determina qual *Pontuação* deve ser considerada para a reposição nas bancas diariamente. Consultando-se a *Pontuação* obtida em cada dia para cada uma das bancas, determina-se o dia que a banca teve a mesma *Pontuação*, podendo assim fazer a reposição baseada no que foi vendido naquele determinado dia, como sugere o Exemplo 4.1 e a Figura 4.17.

Exemplo 4.1: Um especialista, em um domingo, avaliou a *notícia* como 100%, ou seja, naquele determinado domingo as fotos e manchetes da primeira página do jornal eram de grande impacto. Além disso, o dia foi também avaliado como 100% pois era um domingo em que na região de distribuição do jornal não havia indicativos de acontecimentos importantes, ou seja, a maioria das pessoas deveriam se comportar como em um domingo típico. Como ilustra a Figura 4.17, a *Pontuação*, de acordo com a base de regras nebulosas, seria de 82,6%, utilizando a notícia e o dia como 100%. Utilizando o gráfico de ganho da banca em consideração, determina-se o dia que a banca obteve a

mesma *Pontuação* e sugere-se que a reposição seja a mesma vendida naquele dia, ou seja, 15 unidades.



Dia	Reposição	Venda	Pontuação
½	10	8	50,5%
...	...	...	...
03/10	18	**15**	**82,6%**
...	...	...	...

Figura 4.17: Exemplo 4.1.

4.4 Resultados e discussão

Os resultados proporcionados pelo modelo proposto foram comparados com aqueles fornecidos pelo método tradicional de previsão, pelas séries temporais e por um preditor baseado em rede neural. O conjunto de dados utilizado para o treinamento de todos os métodos correspondem aos primeiros 944 dias. O conjunto utilizado para teste corresponde aos 30 dias restantes na base de dados. Para comparação, as Tabelas de 4-2 a 4-5 mostram os valores reais de reposição determinados pelo método tradicional para um período de uma semana e quatro bancas. Supõe-se que o domingo foi um dia em que a

primeira página não apresentava notícias de impacto, mas com um evento em que as pessoas passariam o dia trabalhando. Os dias da semana, ou seja, de segunda-feira a sábado, também foram avaliados como dias normais, com as características de primeira página avaliadas como de baixo impacto.

Tabela 4-2: Resultado real da banca 20.

	Reposição	**Venda**	**Perda**
Domingo	33	30	3
Segunda-feira	11	11	0
Terça-feira	13	12	1
Quarta-feira	10	10	0
Quinta-feira	15	12	3
Sexta-feira	11	11	0
Sábado	15	12	3
Perda Total			**10**

Tabela 4-3: Resultado real da banca 107.

	Reposição	**Venda**	**Perda**
Domingo	23	18	5
Segunda-feira	12	11	1
Terça-feira	16	11	5
Quarta-feira	12	10	2
Quinta-feira	16	11	5
Sexta-feira	12	10	2
Sábado	13	11	2
Perda Total			**22**

Tabela 4-4: Resultado real da banca 207.

	Reposição	**Venda**	**Perda**
Domingo	32	28	4
Segunda-feira	8	6	2
Terça-feira	6	5	1
Quarta-feira	9	6	3
Quinta-feira	5	4	1
Sexta-feira	5	5	0
Sábado	10	6	4
Perda Total			**15**

Tabela 4-5: Resultado real da banca 208.

	Reposição	**Venda**	**Perda**
Domingo	38	35	3
Segunda-feira	8	8	0
Terça-feira	9	8	1
Quarta-feira	6	5	1
Quinta-feira	11	8	3
Sexta-feira	8	8	0
Sábado	9	7	2
Perda Total			**10**

Nos resultados apresentados a seguir a Perda é calculada subtraindo-se a venda real, cujos valores estão mostrados nas Tabelas 4-2, 4-3, 4-4 e 4-5, da previsão realizada pelos métodos. A Perda Total é obtida somando-se as perdas no período (neste caso, uma semana). Valores negativos para a Perda ou Perda Total significam a quantidade de jornais que poderiam ser vendidos se estivessem disponíveis (escassez de reposição), enquanto que valores positivos indicam a quantidade que não foi vendida (excesso de reposição).

Primeiramente, de acordo com a seção 3.3, obteve-se o modelo autoregressivo de ordem dois, AR(2), utilizando o sétimo dia anterior e o dia anterior ao dia da previsão para prever a reposição corrente. Os resultados obtidos com o modelo de série temporal AR(2), obtidos de acordo com a *reposição*, feita em cada banca, num período de dois anos e meio, a fim de prever a *reposição* diária durante uma semana. As Tabelas de 4-6 a 4-9 mostram o comportamento da reposição neste período, utilizando os dados de *reposição* das bancas selecionadas anteriormente.

Tabela 4-6: Modelo autoregressivo: resultado para a banca 20.

	Previsão de Reposição	**Perda**
Domingo	39	9
Segunda-feira	11	0
Terça-feira	13	1
Quarta-feira	13	3
Quinta-feira	20	8
Sexta-feira	13	2
Sábado	17	5
Perda Total		**28**

Tabela 4-7: Modelo autoregressivo: resultado para a banca 107.

	Previsão de Reposição	**Perda**
Domingo	32	14
Segunda-feira	13	2
Terça-feira	15	4
Quarta-feira	14	4
Quinta-feira	23	12
Sexta-feira	14	4
Sábado	9	-2
Perda Total		**38**

Tabela 4-8: Modelo autoregressivo: resultado para a banca 207.

	Previsão de Reposição	**Perda**
Domingo	29	1
Segunda-feira	5	-1
Terça-feira	7	2
Quarta-feira	8	2
Quinta-feira	12	8
Sexta-feira	5	0
Sábado	11	5
Perda Total		**17**

Tabela 4-9: Modelo Autoregressivo: resultado para a banca 208.

	Previsão de Reposição	**Perda**
Domingo	34	-1
Segunda-feira	6	-2
Terça-feira	7	-1
Quarta-feira	9	4
Quinta-feira	16	8
Sexta-feira	7	-1
Sábado	13	6
Perda Total		**13**

Um preditor baseado em rede neural multicamada, seção 3.4, também foi criado utilizando o Tool Box Neural Networks do Matlab 6.1, da MathWorks, Inc.. Neste caso, trata-se a base de dados como dados históricos vistos como uma série temporal e a rede neural como um preditor. A rede foi treinada ao longo de 200 épocas, obtendo um erro quadrático médio de treinamento igual a 0,1. Utilizou-se taxa de aprendizado igual a 0,3, três entradas, *banca*, *reposição*$^{(t-7)}$, sendo a reposição de sete dias anteriores ao determinado dia, e *reposição*$^{(t-1)}$, que é a reposição do dia anterior, 8 neurônios na camada

intermediária e uma saída, a *previsão de reposição*. A arquitetura dessa rede é ilustrada na Figura 4.18. Os resultados obtidos pela rede, para as mesmas bancas consideradas anteriormente, são mostradas nas Tabelas de 4-10 a 4.13.

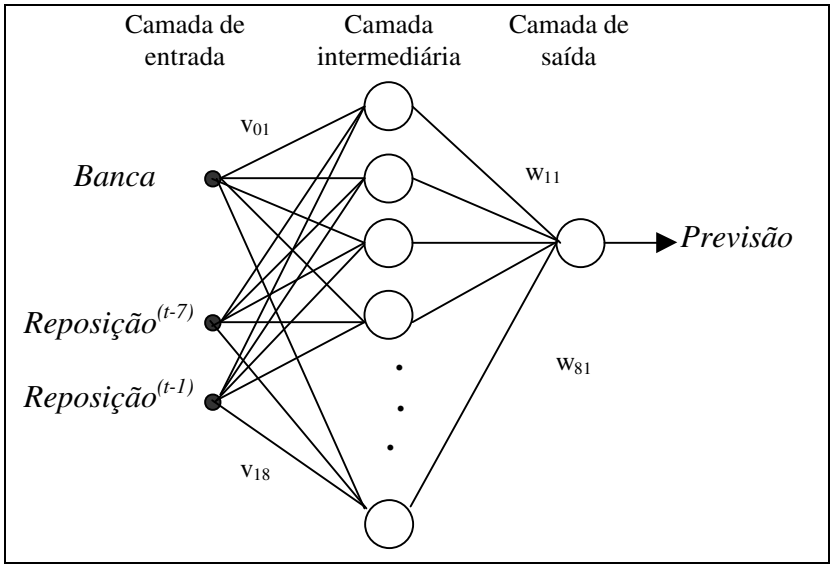


Figura 4.18: Rede neural para previsão

Tabela 4-10: Rede neural multicamada: resultado para a banca 20.

	Previsão de Reposição	**Perda**
Domingo	28	-2
Segunda-feira	10	-1
Terça-feira	8	-4
Quarta-feira	9	-1
Quinta-feira	10	-2
Sexta-feira	9	-2
Sábado	12	0
Perda Total		**-12**

Tabela 4-11: Rede neural multicamada: resultado para a banca 107.

	Previsão de Reposição	**Perda**
Domingo	18	0
Segunda-feira	10	-1
Terça-feira	8	-3
Quarta-feira	7	-3
Quinta-feira	8	-3
Sexta-feira	10	0
Sábado	11	0
Total		**-10**

Tabela 4-12: Rede neural multicamada: resultado para a banca 207.

	Previsão de Reposição	**Perda**
Domingo	25	-3
Segunda-feira	4	-2
Terça-feira	5	0
Quarta-feira	1	-5
Quinta-feira	2	-2
Sexta-feira	3	-2
Sábado	6	0
Perda Total		**-14**

Tabela 4-13: Rede neural multicamada: resultado para a banca 208.

	Previsão de Reposição	**Perda**
Domingo	40	5
Segunda-feira	4	-4
Terça-feira	8	0
Quarta-feira	4	-1
Quinta-feira	3	-5
Sexta-feira	4	-4
Sábado	7	0
Perda Total		**-9**

A aplicação da metodologia de previsão de reposição dos jornais foi aplicada em alguns casos, de acordo com o que foi proposto na seção 4.3. Quando um especialista classifica a primeira página do jornal como *Normal*, ou seja, ele classifica a primeira página do jornal em 42,7%. Por outro lado, um domingo em que tem um acontecimento na região de distribuição indica que muitas pessoas terão que trabalhar, é classificado pelo especialista como sendo 50,8%. Portanto, de acordo com a base de regras, obtém-se uma *Pontuação* de 50%, ou seja, *média*, como ilustra a Figura 4.19.

Em dias de segunda-feira a sábado, classificados pelo especialista como sendo 15%, e notícia de pouco impacto (*Normal*) para a região de distribuição, classificadas com o valor de 55%, de acordo com a base de regras nebulosas a *Pontuação* será *baixa*, ou seja, 25%, como mostra a Figura 4.20.

De acordo com as *Pontuações* obtidas nas Figuras 4.19 e 4.20, o gráfico de ganho e o conjunto de dados das bancas, obteve-se as previsões de *reposição* para todas as

bancas, conforme proposto na seção 4.3. As Tabelas de 4-14 a 4-17, ilustram a previsão de *reposição* para algumas bancas selecionadas anteriormente.

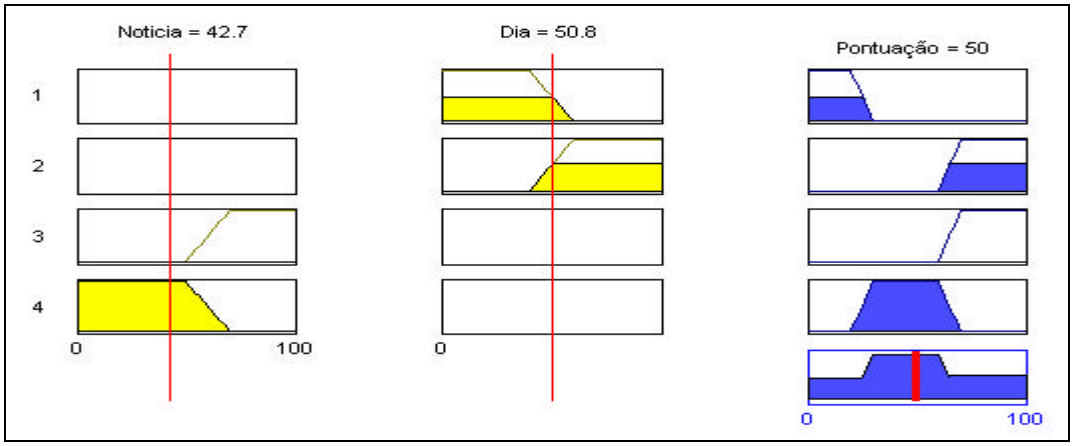


Figura 4.19: Resultado da *Pontuação* onde *Notícia* é *Normal* em um domingo.

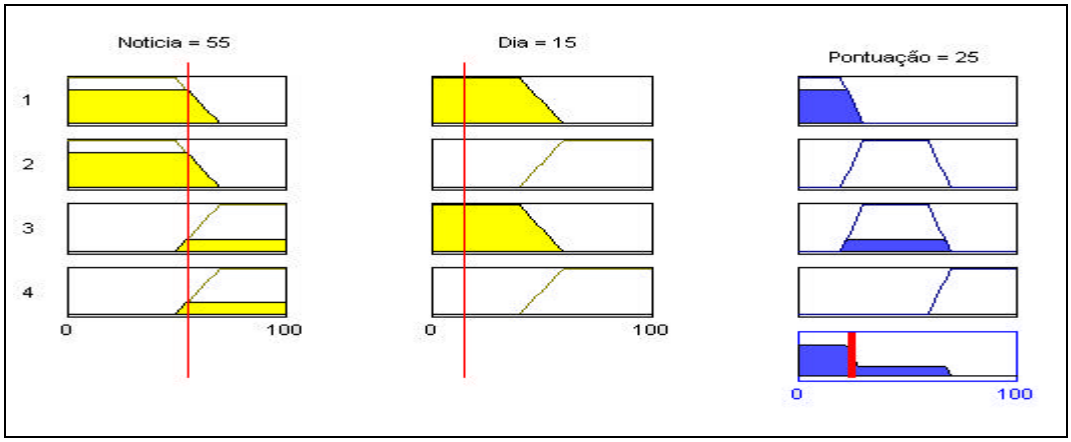


Figura 4.20: Resultado da *Pontuação* onde *Notícia* é *Normal* de segunda-feira à sábado

Tabela 4-14: Previsão de reposição para a banca 20.

Previsão de Reposição	**50%**	**25%**	**Perda**
Domingo	30		0
Segunda-feira		12	1
Terça-feira		12	0
Quarta-feira		12	2
Quinta-feira		12	0
Sexta-feira		12	1
Sábado		12	0
Perda Total			**4**

Tabela 4-15: Previsão de reposição para a banca 107.

Previsão de Reposição	**50%**	**25%**	**Perda**
Domingo	19		1
Segunda-feira		11	0
Terça-feira		11	0
Quarta-feira		11	1
Quinta-feira		11	0
Sexta-feira		11	1
Sábado		11	0
Perda Total			**3**

Tabela 4-16: Previsão de reposição para a banca 207.

Previsão de Reposição	**50%**	**25%**	**Perda**
Domingo	28		0
Segunda-feira		6	0
Terça-feira		6	1
Quarta-feira		6	0
Quinta-feira		6	2
Sexta-feira		6	1
Sábado		6	0
Perda Total			**4**

Tabela 4-17: Previsão de reposição para a banca 208.

Previsão de Reposição	**50%**	**25%**	**Perda**
Domingo	35		0
Segunda-feira		8	0
Terça-feira		8	0
Quarta-feira		8	3
Quinta-feira		8	0
Sexta-feira		8	0
Sábado		8	1
Perda Total			**4**

Tabela 4-18: Resumo dos resultados obtidos com os métodos aplicados

Banca 20	Resultado Real			Modelo autoregressivo		Rede neural		Metodologia de Previsão de Reposição		
	Reposição	Venda	Perda	Previsão	Perda	Previsão	Perda	50%	25%	Perda
Domingo	33	30	3	39	9	28	-2	30		0
Segunda-feira	11	11	0	11	0	10	-1		12	1
Terça-feira	13	12	1	13	1	8	-4		12	0
Quarta-feira	10	10	0	13	3	9	-1		12	2
Quinta-feira	15	12	3	20	8	10	-2		12	0
Sexta-feira	11	11	0	13	2	9	-2		12	1
Sábado	15	12	3	17	5	12	0		12	0
Total	**108**	**98**	**10**	**126**	**28**	**86**	**-12**		**102**	**4**
Perda Total (%)			**9%**		**22%**		**-14%**			**4%**

Tabela 4-19: Resumo dos resultados obtidos com os métodos aplicados

Banca 107	Resultado Real			Modelo autoregressivo		Rede neural		Metodologia de Previsão de Reposição		
	Reposição	Venda	Perda	Previsão	Perda	Previsão	Perda	50%	25%	Perda
Domingo	23	18	5	32	14	18	0	19		1
Segunda-feira	12	11	1	13	2	10	-1		11	0
Terça-feira	16	11	5	15	4	8	-3		11	0
Quarta-feira	12	10	2	14	4	7	-3		11	1
Quinta-feira	16	11	5	23	12	8	-3		11	0
Sexta-feira	12	10	2	14	4	10	0		11	1
Sábado	13	11	2	9	-2	11	0		11	0
Total	**104**	**82**	**22**	**120**	**38**	**72**	**-10**		**85**	**3**
Perda Total (%)			**21%**		**32%**		**-14%**			**4%**

Tabela 4-20: Resumo dos resultados obtidos com os métodos aplicados

Banca 207	Resultado Real			Modelo autoregressivo		Rede neural		Metodologia de Previsão de Reposição		
	Reposição	Venda	Perda	Previsão	Perda	Previsão	Perda	50%	25%	Perda
Domingo	32	28	4	29	1	25	-3	28		0
Segunda-feira	8	6	2	5	-1	4	-2		6	0
Terça-feira	6	5	1	7	2	5	0		6	1
Quarta-feira	9	6	3	8	2	1	-5		6	0
Quinta-feira	5	4	1	12	8	2	-2		6	2
Sexta-feira	5	5	0	5	0	3	-2		6	1
Sábado	10	6	4	11	5	6	0		6	0
Total	**75**	**60**	**15**	**77**	**17**	**46**	**-14**		**64**	**4**
Perda Total (%)			**20%**		**22%**		**-30%**			**6%**

Tabela 4-21: Resumo dos resultados obtidos com os métodos aplicados

Banca 208	Resultado Real			Modelo autoregressivo		Rede neural		Metodologia de Previsão de Reposição		
	Reposição	Venda	Perda	Previsão	Perda	Previsão	Perda	50%	25%	Perda
Domingo	38	35	3	34	-1	40	5	35		0
Segunda-feira	8	8	0	6	-2	4	-4		8	0
Terça-feira	9	8	1	7	-1	8	0		8	0
Quarta-feira	6	5	1	9	4	4	-1		8	3
Quinta-feira	11	8	3	16	8	3	-5		8	0
Sexta-feira	8	8	0	7	-1	4	-4		8	0
Sábado	9	7	2	13	6	7	0		8	1
Total	**89**	**79**	**10**	**92**	**13**	**70**	**-9**		**83**	**4**
Perda Total (%)			**11%**		**14%**		**-13%**			**5%**

Como pode ser observado nas Tabelas 4-6 a 4-9, o modelo AR(2), para esse problema, utilizando tanto os dados de *reposição* e *venda*, em alguns dias conseguiu um resultado razoável. Contudo, em outros o resultado dessa previsão pode levar a uma sobra razoavelmente grande ou em uma perda das vendas de jornais.

A aplicação das redes neurais, neste problema, com os resultados apresentados nas Tabelas 4-10 a 4-13, não se mostrou promissora, pois na maioria dos dias a reposição prevista é menor que a venda real dos respectivos dias. Isso pode ser justificado pelo fato de que, ao contrário de séries temporais de valores de grandezas físicas (e.g. vazão, energia, temperatura, etc.), caso onde redes neurais em geral são muito eficientes, a previsão de reposição depende fundamentalmente do comportamento, da percepção e na natureza humana.

Nos resultados obtidos pelo sistema proposto, Tabelas 4-14 a 4-17, nota-se que os números de jornais não vendidos são pequenos, tendo assim uma perda pequena em alguns dias.

Comparando os resultados reais com aqueles fornecidos por métodos tradicionais e pelo sistema proposto, observa-se que a perda do novo sistema é menor. Além disso, a previsão de reposição evita a falta de jornais em todos os dias, o que não acontece com os métodos baseados em regressão e rede neural.

4.5 Resumo

Este capítulo apresentou o problema de reposição no contexto da metodologia de previsão de reposição de jornais, e propôs um sistema de suporte à decisão para determinar a quantidade de reposição de jornais nas bancas. Os resultados obtidos pelo sistema proposto foram comparados com os resultados reais e os obtidos por redes neurais e regressão.

As Tabelas 4-18 a 4-21 resumem os resultados obtidos pelos métodos aplicados no problema.

CAPÍTULO 5

CONCLUSÃO

Este trabalho abordou o uso de técnicas de ECDB em problemas de decisão. A ênfase foi no aspecto de previsão da mineração de dados. Para tal, considerou-se um problema essencial na logística de distribuição de jornais: a previsão reposição a ser feita a cada uma das bancas em uma determinada região.

Para determinar a previsão para a reposição nas bancas, foram considerados vários métodos. O primeiro baseia-se em modelos de séries temporais e o segundo em redes neurais, com o algoritmo de retropropagação. O terceiro método de previsão baseia-se em agrupamento e regras nebulosas, fundamentado no método de Kaymak e Setnes (2001). Os dois primeiros algoritmos não se mostram adequados, pois em muitos dias a reposição é ou muito baixa ou muito alta em relação ao que se venderia naquele determinado dia, podendo provocar uma perda ou escassez de jornais. Contudo, em outros dias essas previsões proporcionam um resultado razoável.

A previsão baseada em agrupamento e regras nebulosos, proposta neste trabalho, obteve resultados mais próximos das necessidades reais. Além disso, ele permite o uso de conhecimento sobre as características da primeira página do jornal no dia de interesse.

Para uma complementação deste trabalho, sugere-se como um trabalho futuro uma extensão da previsão levando-se em consideração a localização de cada banca e os respectivos dias de funcionamento. Neste caso, é necessário uma base de dados que contemple um número maior de características das bancas. Poder-se-ia também desenvolver uma metodologia de previsão como aquela sugerida por Kandel *et. al*, 2001, onde se propõe um mecanismo específico de extração de conhecimento das bases de dados de séries temporais. Este processo deverá incluir a limpeza e a filtragem dos dados

das séries temporais, a identificação dos atributos mais importantes para a previsão e a extração de um conjunto de regras associativas para a previsão.

REFERÊNCIAS BIBLIOGRÁFICAS

Abelém, A., 1994, *"Redes Neurais Artificiais na Previsão de Séries Temporais"*, tese de Mestrado, Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

Amaral, F., 2001, *"Data Mining: Técnicas e Aplicações para o Marketing Direto"*, 1ª edição, Berkeley Brasil, São Paulo, SP, Brasil.

Ballini, R., 2000, *"Análise e previsão de vazões utilizando modelos de séries temporais, redes neurais e redes neurais nebulosas"*, tese de doutorado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, Campinas, SP, Brasil.

Bellman, R. e Zadeh, L., 1970, *"Decision-making in a fuzzy environment"*, Management Sci., vol. 17, pp. 141-164.

Bezdek, J., 1981, *"Pattern Recognition with Fuzzy Objective Function Algorithms"*, Plenum Press, NY, EUA.

Bose, I. e Mahapatra, R., 2001, *"Business data mining – a machine learning perspective"*, Information & Management 39, pp. 211-225.

Box, G. e Jenkins, G., 1976, *"Time Series Analysis: Forecasting and Control"*, Holden Day, 2ª ed., São Francisco, CA, EUA.

Buer, M., Woodruff, D. and Oslon, R., 1999, *"Solving the medium newspaper production/distribution problem"*, European Journal of Operational Research 115, pp. 237-253.

Cadiz, H., 1994,. *"The Development of a CHAID-based Model for CHITRA93"*, tese de mestrado, Virginia Polytechnic Institute and State University, Blascksburg, Virginia, EUA.

Cardoso, G. e Gomide, F., 2003a, *"Modelo de Previsão de Reposição de Jornais Baseado em Agrupamento Nebuloso e Regras Nebulosas"*, a ser publicado no IV

Encontro Nacional de Inteligência Artificial, XXIII Congresso da SBC, Campinas, SP, Brasil.

Cardoso, G. e Gomide, F., 2003b, *"Newspaper Replacement Prediction Model Based on Fuzzy Clustering and Rules"*, a ser publicado no International Fuzzy System Association World Congress, Istambul, Turquia.

Chae, Y., Ho, S. e Cho, K., 2001, *"Data mining approach to policy analysis in a health insurance domain",* International Journal of Medical Informatics 62, pp. 103-111.

Changchien, S. e Lu, T., 2001, *"Mining association rules procedure to support on-line recommendation by customers and products fragmentation"*, Expert Systems with Applications 20, pp 325-335.

Chen, M., Han, J. e Yu, P., 1996, *"Data mining: an overview from a database perspective"*, IEEE Transactions on Knowledge and Data Engineering, vol. 8, nº 6, pp. 866—883.

Fausett, L., 1994, *"Fundamentals of neural networks: architectures, algorithms, and applications"*, Prentice Hall, NJ, EUA.

Fayyad, U., Piatetssky-Shapiro, G., Smyth, P. e Uthurusamy, 1996, *"Advances in Knowledge Discovery an Data Mining",* CA: AAA Press/MIT Press, MA, EUA.

Feelders, A., Daniels, H., Holsheimer, M., 2000, *"Methodological and practical aspects of data mining"*, Information and Management, vol. 37, nº 5, pp. 271-281.

Fleischfresser, S., 2001, *"Abordagem de um Problema de Entrega de Jornais a Assinantes por Métodos Heurísticos e Estáticos"*, tese de mestrado, Departamento de Matemática da Universidade Federal do Paraná, Curitiba, PR, Brasil.

Frawley, W., Piatetsky-Shapiro, G. e Matheus, C., 1991, *"Knowledge Discovery in Databases: An Overview", in: U. Fayyad, G. Piatetsky-Shapiro, P.Smyth(Eds), Knowledge Discovery in Databases,* AAAI/MIT Press, Cambridge, MA, USA, pp. 1-27.

Freeman, J. e Skapura, D., 1992, *"Neural Networks Algorithms, Applications and Programmming Techniques"*, Addison-Wesley, MA, EUA.

Han, J. e Kamber, M., 2001, *"Data Mining: concepts and techniques"*, Morgan Kaufmann, EUA.

Haykin, S., 1994, *"Neural Networks: A Comprehensive Foundation"*, Macmillan College Publishing Company, New York.

Indurkhya, N. e Weiss, S., 1998, *"Predictive Data Mining – A Pratical Guide"*, Morgan Kaufmann Publishers, Inc., São Francisco, CA, EUA.

Inmon, W., Terdeman, R. e Imhoff. C., 2001, *"Data Warehousing – Como transformar informações em oportunidades de negócio"*, tradução para o português Melissa Kassner, 1ª edição, Berkeley Brasil, São Paulo, Brasil.

Jang, S., Sun C. e Mizutani, E., 1997, *"Neuro-Fuzzy and Soft Computing"*, Prentice Hall.

Kandel, A., Last, M. e Klein, Y, 2001, *"Knowledge Discovery in Time Series Databases"*, IEEE Transactions on Systems, Man and, Cybernetics – Part B: Cybernetics, vol. 31, nº 1, pp. 16-169.

Kass, G., 1975, *"Significance testing in Automatic Interaction Detection (AID)"*, Appplied Statistics, vol. 24, pp.178-189.

Kaymak, U e Setnes, M., 2000, *"Extended Fuzzy Clustering Algorithms"*, ERIM Report Series Research in Management.

Kaymak, U. e Setnes, M., 1998, *"Extended Fuzzy C-Means with Volume Prototypes and Cluster Merging"*, Proceding EUFIT'98, Aachen, Germany, pp. 1360-1364.

Kaymak, U. e Setnes, M., 2001, *"Fuzzy Modeling of Client Preference from Large Data Sets: An Application to Target Selection in Direct Marketing"*, IEEE Transactions on Fuzzy Systems, vol.9, nº 1, pp. 153-163.

Kaymak,U., Sousa, J. e Madeira, S., 2002, *"A Comparative Study of Fuzzy Target Selection Methods in Direct Marketing"*, IEEE International Conference on Fuzzy Systems, Honolulu, HI, EUA, pp. 1251-1256.

Klir, G. e Yuan, B., 1993, *"Fuzzy Sets and Fuzzy Logic - Theory and Apllication"*, Prentice Hall, NJ, EUA.

Lee, J., Yu, S. e Park, S., 2001,*"Design of Intelligent Data Sampling Methodology Based on Data Mining"*, IEEE Transactions on Robotics and Automation, vol. 17, nº 5, pp. 637-649.

Lehmann, T. e Eherler, D., 2001, *"Responder profiling with CHAID and dependency analysis"*, Data Mining for Marketing Applications, Freiburg, Alemanha.

Leung, Y., Ma, J. e Zhang, W., 2001, *"A new method for mining regression classes in large data sets"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, nº 1.

Lin, T., 2002, *"Issues in Modeling for Data Mining"*, Proceedings of the 26th Annual International Computer Software and Applications Conference (COMPSAC'02), Oxford, Inglaterra.

Mannila, H., 1996, *"Data mining: machine learning, statistics, and databases"*, VIII International Conference on Scientific and Statistical Database Management, Estocolmo, Suécia, p. 1-8.

Mannila, H., 1997, *"Methods and problems in data mining"*, Proceding on Conference. on Database Theory (ICDT'97), Delphi, Grécia, pg. 41-55.

Mitchell, T., 1997, *"Machine Learning"*, WCB/McGraw-Hill, NY, EUA.

Mitra S., Pal, S. e Mitra, P., 2002, *"Data Mining in Soft Computing Framework: A Survey"*, IEEE Transactions on Neural Networks, vol.13, nº 1, pp. 3-14.

Morettin, P. e Toloi, C., 1981, *"Modelos para Previsão de Séries Temporais"*, 13° Colóquio Brasileiro de Matemática, Rio de Janeiro, RJ, Brasil.

Morettin, P. e Toloi, C., 1987. *"Previsão de Séries Temporais"*, 2ª edição, Editora Atual, São Paulo, SP, Brasil.

Morgan, J. e Sonquist, J., 1963, *"Problems in the Analysis of Survey Data, and a Proposal"*, Journal of the American Statistical Association, nº 58, pp.415-435.

Mueller, A., 1996, *"Uma Aplicação de Redes Neurais Artificiais na Previsão do Mercado Acionário"*, tese de mestrado, Pós-graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil.

Nascimento, S., Mirkin, B. e Moura-Pires, F., 2000, *"A fuzzy clustering model of data and fuzzy c-means"*, 19º IEEE International Conference on Fuzzy Systems, 2000, vol.1, pg: 302 –307.

Neville , P., 1999, *"Decision Tree for Predictive Modeling"*, SAS Institute Inc, NC, EUA.

Neville, P. e Barlow, T., 2001, *"Case Study: Visualization for Decision Tree Analysis in Data Mining",* IEEE Symposium on Information Visualization 2001 (INFOVIS'01), San Diego, Canadá, pp. 149-152.

Pedrycz , W. e Hirota, K., 1999, *"Fuzzy Computing for Data Mining"*, Proc. of the IEEE, vol. 87, no. 9, pp. 1575-1600.

Pedrycz, W. e Gomide, F., 1998, *"An Introduction to Fuzzy Sets - Analysis and Design"*, MIT Press, MA, EUA.

Pedrycz, W., 1996, *"Data Mining and Fuzzy Modeling"*, NAFIPS, Biennial Conference of the North America Fuzzy Information Processing Society, Berkeley, CA, EUA, pp. 263-267.

Quinlan, R., 1983, *"Learning efficient classification procedures and their application to chess end games"*, In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, eds., Machine Learning - An Artificial Intelligence Approach, pp. 463 – 482, Tioga, Palo Alto, CA.

Ree, S. and Yoon, B., 1996, *"A Two-stage heuristic approach for the newspaper delivery problem"*, Computers and. Engineering., vol. 30, pp 501-509.

Richards, G., Rayward-Smith, V., Sönksen, P., Carey, S. e Weng, C., 2001, *"Data Mining for Indicators of Early Mortality in a Database of Clinical Records"*, Artificial Intelligence in Medicine 22, pp. 215-231.

Russell, S. e Lodwick, W., 1999, *"Fuzzy Clustering in Data Mining for Telco Database Marketing Campaigns"*, North American Fuzzy Information Processing Society (NAFIPS), NY, EUA, pp. 720-726 .

Sade, A. e Souza, J., 1996, *"Prospecção de Conhecimento em Bases de Dados Ambientais"*, C_3AD, Rio de janeiro, Brasil.

Saygin, Y. e Ulusoy, O., 2001, *"Automated construction of fuzzy event sets and its application to active databases"*, IEEE Transactions On Fuzzy Systems, vol. 9, nº 3, pp. 450-460".

Saygin, Y. e Ulusoy, O., 2002, *"Exploiting Data Mining Techniques for Broadcasting Data in Mobile Computing Environments"*, IEEE Transactions on Knowledge and Data Engineering, vol.14, nº 6.

Silva, L., 2003, *"Aprendizagem Participativa em Agrupamento Nebuloso de Dados"*, tese de mestrado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, Campinas, SP, Brasil.

Song, M. e Yoon,Y., 2000, *"A Comparative Study on Variable Selection Methods in Data Mining Software Packages"*, Proceedings of the 10º Japan and Korea Joint Conference of Statistics, Japan, pp.125-130.

Wilkinson, L., 1992, *"Tree Structured Data Analysis: AID, CHAID and CART"*, 1992 Sun Valley, ID, Sawtooth/Systat Joint Software Conference, USA.

Wu, H. e Lu, C., 2002, *"A Data Mining Approach for Spatial Modeling in Small Area Load Forecast"*, IEEE Transactions on Power Systems, vol. 17, nº 2.

Yager, R., 1990, *"A model of participatory learning"*, IEEE Transactions on Systems,Man and Cybernetics 20, n 5, pp. 1229-1234.

Zadeh,L. A., 1965, *"Fuzzy sets"*, Information and Control., vol. 8, pp.338-353.

APÊNDICE A

Entre muitos algoritmos de agrupamento nebulosos, a complexidade de alguns foi avaliada pelo tempo de processamento em relação ao número de pontos em *X* e o número de grupos existentes. Os algoritmos avaliados foram o FCM (seção 2.3.1), Agrupamento Participativo (AP) (seção 2.3.3) e E-FCM (seção 2.3.2)**.**

Estes algoritmos foram implementados em Java, em ambiente Windows 1998 e processador Intel Pentium III 800 MHz, 64 Mb memória RAM. Os tempos de processamento foram obtidos pela média aritmética de 30 execuções de cada algoritmo, com os centros inicializados aleatoriamente. A apresentação dos dados foi aleatória, sendo às mesmas para todos os algoritmos.

Os tempos de processamento dos algoritmos, ilustrados na Tabela A-1 e na Figura A.2, em relação ao número de pontos, foram obtidos com conjuntos de dados, onde o aumento foi somente o número de pontos, de modo que o número de grupos esperados é igual a dois, como mostra a Figura A.1.

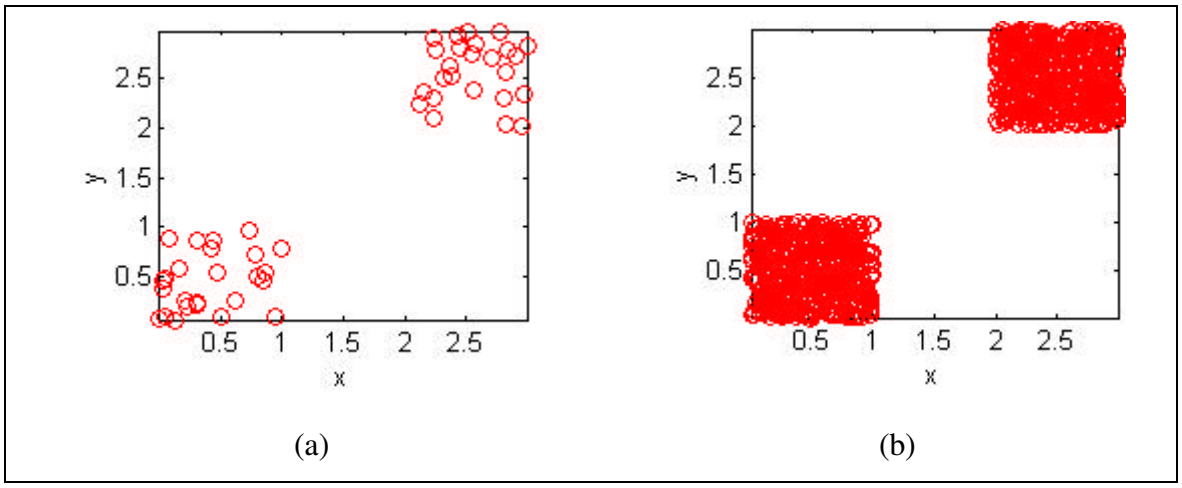


Figura A.1: Conjunto de dados original, com n = 50 (a) e n = 1000 (b).

Tabela A-1: Tempo de processamento[1] dos algoritmos em relação ao número de pontos[2].

nº. pontos	FCM	AP	E-FCM
50	0,00	3,22	1,27
100	0,00	10,64	7,32
160	0,00	16,11	12,58
240	0,00	29,66	23,22
300	0,00	51,71	45,55
500	0,00	101,20	133,62
800	0,00	217,75	489,28
1000	0,01	394,66	897,65

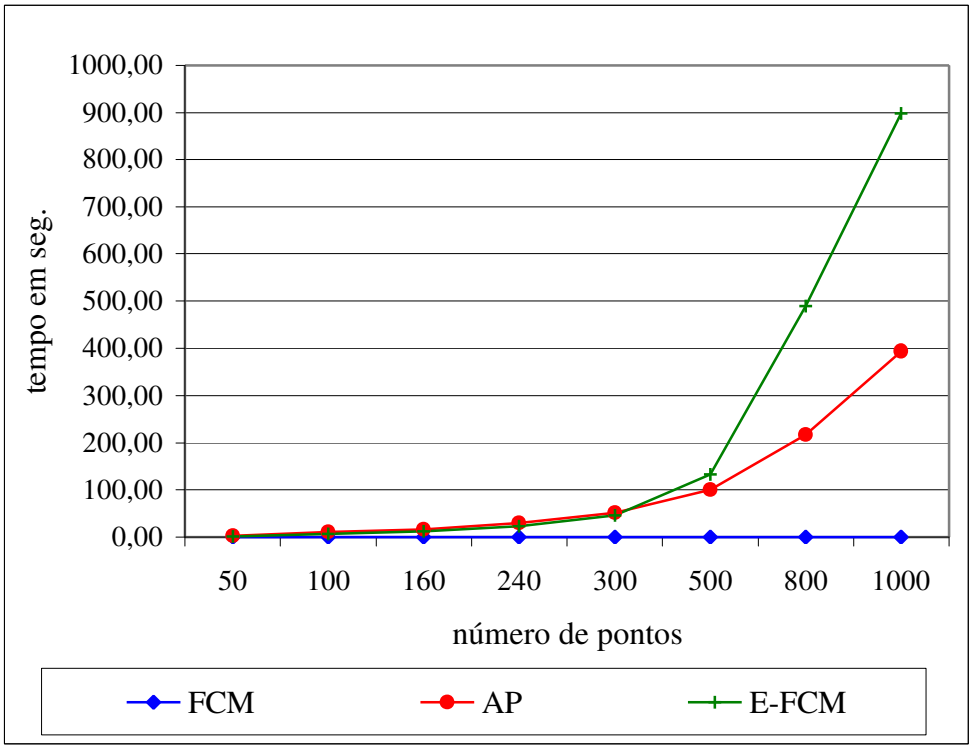


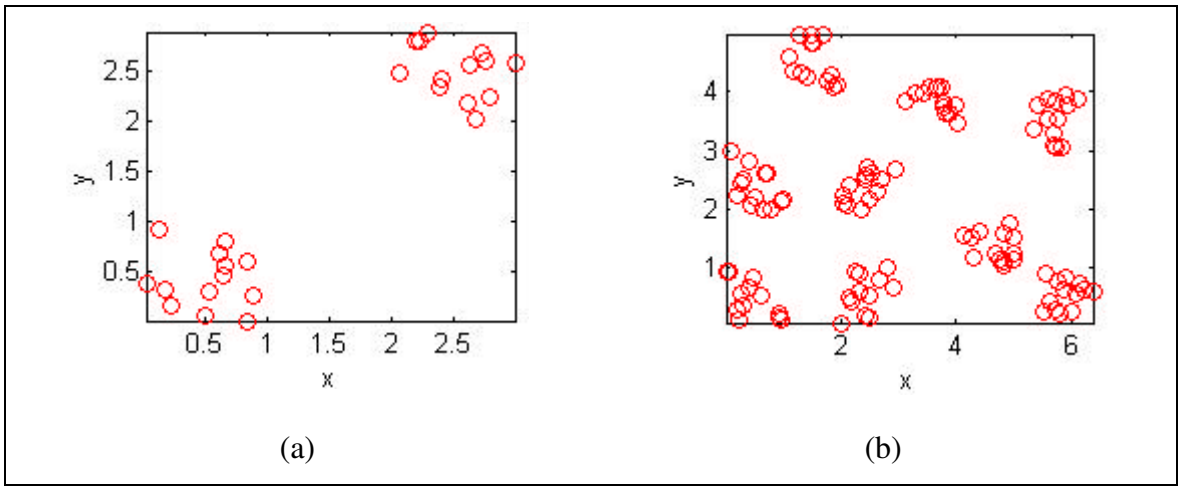
Figura A.2: Tempo de processamento dos algoritmos em relação ao número de pontos.

Os tempos de processamento dos algoritmos, como mostra a Tabela A-2 e a Figura A.4, em relação ao número de grupos, foram obtidos com conjuntos de dados

[1] Tempo de processamento em segundos.

[2] Os tempos de processamento menores que 1 milisegundo não foram considerados aqui.

onde cada grupo possui 13 pontos. Deste modo, o número de grupos esperados aumenta permanecendo o mesmo número de pontos em cada um dos grupos, como ilustra a Figura A.3.



(a) (b)

Figura A.3: Conjunto de dados original, com $c = 2$ (a) e $c = 9$ (b).

Tabela A-2: Tempo de processamento[3] dos algoritmos em relação ao número de grupos[4].

nº. grupos	**FCM**	**AP**	**E-FCM**
2	0,00	1,06	0.52
3	0,00	3,07	1.75
4	0,01	16,00	7.05
5	0,01	22,21	15.65
6	0,01	37,60	30.55
7	0,04	55,42	57.62
8	0,05	163,55	97.04
9	0,08	196,43	159.76

[3] Tempo de processamento em segundos.

[4] Os tempos de processamento menores que 1 milisegundo não foram considerados aqui.

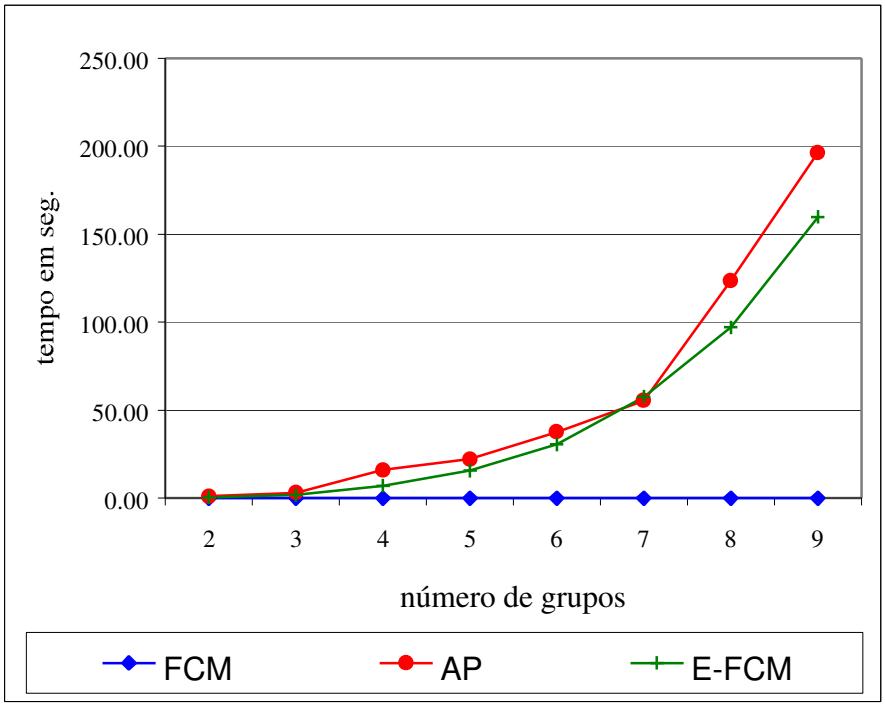


Figura A.4: Tempo de processamento dos algoritmos em relação ao número de grupos.