UNIVERSIDADE PRESBITERIANA MACKENZIE

MARCELO VIANA DONI

ANÁLISE DE *CLUSTER*: MÉTODOS HIERÁRQUICOS E DE PARTICIONAMENTO

São Paulo
2004

MARCELO VIANA DONI

ANÁLISE DE *CLUSTER*: MÉTODOS HIERÁRQUICOS E DE PARTICIONAMENTO

Trabalho apresentado à disciplina de
Trabalho de Graduação Interdisciplinar II,
como parte das exigências para a obtenção
do título de Bacharel em Sistemas de Informação
pela Faculdade de Computação e Informática
da Universidade Presbiteriana Mackenzie.

ORIENTADOR: ROGÉRIO DE OLIVEIRA

São Paulo
2004

RESUMO

Atualmente, a disponibilidade de avançados recursos computacionais e a relativa diminuição do custo operacional facilitou o armazenamento de dados em meio magnético. Devido ao acúmulo de uma grande quantidade de dados, existe a necessidade de identificar e utilizar as informações implícitas contidas nos dados, através de um processo conhecido como Extração de Conhecimento em Bases de Dados (*Knowledge Discovery in Databases*). Umas das etapas da Extração de Conhecimento em Bases de Dados é o processo de extração de informações em um banco de dados, sem conhecimento prévio, a Mineração de Dados (*Data Mining*). Neste estudo, descrevemos o processo de Extração de Conhecimento em Bases de Dados, Mineração de Dados e algumas de suas técnicas, tendo como enfoque os métodos hierárquicos e não-hierárquicos de análise de *cluster*, realizando três estudos de caso aplicando-se esses métodos.

Palavras-chave: Mineração de Dados. Análise de *Cluster*. Análise Multivariada.

ABSTRACT

Nowadays, the availability of advanced computational resources and the relative decrease of the operational costs, reduced the data storage in magnetic medium. When we have a large number of data, it is necessary to identify and to use the information included in this data, using a process known as knowledge discovery in databases. One of the steps of the knowledge discovery in databases is the process of extracting data in a database, without a previous knowledge, called data mining. In this study, we describe the knowledge discovery database process, the data mining, and some techniques focusing the hierarchical and non-hierarchical methods of cluster analysis and also three case studies applying those methods.

Keywords: Data mining, Cluster analysis, Multivariate analysis.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Durante os últimos anos, verificou-se um crescimento substancial da quantidade de dados armazenados em meios magnéticos. Avanços nas tecnologias de armazenamento de dados, o aumento na velocidade e capacidade dos sistemas, o barateamento dos dispositivos de armazenamento e a melhoria dos sistemas gerenciadores de banco de dados e *data warehouse*, têm permitido transformar essa enorme quantidade de dados em grandes bases de dados (FAYYAD et al., 1996). Estima-se que a cada 20 meses as empresas no mundo dobrem o volume de dados acumulados em seus computadores (DINIZ, 2000).

Esses dados, produzidos e armazenados em larga escala, são inviáveis de serem analisados por especialistas através de métodos tradicionais, tais como planilhas de cálculos e relatórios informativos operacionais, onde o especialista testa sua hipótese contra a base de dados (AURÉLIO, 1999). Com isso, surge a necessidade de se explorar esses dados para extrair informações e conhecimentos implícitos, a serem empregados na tomada de decisões.

Esse conhecimento é obtido por um processo organizado de transformação de dados em conhecimento, denominado extração de conhecimento em bases de dados (*Knowledge Discovery in Databases)*.

O termo extração de conhecimento em bases de dados refere-se às etapas no processo de descoberta e uso do conhecimento dos dados, onde a mineração de dados (*Data Mining*) é uma das etapas desse processo (JACKSON, 2002).

A mineração de dados é a fase onde são utilizadas diversas ferramentas computacionais na busca de padrões dos dados. Essas ferramentas empregam técnicas como indução, classificação, analise de *cluster*, regressão, redes neurais, redes bayesianas, algoritmos genéticos, entre outras.

Neste estudo, damos enfoque a alguns métodos de análise de *cluster*. Esses métodos buscam ajudar o usuário a entender a estrutura natural em um conjunto de dados. A análise de *cluster* é uma das técnicas mais utilizadas no processo de mineração de dados para descoberta de agrupamentos e identificação de importantes distribuições e padrões para entendimento dos dados (HALDIKI, 2001).

O agrupamento em bancos de dados é o processo de separar o conjunto de dados em componentes que refletem padrões consistentes de comportamento, particionando o banco de dados de forma que cada partição ou grupo seja similar de acordo com algum critério ou métrica. Uma vez que os padrões tenham sido estabelecidos, estes podem ser utilizados para "desmontar" os dados em subconjuntos mais compreensíveis e também podem prover subgrupos de uma população para futuras análises. Por exemplo, um banco de dados poderia ser utilizado para a geração de perfis de *marketing* direcionado onde a resposta prévia às campanhas de mala direta geraria um perfil das pessoas que responderam. A partir disso, faz-se a previsão de resposta e filtra-se a lista de mala direta para obter melhor resultado.

As metodologias de análise de *cluster* têm sido largamente utilizadas em numerosas aplicações, incluindo reconhecimento de padrões, análise de dados, processamento de imagens e pesquisa de mercado (JAIN, 1999).

Dada sua importância nos processos de mineração de dados, apresentam-se, nesse trabalho, diversas técnicas de análise de *cluster*, seus algoritmos e características, além de experimentos aplicando-se alguns métodos em diferentes conjuntos de dados.

O trabalho está organizado da seguinte forma. No capítulo 2, trata-se a extração de conhecimento em bases de dados, mostrando as atividades executadas em cada uma de suas fases. No capítulo 3, trata-se, de forma concisa, o conceito de mineração de dados e algumas técnicas utilizadas nessa fase. No capítulo 4, tratam-se, com maiores detalhes, os métodos hierárquicos e não-hierárquicos de análise de *cluster*, apresentando os algoritmos,

características e exemplos de cada método. Além disso, abordam-se, brevemente, outras técnicas de análise de *cluster*, como agrupamentos *fuzzy* e mapas de Kohonen. O capítulo 8, dedica-se a três estudos de caso, onde foram aplicadas as técnicas de análise de *cluster*. No capítulo 9, apresentam-se as conclusões deste estudo. A seguir, as referências bibliográficas que deram suporte ao estudo desenvolvido.

2 EXTRAÇÃO DE CONHECIMENTO EM BASES DE DADOS

O processo de extração de conhecimento em bases de dados (*Knowledge Discovery in Databases – KDD*) foi proposto em 1989 por Fayyad, referindo-se às etapas que produzem conhecimento[1] a partir de dados e, principalmente, à de mineração dos dados, que é a fase que transforma dados em informação (FAYYAD et al., 1996). Esse processo é formado pela intersecção de diferentes áreas, como aprendizado de máquina, inteligência artificial, estatística, visualização dos dados, entre outras.

Amplamente utilizada no processo de extração de conhecimento em bases de dados (ECBD), a inteligência artificial é uma área da ciência que busca o desenvolvimento de sistemas inteligentes baseados em aspectos do comportamento humano, tais como aprendizado, percepção, raciocínio, evolução e adaptação. Suas técnicas mais utilizadas são redes neurais artificiais, indução de regras e algoritmos genéticos.

O processo de ECBD tem como objetivo encontrar conhecimento a partir de um conjunto de dados e utilizá-lo em um processo decisório.

A definição do problema que se deseja solucionar é fundamental para o processo de ECBD. Essa definição requer que a pessoa que solicita a tarefa entenda o problema existente e tenha um objetivo bem especificado, ou seja, aquilo que se deseja conhecer ou extrair. Tendo definido o problema, pode-se fixar metas para os objetivos da tarefa de ECBD.

Esse processo é centrado na interação entre três classes de usuários, sendo: o especialista de domínio, o usuário que deve possuir amplo conhecimento do domínio da aplicação e deve fornecer apoio para a execução do processo; o analista, que é o usuário especialista no processo de extração de conhecimento e responsável por sua execução, o qual deve, ainda, conhecer profundamente as etapas que compõem o processo; e o usuário final,

que utiliza o conhecimento extraído para auxilia-lo nos processos de tomadas de decisão. Portanto, um requisito importante é que o conhecimento produzido seja compreensível e útil para os usuários finais.

A ECBD é um conjunto de atividades contínuas que compartilham o conhecimento descoberto a partir de bases de dados, e é composto pelas etapas de: seleção de dados; processamento e limpeza; transformação; mineração de dados (*Data Mining)* e interpretação dos resultados. A figura 2.1 ilustra as etapas envolvidas no processo de ECBD.

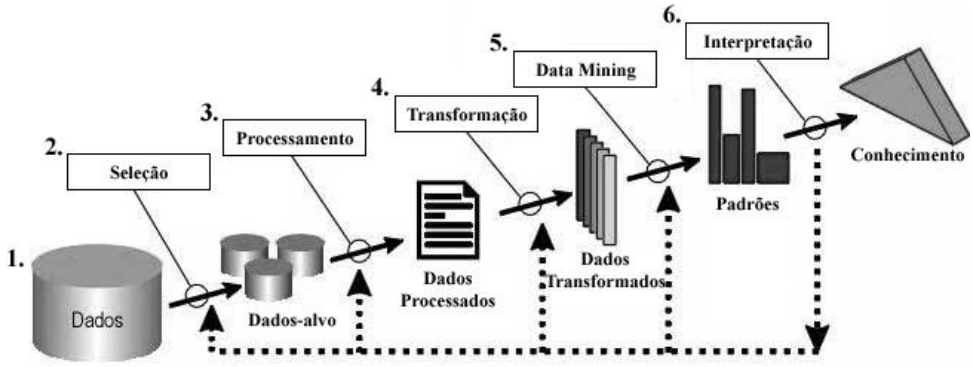

Figura 2.1: Processo de extração de conhecimento em bases de dados e suas etapas.

Pode haver intersecção entre as fases, e os resultados produzidos numa fase podem ser utilizados para melhorar os resultados das seguintes. Isso significa que o processo de ECBD é iterativo, buscando aprimorar os resultados a cada iteração.

Essas etapas podem diferir no tempo e no esforço consumido. A preparação dos dados, por exemplo, que inclui a seleção, processamento e transformação dos dados, pode consumir de 60% a 80% do tempo total do processo, sendo a maior parte do tempo empregada na limpeza de dados, segundo Diniz (2000). Cada uma dessas etapas é tratada a seguir.

2.1 SELEÇÃO DOS DADOS

[1] Conhecimento refere-se à informações armazenadas ou modelos usados por uma pessoa ou máquina para interpretar, predizer e responder de forma correta a algo do mundo real (HAYKIN, 1994).

A seleção de dados tem como objetivo identificar as origens internas e externas da informação, extraindo um subconjunto de dados necessário para a aplicação da mineração de dados, selecionando apenas atributos relevantes aos objetivos do processo de extração de conhecimento.

As variáveis selecionadas podem ser do tipo categórica ou quantitativa. As variáveis categóricas assumem valores finitos e são nominais ou ordinais. As variáveis ordinais apresentam uma ordem entre seus valores possíveis. Exemplos de variáveis nominais são estado civil (“solteiro”, “casado”, “divorciado”) e sexo (“masculino”, “feminino”). Um exemplo de variável ordinal é grau de instrução (“primeiro grau”, “segundo grau”) (DINIZ, 2000).

As variáveis quantitativas assumem valores numéricos. Podem ser do tipo contínua (assumem valores reais) ou discreta (assumem valores de um conjunto finito ou infinito enumerável). Exemplos de variáveis discretas são o número de empregados e número de filhos. Exemplos de variáveis contínuas são a receita e taxa.

A etapa de seleção otimiza o tempo de processamento da técnica de mineração, visto que ele apenas trabalhará com um subconjunto de atributos, diminuindo o espaço de busca da mineração.

É importante que essa fase seja guiada pelos objetivos do processo de extração de conhecimento, a fim de que o conjunto de dados gerado apresente características necessárias para que os objetivos sejam alcançados.

2.2 PROCESSAMENTO DOS DADOS

O processamento dos dados tem como objetivo assegurar a qualidade dos dados selecionados. Como o resultado do processo de extração possivelmente será utilizado no processo de tomada de decisão, a qualidade dos dados é um fator extremamente importante.

Esta etapa inicia-se com uma revisão geral da estrutura dos dados e a definição de medidas de qualidade, utilizando uma combinação de métodos estatísticos e técnicas de visualização de dados.

Entre os problemas tratados na etapa de processamento dos dados, encontramos:

- Eliminação de dados duplicados - são removidos dados duplicados e/ou corrompidos;
- Tratamento de *outliers* - são valores significantemente fora do esperado para uma variável;
- Valores faltantes - valores que não estão presentes no conjunto selecionado e valores inválidos que foram eliminados durante a detecção de *outliers*.

2.3 TRANSFORMAÇÃO DOS DADOS

O objetivo desta fase é converter o conjunto bruto de dados em uma forma padrão de uso, tornando os dados úteis para a mineração.

Devido às restrições de espaço em memória e tempo de processamento, o número de atributos disponíveis para análise pode inviabilizar a mineração de dados. Com isso, técnicas de redução de dados são aplicadas, sendo feitas de três modos:

- Redução do número de exemplos – deve ser feita mantendo as características do conjunto de dados original, por meio da geração de amostras representativas dos dados;
- Redução do número de atributos – realizada pelo especialista do domínio, seleciona-se um subconjunto dos atributos de forma que não tenha impacto na qualidade do conjunto final;
- Redução do número de valores de um atributo – consiste na redução do número de valores de um atributo, geralmente utilizando-se técnicas de discretização e suavização de valores. A discretização de um atributo consiste na substituição de um atributo contínuo por um atributo discreto, por meio do agrupamento de seus valores. A

suavização de valores agrupa determinados atributos em um valor numérico que o represente, podendo ser, por exemplo, a média ou a mediana.

Essa etapa auxilia na redução do tempo de processamento para a técnica de mineração, diminuindo o espaço de busca. As transformações devem, entretanto, preservar ao máximo, nas amostras geradas, as informações presentes nos dados brutos.

2.4 MINERAÇÃO DE DADOS

É a etapa mais importante do processo de ECBD e caracteriza-se pela existência de uma técnica de mineração capaz de extrair conhecimento implícito de um banco de dados em função de um objetivo proposto. Este item é abordado com maiores detalhes no capítulo 3.

2.5 INTERPRETAÇÃO E AVALIAÇÃO

As técnicas de mineração de dados podem ter gerado uma quantidade enorme de padrões, dos quais podem não ser relevantes ou interessantes ao usuário.

Um dos objetivos principais do processo de ECBD é que o usuário possa compreender e utilizar o conhecimento descoberto. Mas, podem ocorrer casos em que os modelos são muito complexos ou não fazem sentido para os especialistas.

Existem algumas formas de se caracterizar a qualidade dos resultados obtidos, como a compreensibilidade e a interessabilidade. A compreensibilidade de um conjunto de regras relaciona-se com a facilidade de interpretação dessas regras por um ser humano. Assim, quanto menor a quantidade de regras de um dado modelo e menor o número de condições por regra, maior será a compreensibilidade das regras descobertas. A interessabilidade é uma medida de qualidade que tenta estimar o quanto de conhecimento interessante ou inesperado existe, e deve combinar fatores numa medida que reflete como o especialista julga o padrão.

Após a análise do conhecimento, caso esse não seja de interesse do usuário final ou não cumpra os objetivos propostos, o processo de extração pode ser repetido, ajustando-se os parâmetros ou melhorando o processo de escolha dos dados, para a obtenção de melhores resultados em uma próxima iteração.

2.6 CONCLUSÃO

Em resumo, a extração de conhecimento em bases de dados é composta por etapas iterativas, sendo a seleção dos dados, o processamento dos dados, a transformação dos dados, a mineração de dados e a interpretação e avaliação. Dada a importância da mineração de dados no processo de ECBD, no capítulo seguinte, são tratadas, brevemente, conceitos e técnicas empregadas em mineração de dados.

3 MINERAÇÃO DE DADOS

A descoberta de conhecimento em bases de dados é um campo de pesquisa que tem crescido rapidamente, e cujo desenvolvimento tem sido dirigido ao benefício de necessidades práticas, sociais e econômicas. A motivação para esse crescimento está ligada à existência de uma poderosa tecnologia de coleta, armazenamento e gerenciamento de grande quantidade de dados. Muitos desses dados possuem informações valiosas, como tendências e padrões que podem ser usadas em tomadas de decisões (REZENDE, 2003).

A mineração de dados é termo que se refere aos métodos e técnicas computacionais utilizadas para a extração de informações em bases de dados.

As técnicas de mineração de dados são aplicadas sobre bancos de dados operacionais ou *Data Warehouse*, nos quais os dados são preparados antes de serem armazenados. Um *Data Warehouse* é um conjunto de dados baseado em assuntos, integrado, não-volátil e variante em relação ao tempo, para apoio às decisões gerenciais (INMON, 1997). Ele tem por objetivo oferecer organização, gerenciamento e integração de bancos de dados, assim como ferramentas de exploração dos mesmos, usado, geralmente, em aplicações de suporte à tomada de decisão. Devido às características do *Data Warehouse*, geralmente, as técnicas de mineração são aplicadas sobre ele.

Nas próximas seções, são apresentadas, brevemente, algumas técnicas utilizadas em mineração de dados, tratando-se, em maiores detalhes, as técnicas de análise de *cluster* no capítulo 4.

3.1 ESTATÍSTICA

A estatística é aplicada à maioria das técnicas de mineração de dados. Com isso, existe dificuldade na distinção entre mineração de dados e estatística. A mineração de dados deve ser tratada como uma adaptação de técnicas estatísticas tradicionais, visando a análise de grandes bases de dados (DINIZ, 2000).

As técnicas estatísticas mais utilizadas em mineração de dados são (JACKSON, 2002):

- Estatística descritiva - trata-se de medidas de locação (média, mediana) e variabilidade (desvio padrão), medidas de freqüências e porcentagens, tabelas de contingência e coeficientes de correlação;
- Técnicas de visualização - incluem histogramas e gráficos de dispersão;
- Análise de discriminante - utilizada para criar regras de associação para dados previamente classificados, aplicando essa regra para associar novos elementos às classes;
- Analise fatorial - utilizada para reduzir o número de variáveis de um conjunto de dados e detectar uma relação estrutural entre elas;
- Análise de regressão - modelos utilizados para estabelecer relações entre variáveis, de forma que uma variável possa ser predita através de outras.

Uma revisão detalhada desse tópico pode ser encontrada em (JONHSON, 1992).

3.2 INDUÇÃO

A indução é o processo de se obter uma hipótese a partir dos dados e fatos existentes. Em geral, os fatos são os registros existentes nos bancos de dados e a hipótese é uma árvore de decisão que deverá segmentar os dados de forma significativa.

A árvore de decisão é usada para criar regras com os nós, servindo como pontos de decisão. Assim, a árvore de decisão alcança sua decisão executando uma seqüência de testes,

onde cada nó interno na árvore corresponde a um teste do valor de uma das propriedades, e as ramificações a partir do nó são identificadas com os valores possíveis. Cada nó da folha na árvore especifica o valor a ser retornado se aquela folha for alcançada. A figura 3.1 traz um exemplo de árvore decisão, onde é definido se um cliente vai ou não esperar por uma mesa em um restaurante (RUSSELL, 1995). As árvores de decisão podem, também, envolver probabilidades na decisão de um caminho ou outro. Uma revisão concisa desse tópico pode ser encontrada em (RUSSEL, 1995).

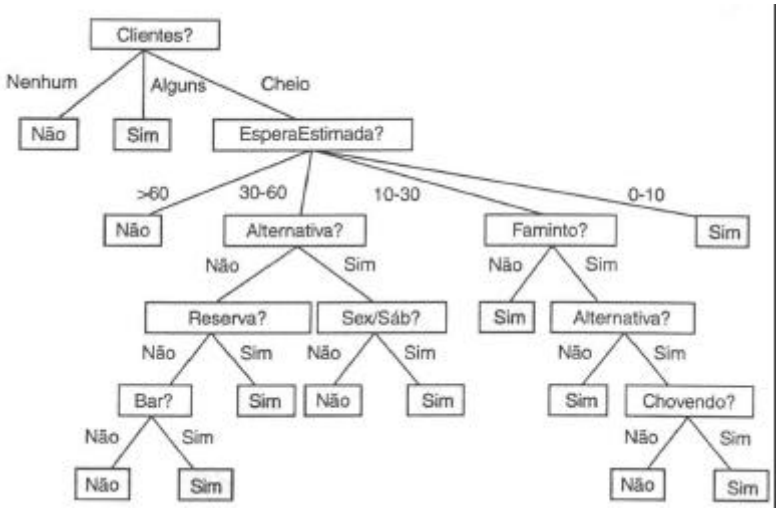

Figura 3.1: Exemplo de árvore de decisão.

3.3 ALGORITMOS GENÉTICOS

Os algoritmos genéticos são métodos generalizados de busca e otimização que simulam os processos naturais de evolução. Um algoritmo genético é um procedimento iterativo para evoluir uma população de organismos e é usado em mineração de dados para obter relações sobre dependências entre variáveis, na forma de algum formalismo interno (AURÉLIO, 1999).

Os algoritmos genéticos usam os operadores de seleção, cruzamento e mutação para desenvolver sucessivas gerações de soluções. Com a evolução do algoritmo, somente as

soluções com maior poder de previsão tendem a sobreviver, até os organismos convergirem em uma solução ideal.

A técnica de algoritmos genéticos é apropriada às tarefas de classificação e segmentação e vem sendo amplamente difundida. Para uma revisão sobre o uso desses algoritmos e maiores referências, ver (REZENDE, 2003).

3.4 CLASSIFICAÇÃO

A tarefa de classificação consiste em construir um modelo de algum tipo que possa ser aplicado a dados não classificados visando categorizá-los em classes. Um objeto é examinado e classificado de acordo com classes pré-definidas (REZENDE, 2003).

São exemplos de tarefas de classificação: classificar pedidos de créditos como de baixo, médio e alto risco; esclarecer pedidos de seguros fraudulentos; identificar a forma de tratamento na qual um paciente está mais propício a responder, baseando-se em classes de pacientes que respondem bem a determinado tipo de tratamento médico. Para maiores referências a respeito dessa técnica, consulte (FREITAS, 1998) e (WEISS, 1998).

3.5 ANÁLISE DE *CLUSTER*

A análise de *cluster* é um processo de partição de uma população heterogênea em vários subgrupos mais homogêneos. No agrupamento, não há classes pré-definidas, os elementos são agrupados de acordo com a semelhança, o que a diferencia da tarefa de classificação.

Este tópico será tratado com detalhes mais adiante, constituindo-se o tema principal deste trabalho.

3.6 REDES NEURAIS ARTIFICIAIS

Redes neurais artificiais (RNAs) são modelos matemáticos que se assemelham às estruturas biológicas e que têm capacidade computacional adquirida por meio de aprendizado e generalização (HAYKIN, 1994).

O aprendizado em RNAs está associado à capacidade de adaptarem seus parâmetros como conseqüência de sua interação com o ambiente de aprendizagem. O processo de aprendizado é interativo, e por meio dele, a RNA deve melhorar o seu desempenho gradativamente (HAYKIN, 1994). O critério de desempenho que determina a qualidade do modelo neural e o ponto de parada de treinamento são preestabelecidos pelos parâmetros de treinamento. Por sua vez, a generalização de uma RNA está associada à capacidade de fornecer respostas coerentes para dados não apresentados a ela, previamente, durante o treinamento.

O processamento da informação em RNAs é feito por meio de estruturas neurais artificiais, em que o armazenamento e o processamento da informação são realizados de maneira paralela e distribuída por elementos processadores relativamente simples, chamados de neurônios artificiais.

O neurônio artificial é dividido em duas seções funcionais, conforme figura 3.1.

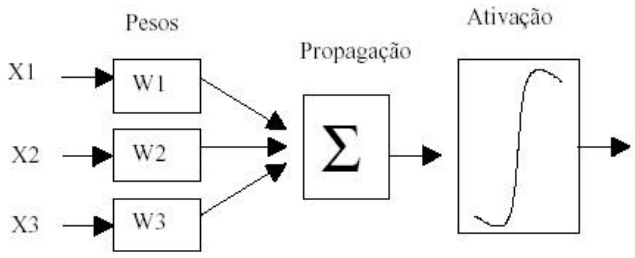

Figura 3.2: Esquema de um neurônio artificial.

A primeira seção combina todas as entradas que alimenta o neurônio, podendo ser estímulos do sistema ou saídas de outros neurônios. Essa etapa indica como as entradas serão computadas (regra de propagação). A segunda seção recebe esse valor e faz um cálculo

determinando o grau de importância da soma ponderada utilizando uma função de ativação. Essa função determina a que grau uma soma causará uma excitação ou inibição do neurônio.

A figura 3.2 traz um exemplo de uma RNA, onde os círculos representam os neurônios artificiais e as linhas representam os pesos das conexões. A camada que recebe os dados é chamada camada de entrada e a camada que mostra o resultado é chamada camada de saída. A camada interna, onde se localiza o processamento interno, é tradicionalmente chamada de camada oculta. Uma RNA pode conter uma ou várias camadas ocultas, de acordo com a complexidade do problema, sendo, em geral, até 3 camadas..

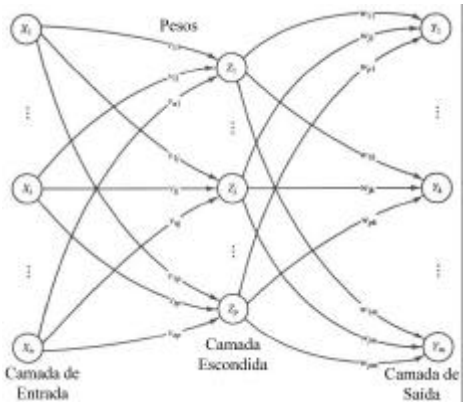

Figura 3.3: Exemplo de uma rede neural.

Uma das características mais significativas de uma RNA é a capacidade de aprendizagem. A aprendizagem é um processo no qual uma rede neural adapta seus parâmetros através de um processo de interação com o ambiente na qual esta inserida (HAYKIN, 1994). Para isso, a rede deve passar por uma fase de treinamento. O objetivo do treinamento é fazer com que a aplicação de um conjunto de entradas produza um conjunto de saídas desejado ou no mínimo um conjunto de saídas consistentes. Durante o treinamento, os pesos da rede gradualmente devem convergir para determinados valores, tal que a aplicação dos vetores de entrada produza as saídas desejadas.

Os procedimentos de aprendizagem que levam as RNAs a aprenderem determinadas tarefas podem ser classificados em duas classes, sendo o aprendizado supervisionado e o não-supervisionado.

O aprendizado supervisionado caracteriza-se pela existência de um elemento supervisor, externo à rede, que tem a função de monitorar a resposta da rede para cada vetor de entrada. O conjunto de treinamento é formado por pares de entrada e saída. O ajuste de pesos é feito de maneira que a resposta da rede para o vetor de entrada se aproxime dentro de limites de tolerância preestabelecidos. Cada resposta da rede é comparada pelo supervisor com o valor esperado para se obter a direção de ajuste dos pesos (BRAGA, 2003).

O aprendizado não supervisionado caracteriza-se pela não existência de saídas desejadas para as entradas, sendo o conjunto de treinamento formado apenas pelos vetores de entrada. Portanto, não há supervisor externo e o ajuste de pesos é obtido apenas com base nos valores dos vetores de entrada (BRAGA, 2003).

As principais aplicações de redes neurais em mineração são classificação, análise de *cluster*, aproximação de funções, previsão e verificação de tendências.

De um modo geral, a arquitetura de uma RNA recebe um atributo como entrada através da camada de entrada. Tratando-se de aprendizado não supervisionado, não existem atributos objetivo que possam ser utilizados para corrigir os pesos da rede. Esse tipo de aprendizado aplica-se, por exemplo, em tarefas de análise de cluster. As redes auto-organizáveis, por exemplo, Kohonen, baseadas em aprendizado competitivo, destacam-se como um bom algoritmo.

Entretanto, em algoritmos supervisionados, os atributos objetivo são modelados pela camada de saída da rede. Deste modo, o algoritmo pode estimar o quanto a saída desejada está distante da saída real. O algoritmo mais comum em RNAs com aprendizado supervisionado é o *back propagation*. Seu objetivo é minimizar a função erro entre a saída real da rede e a saída

desejada utilizando o método do gradiente descendente (HAYKIN, 1994). *Back-propagation* é utilizado para classificar, aproximar funções, prever e verificar tendências.

Para a tarefa de classificação também são utilizadas as redes neurais probabilísticas, baseadas em classificadores bayesianos e as redes RBF (*Radial-Basis Function*), baseadas em funções *gaussianas*. Esses algoritmos geram curvas de densidade de probabilidade, fornecendo resultados com bases estatísticas. Esses resultados indicam o grau de evidência sobre o qual se baseia a decisão. Entretanto, essa metodologia só funciona bem se existir um número suficiente de exemplos na base de dados. Maiores detalhes e referências sobre redes neurais artificiais podem ser obtidos em (HAYKIN, 1994), (FAUSETT, 1994) e (BRAGA, 2003).

3.7 CONCLUSÃO

Nesse capítulo, verificou-se, de forma concisa, algumas técnicas utilizadas em mineração de dados, onde cada uma possui características próprias, podendo ser empregadas em conjunto para a resolução de problemas. Além das técnicas descritas, outras são utilizadas, como redes bayesianas (HECKERMAN, 1996) e sistemas neuro-*fuzzy* (AURÉLIO, 1999).

Dentre as técnicas mencionadas, a análise de *cluster* será tratada em detalhes no próximo capítulo.

4 ANÁLISE DE *CLUSTER*

Nesse capítulo, descreve-se alguns métodos de análise de *cluster*, tendo como objeto de estudo, os métodos hierárquicos e os não-hierárquicos de agrupamento. Primeiramente, destacam-se medidas de similaridade e o uso da matriz de similaridade. Em seguida, a descrição dos métodos com seus algoritmos, funções distância e algumas características, trazendo um exemplo da formação dos grupos em cada método. Na última seção, apresentam-se, brevemente, outros métodos, como agrupamentos *fuzzy* e mapas de Kohonen.

A análise de *cluster* busca agrupar elementos de dados baseando-se na similaridade entre eles. Os grupos são determinados de forma a obter-se homogeneidade dentro dos grupos e heterogeneidade entre eles.

A necessidade de classificar elementos em grupos por suas características está presente em várias áreas do conhecimento, como nas ciências biológicas, ciências sociais e comportamentais, ciências da terra, medicina, informática, entre outras.

Tendo em vista a dificuldade de se examinar todas as combinações de grupos possíveis em um grande volume de dados, desenvolveram-se diversas técnicas capazes de auxiliar na formação dos agrupamentos.

Uma análise de *cluster* criteriosa exige métodos que apresentem as seguintes características (ZAIANE, 2003):

- Ser capaz de lidar com dados com alta dimensionalidade;
- Ser “escalável” com o número de dimensões e com a quantidade de elementos a serem agrupados;
- Habilidade para lidar com diferentes tipos de dados;
- Capacidade de definir agrupamentos de diferentes tamanhos e formas;
- Exigir o mínimo de conhecimento para determinação dos parâmetros de entrada;

- Ser robusto à presença de ruído;
- Apresentar resultado consistente independente da ordem em que os dados são apresentados;

Em geral, algoritmo algum atende a todos esses requisitos e, por isso, é importante entender as características de cada algoritmo para a escolha de um método adequado a cada tipo de dado ou problema (HALDIKI, 2001).

4.1 MEDIDAS DE SIMILARIDADE

A maioria dos métodos de análise de *cluster* requer uma medida de similaridade entre os elementos a serem agrupados, normalmente expressa como uma função distância ou métrica.

Seja *M* um conjunto, uma métrica em *M* é uma função d: M×M $\rightarrow \Re$, tal que para quaisquer x, y, z $\in$ *M,* tenhamos:

1. $d_{xy} > 0$ - para todo $x \neq y$

2. $d_{xy} = 0 \Leftrightarrow x = y$

3. $d_{xy} = d_{yx}$

4. $d_{xy} \leq d_{xz} + d_{zy}$

4.1.1 Distância Euclidiana

A distância euclidiana é a distância geométrica no espaço multidimensional.

A distância euclidiana entre dois elementos $X = [X_1, X_2, ..., Xp]$ e $Y = [Y_1, Y2, ..., Yp]$, é definida por:

$$d_{xy} = \sqrt{(X1 - Y1)^2 + (X2 - Y2)^2 + ... + (Xp - Yp)^2} = \sqrt{\sum_{i=1}^{p} (Xi - Yi)^2} \qquad (4.1)$$

Exemplo. Considerando-se elementos como pontos no plano (espaço euclidiano $\Re^2$), a distância entre os elementos $X_0 = (1,2)$ e $X_1 = (3,4)$ é dada por:

$d_{x_0x_1} = \sqrt{(3-1)^2 + (4-2)^2} = \sqrt{8} = 2{,}83$ (figura 4.1)

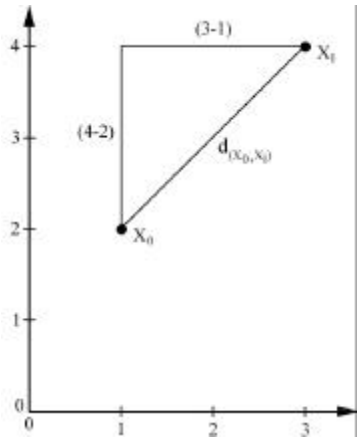

Figura 4.1: Distância euclidiana entre os pontos X_0 e X_1 no plano.

4.1.2 Distância Euclidiana Quadrática

A distância euclidiana quadrática é definida pela expressão:

$$d_{xy} = (X1 - Y1)^2 + (X2 - Y2)^2 + \ldots + (Xp - Yp)^2 = \sum_{i=1}^{p} (Xi - Yi)^2 \qquad (4.2)$$

Exemplo. Considerando-se os mesmos pontos X_0 e X_1 do exemplo anterior, observa-se a intensificação da distância:

$d_{x_0x_1} = (3-1)^2 + (4-2)^2 = 8$

4.1.3 Distância de *Manhattan*

A distância de *Manhattan* é definida pela expressão:

$$d_{xy} = |X1 - Y1| + |X2 - Y2| + \ldots + |Xp - Yp| = \sum_{i=1}^{p} |Xi - Yi| \qquad (4.3)$$

Em muitos casos, a distância de *Manhattan* apresenta resultados similares ao da distância Euclidiana. Entretanto, nessa medida, o efeito de uma grande diferença entre uma das dimensões de um elemento é minimizado, já que a mesma não é elevada ao quadrado.

Exemplo. Empregando-se os pontos do exemplo anterior, temos:

$$d_{x_0x_1} = |3-1| + |4-2| = |2| + |2| = 4$$

4.1.4 Distância de *Chebychev*

A distância de *Chebychev* é apropriada no caso em que se deseja definir dois elementos como diferentes, se apenas umas das dimensões é diferente. Ela é definida por:

$$d_{xy} = máximo\left(|X1 - Y1|, |X2 - Y2|, ..., |Xp - Yp|\right) \quad (4.4)$$

Exemplo. Considerando-se os pontos $X_2 = (9,2)$ e $X_3 = (2,5)$, a distância de *Chebychev* é dada por:

$$d_{x_2x_3} = máximo(|9-2|, |2-5|) = (|7|, |-3|) = 7$$

As medidas de similaridade são utilizadas na análise de *cluster* de forma a determinar a distância entre elementos. Essa distância, é normalmente representada na forma de matriz, ou seja, em uma matriz de similaridade.

A matriz de similaridade é simétrica e utiliza, na maioria dos casos, a distância Euclidiana.

Exemplo. Considerando os elementos da tabela 4.1, obtemos a matriz de similaridade *D*.

ELEMENTO	X	Y
1	4	3
2	2	7
3	4	7
4	2	3
5	3	5
6	6	1

Tabela 4.1: Elementos do exemplo para matriz de similaridade.

$$D = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{bmatrix} 0 & 4,47 & 4 & 2 & 2,24 & 2,83 \\ 4,47 & 0 & 2 & 4 & 2,24 & 7,21 \\ 4 & 2 & 0 & 4,47 & 2,24 & 6,32 \\ 2 & 4 & 4,47 & 0 & 2,24 & 4,47 \\ 2,24 & 2,24 & 2,24 & 2,24 & 0 & 5 \\ 2,83 & 7,21 & 6,32 & 4,47 & 5 & 0 \end{bmatrix}$$

- sendo:

5 a distância Euclidiana entre os elementos 6 e 5;

2,83 a distância Euclidiana entre os elementos 1 e 6;

6,32 a distância Euclidiana entre os elementos 3 e 6.

4.2 MÉTODOS HIERÁRQUICOS

O método hierárquico de *cluster* consiste em uma série de sucessivos agrupamentos ou sucessivas divisões de elementos, onde os elementos são agregados ou desagregados. Os métodos hierárquicos são subdivididos em métodos aglomerativos e divisivos.

Os grupos, nos métodos hierárquicos, são geralmente representados por um diagrama bi-dimensional chamado de dendograma ou diagrama de árvore. Neste diagrama, cada ramo representa um elemento, enquanto a raiz representa o agrupamento de todos os elementos. A figura 4.2 traz um exemplo de dendograma.

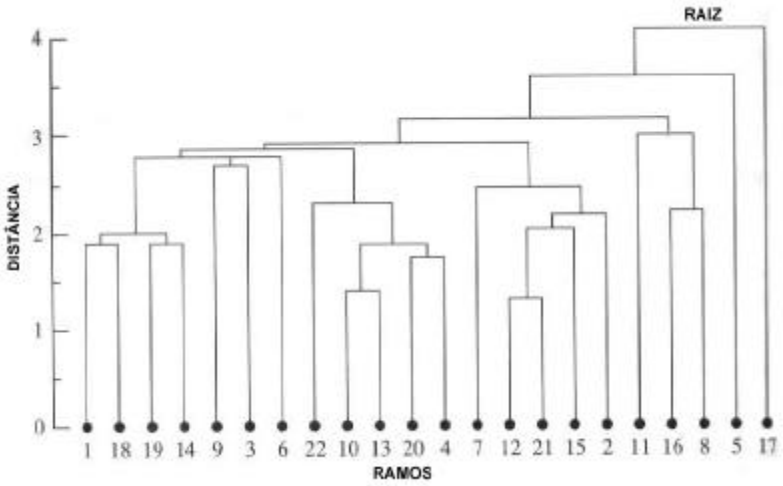

Figura 4.2: Exemplo de dendograma.

Através do dendograma e do conhecimento prévio sobre a estrutura dos dados, deve-se determinar uma distância de corte para definir quais serão os grupos formados. Essa decisão é subjetiva, e deve ser feita de acordo o objetivo da análise e o número de grupos desejados.

Exemplo. Considerando o dendograma da figura 4.3, pode-se verificar que com três diferentes cortes, obtemos diferentes grupos:

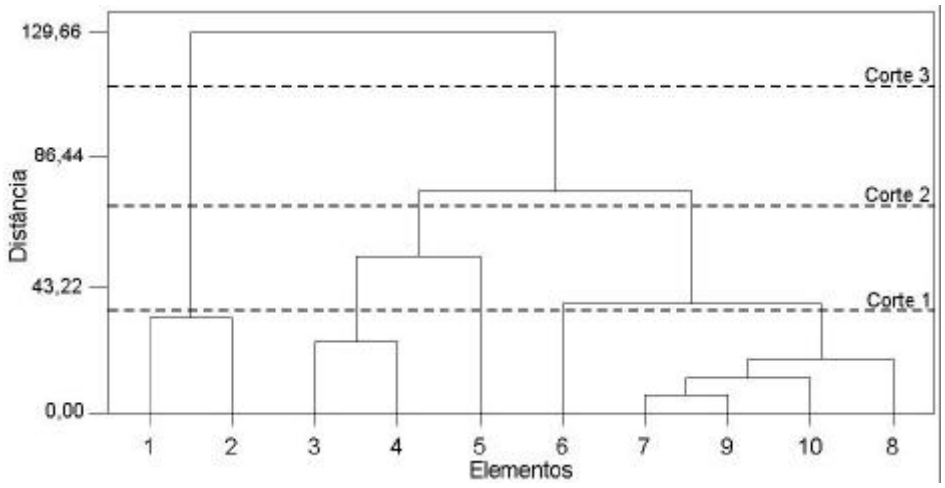

Figura 4.3: Exemplo no qual o dendograma é cortado em três diferentes níveis.

No corte 1, verifica-se a existência de cinco grupos, sendo (1,2), (3,4), (5), (6) e (7,9,10,8). No corte 2, o número de grupos diminui para três, sendo (1,2), (3,4,5) e

(6,7,9,10,8). Considerando o corte 3, o número de grupos diminui para dois, sendo (1,2) e (3,4,5,6,7,9,10,8).

Dessa forma, o usuário deverá escolher o corte mais adequado às suas necessidades e à estrutura dos dados.

4.2.1 Métodos Aglomerativos

No método aglomerativo, cada elemento inicia-se representando um grupo, e a cada passo, um grupo ou elemento é ligado a outro de acordo com sua similaridade, até o último passo, onde é formado um grupo único com todos os elementos.

Existe uma variedade de métodos aglomerativos, que são caracterizados de acordo com o critério utilizado para definir as distâncias entre grupos. Entretanto, a maioria dos métodos parecem ser formulações alternativas de três grandes conceitos de agrupamento aglomerativo (ANDERBERG, 1973):

1) Métodos de ligação (*single linkage*, *complete linkage*, *average linkage*, *median linkage*);
2) Métodos de centróide;
3) Métodos de soma de erros quadráticos ou variância (método de *Ward*).

Os métodos aglomerativos possuem a complexidade de tempo da ordem de $O(n^2 \log n)$ e a complexidade de espaço da ordem de $O(n^2)$, onde n é o número de elementos (JAIN, 1999).

De modo geral, os métodos aglomerativos utilizam os passos de um algoritmo padrão, conforme descrito na figura 4.4. A diferença entre os métodos ocorre no passo 5, onde a função distância é definida de acordo com cada método (JOHNSON, 1992). Essas distâncias estão definidas para cada método nas próximas seções.

Entrada: Uma base de dados com ***N*** elementos.
Saída: Um conjunto de grupos.
1. Iniciar com ***N*** grupos, contendo um elemento em cada grupo e uma matriz de similaridade $D_{N'N}$;
2. Repetir;
3. Localizar a menor distância d_{UV} (maior similaridade);
4. Atualizar a matriz ***D***, retirando os elementos ***U*** e ***V***;
5. Atualizar a matriz ***D***, adicionando as novas distâncias do grupo ***(U,V)***;
6. Até ***N*-1**, quando todos elementos estarão em um único grupo.

Figura 4.4: Algoritmo padrão.

Exemplo. Considerando-se os elementos da tabela 4.1, obteve-se a matriz *D* abaixo, onde, aplicando uma iteração do algoritmo padrão, temos:

$$D = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{bmatrix} 0 & 4{,}47 & 4 & 2 & 2{,}24 & 2{,}83 \\ 4{,}47 & 0 & 2 & 4 & 2{,}24 & 7{,}21 \\ 4 & 2 & 0 & 4{,}47 & 2{,}24 & 6{,}32 \\ 2 & 4 & 4{,}47 & 0 & 2{,}24 & 4{,}47 \\ 2{,}24 & 2{,}24 & 2{,}24 & 2{,}24 & 0 & 5 \\ 2{,}83 & 7{,}21 & 6{,}32 & 4{,}47 & 5 & 0 \end{bmatrix}$$

A menor distância d_{UV} está entre os elementos 1 e 4, e 3 e 2. Portanto, $d_{14} = d_{32} = 2$. Como a distância d_{14} é a primeira a aparecer na matriz, os elementos 1 e 4 serão considerados no primeiro agrupamento.

Assim, a matriz resultante após uma iteração do algoritmo será:

$$\begin{array}{c} (1,4) \\ 2 \\ 3 \\ 5 \\ 6 \end{array} \begin{bmatrix} 0 & d(1,4)2 & d(1,4)3 & d(1,4)5 & d(1,4)6 \\ d2(1,4) & 0 & 2 & 2{,}24 & 7{,}21 \\ d3(1,4) & 2 & 0 & 2{,}24 & 6{,}32 \\ d5(1,4) & 2{,}24 & 2{,}24 & 0 & 5 \\ d6(1,4) & 7{,}21 & 6{,}32 & 5 & 0 \end{bmatrix}$$

4.2.1.1 Método *Single Linkage* ou ligação por vizinho mais próximo

O método de ligação por vizinho mais próximo emprega a distância de valor mínimo:

$$d_{(UV)W} = \min(d_{UW}, d_{VW}) \qquad (4.5)$$

Exemplo. Utilizando a matriz do exemplo anterior, as distâncias mínimas encontradas são:

CÁLCULO DAS DISTÂNCIAS	VALOR
$d_{(1,4)2} = \min(d_{12}, d_{42}) = \min(4,47;4)$	4
$d_{(1,4)3} = \min(d_{13}, d_{43}) = \min(4;4,47)$	4
$d_{(1,4)5} = \min(d_{15}, d_{45}) = \min(2,24;2,24)$	2,24
$d_{(1,4)6} = \min(d_{16}, d_{46}) = \min(2,83;4,47)$	2,83

Assim, a matriz resultante será:

$$\begin{array}{c} \\ (1,4) \\ 2 \\ 3 \\ 5 \\ 6 \end{array} \begin{bmatrix} 0 & 4 & 4 & 2,24 & 2,83 \\ 4 & 0 & 2 & 2,24 & 7,21 \\ 4 & 2 & 0 & 2,24 & 6,32 \\ 2,24 & 2,24 & 2,24 & 0 & 5 \\ 2,83 & 7,21 & 6,32 & 5 & 0 \end{bmatrix}$$

As demais iterações serão repetidas como descrito no algoritmo padrão, sempre calculando, para esse método, as distâncias mínimas entre os elementos ou grupos.

A figura 4.5 traz a seqüência dos grupos formados em cada iteração do algoritmo.

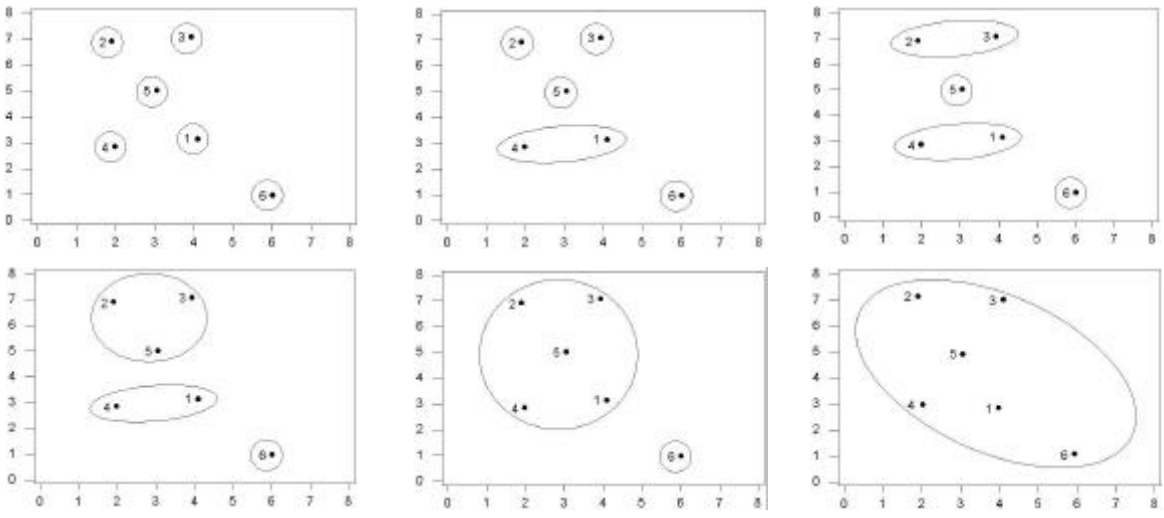

Figura 4.5: Seqüência de agrupamentos realizada no método de ligação por vizinho mais próximo.

A figura 4.6 traz o dendograma gerado pelo do método de ligação por vizinho mais próximo.

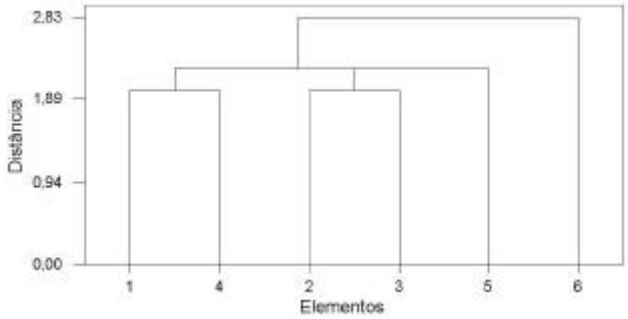

Figura 4.6: Dendograma aplicando o método de ligação por vizinho mais próximo.

Algumas características desse método são (ANDERBERG, 1973):

- Em geral, grupos muito próximos podem não ser identificados;
- Permite detectar grupos de formas não-elípticas;
- Apresenta pouca tolerância a ruído, pois tem tendência a incorporar os ruídos em um grupo já existente;
- Apresenta bons resultados tanto para distâncias Euclidianas quanto para outras distâncias;
- Tendência a formar longas cadeias (encadeamento).

Encadeamento é um termo que descreve a situação onde há um primeiro grupo de um ou mais elementos que passa a incorporar, a cada iteração, um grupo de apenas um elemento. Assim, é formada uma longa cadeia, onde torna-se difícil definir um nível de corte para classificar os elementos em grupos (ROMESBURG, 1984), conforme figura 4.7.

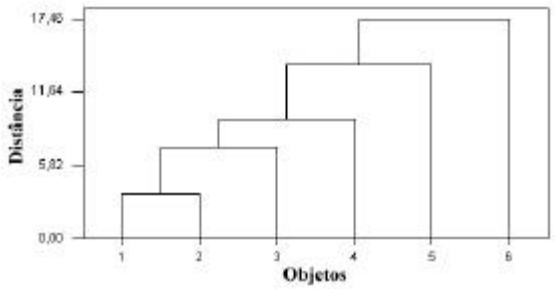

Figura 4.7: Fenômeno do encadeamento.

Esse fenômeno ocorre em dados com a distribuição mostrada na figura 4.8, onde cada elemento tem como vizinho mais próximo o grupo formado na iteração anterior.

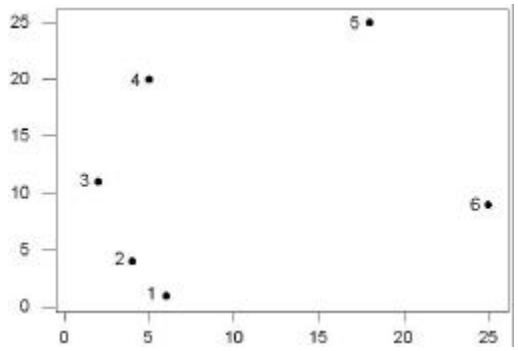

Figura 4.8: Gráfico de dispersão de dados com estrutura de encadeamento.

4.2.1.2 Método *Complete Linkage* ou ligação por vizinho mais distante

Nesse método, é empregada a distância máxima, dada por:

$$d_{(UV)W} = \max(d_{UW}, d_{VW}) \tag{4.6}$$

Exemplo. Utilizando a matriz do exemplo anterior, as distâncias máximas encontradas são:

CÁLCULO DAS DISTÂNCIAS	VALOR
$d_{(1,4)2} = \max(d_{12}, d_{42}) = \max(4{,}47;4)$	4,47
$d_{(1,4)3} = \max(d_{13}, d_{43}) = \max(4;4{,}47)$	4,47
$d_{(1,4)5} = \max(d_{15}, d_{45}) = \max(2{,}24;2{,}24)$	2,24
$d_{(1,4)6} = \max(d_{16}, d_{46}) = \max(2{,}83;4{,}47)$	4,47

Assim, a matriz resultante será:

$$\begin{array}{c} \\ (1,4) \\ 2 \\ 3 \\ 5 \\ 6 \end{array} \begin{bmatrix} 0 & 4{,}47 & 4{,}47 & 2{,}24 & 4{,}47 \\ 4{,}47 & 0 & 2 & 2{,}24 & 7{,}21 \\ 4{,}47 & 2 & 0 & 2{,}24 & 6{,}32 \\ 2{,}24 & 2{,}24 & 2{,}24 & 0 & 5 \\ 4{,}47 & 7{,}21 & 6{,}32 & 5 & 0 \end{bmatrix}$$

As demais iterações serão repetidas como descrito no algoritmo padrão, sempre calculando, para esse método, as distâncias máximas entre os elementos ou grupos.

A figura 4.9 traz a seqüência dos grupos formados em cada iteração do algoritmo.

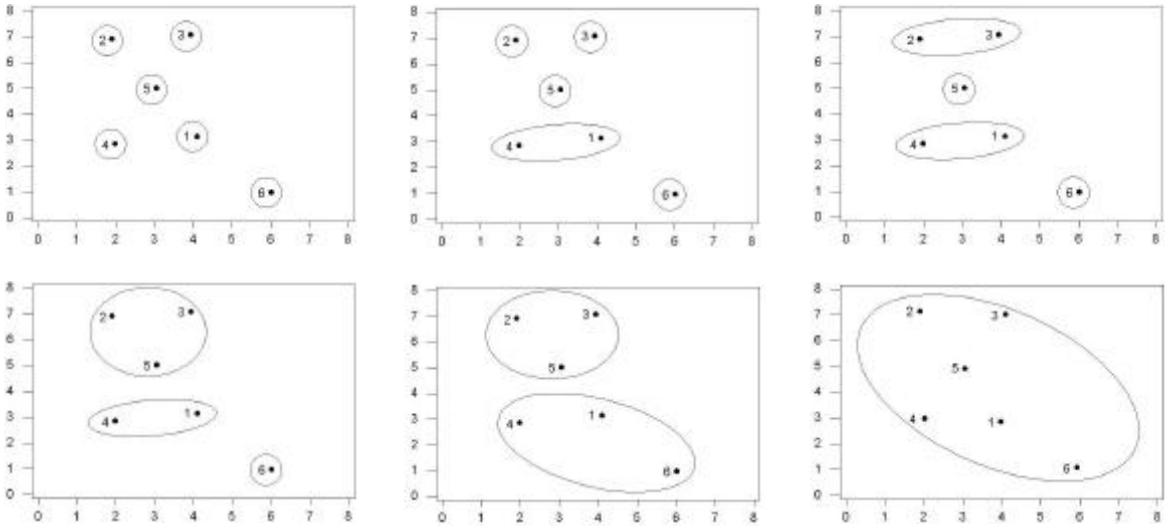

Figura 4.9: Seqüência de agrupamentos realizada no método de ligação por vizinho mais distante.

De acordo com a figura 4.9, pode-se verificar que na quarta iteração do algoritmo no método de ligação por vizinho mais distante, os agrupamentos são realizados de maneira diferente ao método de ligação por vizinho mais próximo.

A figura 4.10 traz o dendograma gerado pelo do método de ligação por vizinho mais distante.

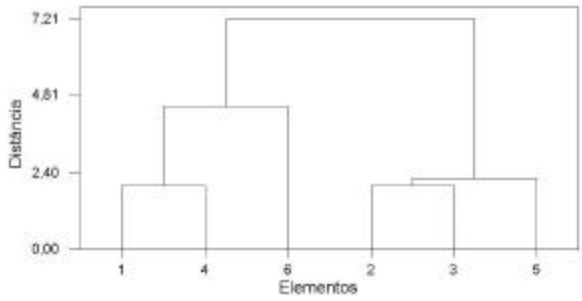

Figura 4.10: Dendograma aplicando o método de ligação por vizinho mais distante.

Algumas características desse método são (KAUFMANN, 1990):

- Apresenta bons resultados tanto para distâncias Euclidianas quanto para outras distâncias;
- Tendência a formar grupos compactos;
- Os ruídos demoram a serem incorporados ao grupo.

Os métodos de ligação por mais próximo e por vizinho mais distante trabalham em direções opostas. Se eles apresentam resultados semelhantes, significa que o grupo está bem definido no espaço, ou seja, o grupo é real. Mas se ocorre o contrário, os grupos provavelmente não existem (ROMESBURG, 1984).

4.2.1.3 Método *Average Linkage* ou ligação por média

Nesse método, a função distância é definida por:

$$d_{(UV)W} = \frac{(N_u.d_{UW} + N_v.d_{VW})}{N_u + N_v} \tag{4.7}$$

- onde: N_U e N_V são os números de elementos no grupo U e V, respectivamente;
d_{UW} e d_{VW} são as distâncias entre os elementos UW e VW, respectivamente.

Exemplo. Considerando-se a matriz do exemplo anterior, as distâncias médias são calculadas a seguir:

CÁLCULO DAS DISTÂNCIAS	VALOR
$d_{(1,4)2} = \frac{(1.4{,}47 + 1.4)}{1+1}$	4,24
$d_{(1,4)3} = \frac{(1.4 + 1.4{,}47)}{1+1}$	4,24
$d_{(1,4)5} = \frac{(1.2{,}24 + 1.2{,}24)}{1+1}$	2,24
$d_{(1,4)6} = \frac{(1.2{,}83 + 1.4{,}47)}{1+1}$	3,65

Assim, a matriz resultante será:

$$\begin{array}{c} (1,4) \\ 2 \\ 3 \\ 5 \\ 6 \end{array} \begin{bmatrix} 0 & 4{,}24 & 4{,}24 & 2{,}24 & 3{,}65 \\ 4{,}24 & 0 & 2 & 2{,}24 & 7{,}21 \\ 4{,}24 & 2 & 0 & 2{,}24 & 6{,}32 \\ 2{,}24 & 2{,}24 & 2{,}24 & 0 & 5 \\ 3{,}65 & 7{,}21 & 6{,}32 & 5 & 0 \end{bmatrix}$$

As demais iterações serão repetidas como descrito no algoritmo padrão, sempre calculando, para esse método, as distâncias médias entre os elementos ou grupos.

A figura 4.11 traz a seqüência dos grupos formados em cada iteração do algoritmo:

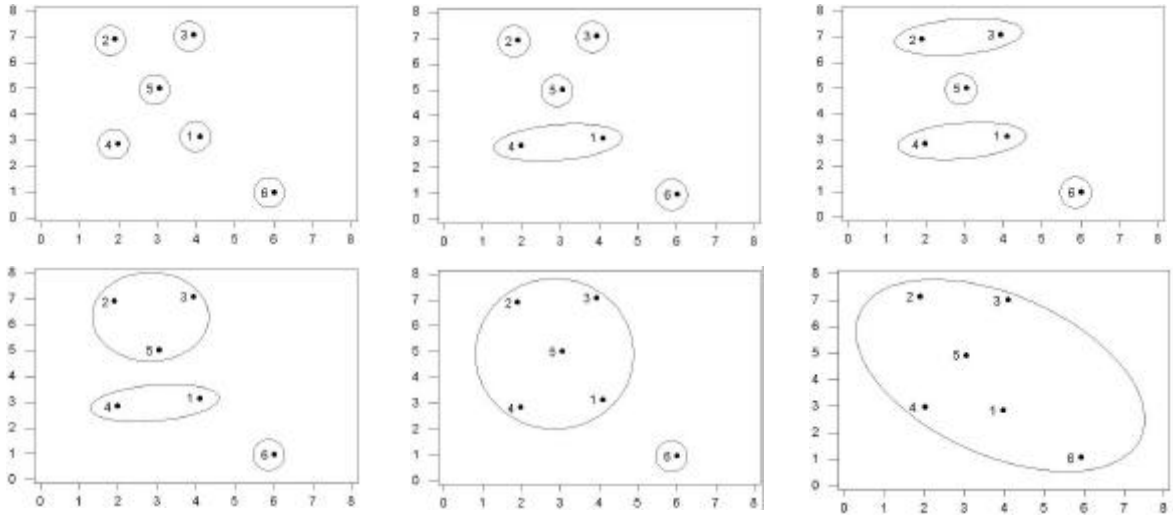

Figura 4.11: Seqüência de agrupamentos realizada no método de ligação por média.

De acordo com a figura 4.11, pode-se verificar que na quarta iteração do algoritmo no método de ligação por média, os agrupamentos são realizados de maneira diferente do método de ligação por vizinho mais distante e igual ao método de ligação por vizinho mais próximo.

A figura 4.12 traz o dendograma gerado pelo do método de ligação por média.

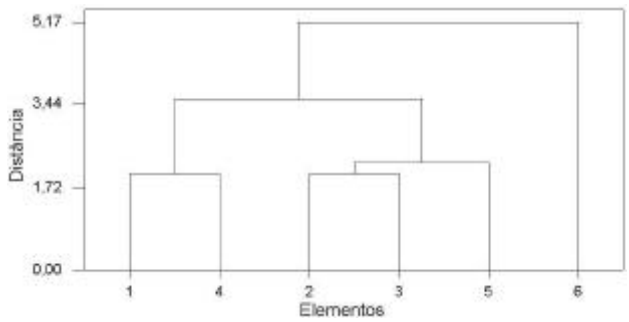

Figura 4.12: Dendograma aplicando o método de ligação por média.

Algumas características desse método são (KAUFMANN, 1990):

- Menor sensibilidade à ruídos que o os métodos de ligação por vizinho mais próximo e por vizinho mais distante;
- Apresenta bons resultados tanto para distâncias Euclidianas quanto para outras distâncias;
- Tendência a formar grupos com número de elementos similares.

4.2.1.4 Método *Centroid Linkage* ou ligação por centróide

Nesse método, a função distância é definida por:

$$d_{(UV)W} = \frac{N_U.d_{UW} + N_V.d_{VW}}{N_U + N_V} - \frac{N_U.N_V.d_{UV}}{(N_U + N_V)^2} \tag{4.8}$$

- onde: N_U e N_V são os números de elementos no grupo U e V, respectivamente;
d_{UW} e d_{VW} são as distâncias entre os elementos UW e VW, respectivamente.

Exemplo. Utilizando a matriz do exemplo anterior, as distâncias são calculadas a seguir:

CÁLCULO DAS DISTÂNCIAS	VALOR
$d_{(1,4)2} = \frac{1.4,47 + 1.4}{1+1} - \frac{1.1.1}{(1+1)^2}$	3,99
$d_{(1,4)3} = \frac{1.4 + 1.4,47}{1+1} - \frac{1.1.1}{(1+1)^2}$	3,99
$d_{(1,4)5} = \frac{1.2,24 + 1.2,24}{1+1} - \frac{1.1.1}{(1+1)^2}$	1,99
$d_{(1,4)6} = \frac{1.2,83 + 1.4,47}{1+1} - \frac{1.1.1}{(1+1)^2}$	3,4

Assim, a matriz resultante será:

$$\begin{array}{c} (1,4) \\ 2 \\ 3 \\ 5 \\ 6 \end{array} \begin{bmatrix} 0 & 3,99 & 3,99 & 1,99 & 3,4 \\ 3,99 & 0 & 2 & 2,24 & 7,21 \\ 3,99 & 2 & 0 & 2,24 & 6,32 \\ 1,99 & 2,24 & 2,24 & 0 & 5 \\ 3,4 & 7,21 & 6,32 & 5 & 0 \end{bmatrix}$$

As demais iterações serão repetidas como descrito no algoritmo padrão, sempre calculando, para esse método, as distâncias entre os centróides.

A figura 4.13 traz a seqüência dos grupos formados em cada iteração do algoritmo:

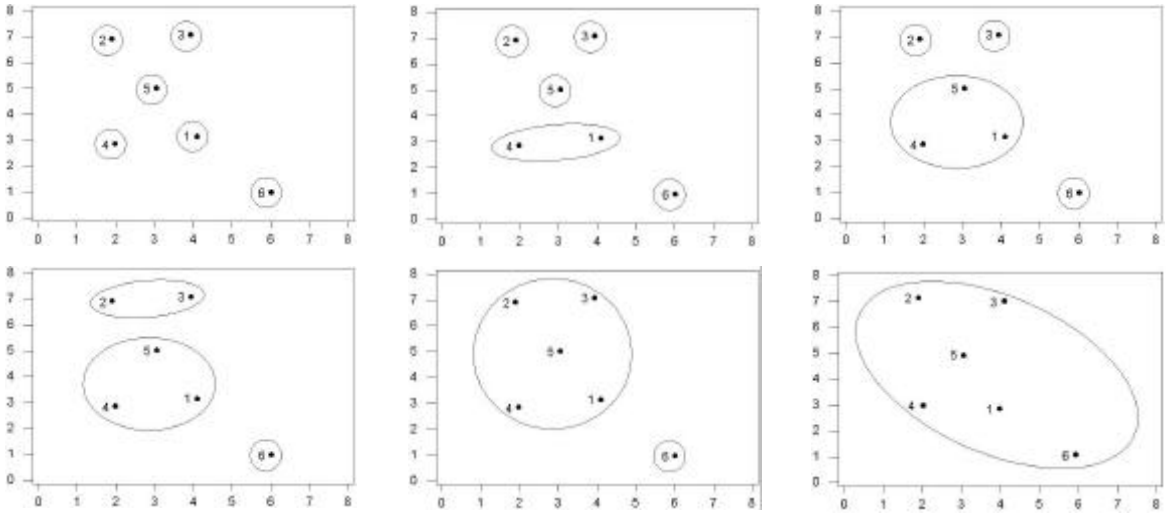

Figura 4.13: Seqüência de agrupamentos realizada no método de ligação por centróide.

A figura 4.14 mostra o dendograma gerado pelo do método de ligação por centróide:

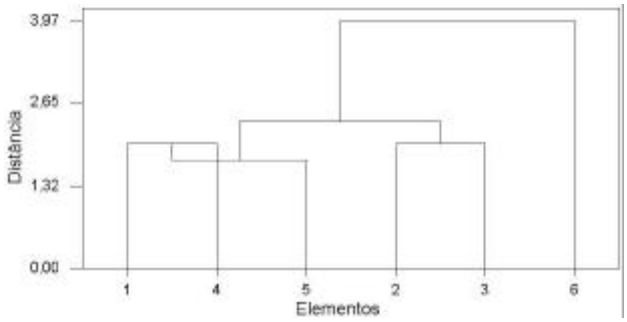

Figura 4.14: Dendograma aplicando o método de ligação por centróide.

Como características desse método, encontram-se:

- Robustez à presença de ruídos;
- Devido ao fenômeno da reversão, o método não é muito utilizado.

O fenômeno da reversão ocorre quando a distância entre centróides é menor que a distância entre grupos já formados, isso fará com que os novos grupos se formem ao um nível inferior aos grupos existentes, tornando o dendograma confuso.

O centróide é o ponto médio no espaço multidimensional e representa o centro de gravidade do respectivo grupo.

No exemplo, observa-se o fenômeno da reversão, pois na primeira ligação, entre os elementos 1 e 4, a distância entre centróides foi maior que na segunda ligação, entre o grupo (1,4) e o elemento 5. A figura 4.15 traz, passo a passo, a ligação entre os elementos.

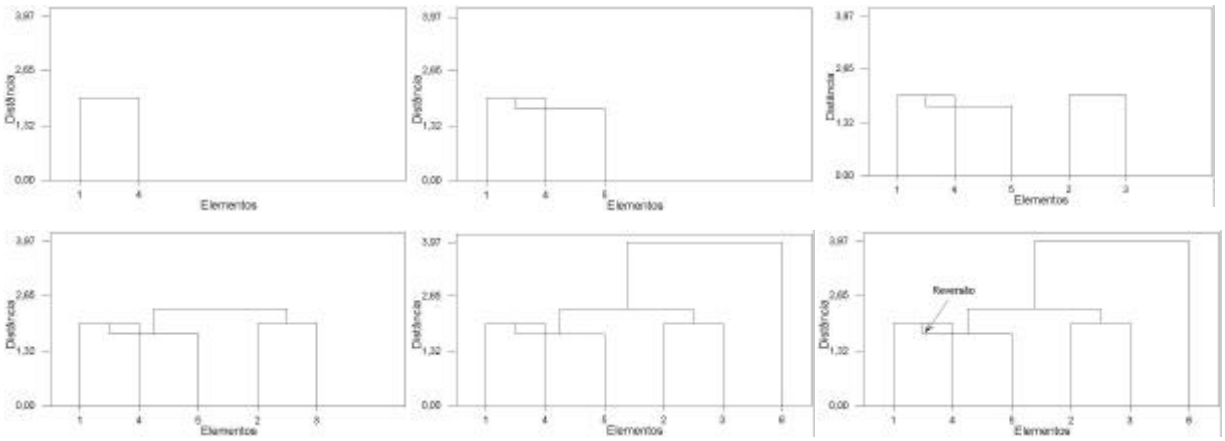

Figura 4.15: Etapas de ligação entre os elementos.

4.2.1.5 Método *Median Linkage* ou ligação por mediana

Nesse método, a função distância é dada por:

$$d_{(UV)W} = \frac{d_{UW} + d_{VW}}{2} - \frac{d_{UV}}{4} \tag{4.9}$$

- onde: d_{UW} e d_{VW} são as distâncias entre os elementos UW e VW, respectivamente.

Exemplo. Utilizando a matriz do exemplo anterior, as distâncias são calculadas a seguir:

CÁLCULO DAS DISTÂNCIAS	VALOR
$d_{(1,4)2} = \frac{4{,}47 + 4}{2} - \frac{1}{4}$	3,99
$d_{(1,4)3} = \frac{4 + 4{,}47}{2} - \frac{1}{4}$	3,99
$d_{(1,4)5} = \frac{2{,}4 + 2{,}4}{2} - \frac{1}{4}$	1,99
$d_{(1,4)6} = \frac{2{,}83 + 4{,}47}{2} - \frac{1}{4}$	3,4

Assim, a matriz resultante será:

$$\begin{array}{c} (1,4) \\ 2 \\ 3 \\ 5 \\ 6 \end{array} \begin{bmatrix} 0 & 3{,}99 & 3{,}99 & 1{,}99 & 3{,}4 \\ 3{,}99 & 0 & 2 & 2{,}24 & 7{,}21 \\ 3{,}99 & 2 & 0 & 2{,}24 & 6{,}32 \\ 1{,}99 & 2{,}24 & 2{,}24 & 0 & 5 \\ 3{,}4 & 7{,}21 & 6{,}32 & 5 & 0 \end{bmatrix}$$

As demais iterações serão repetidas como descrito no algoritmo padrão, sempre calculando, para esse método, as distâncias entre os elementos ou grupos de acordo com a equação 4.9.

A figura 4.16 traz a seqüência dos grupos formados em cada iteração do algoritmo.

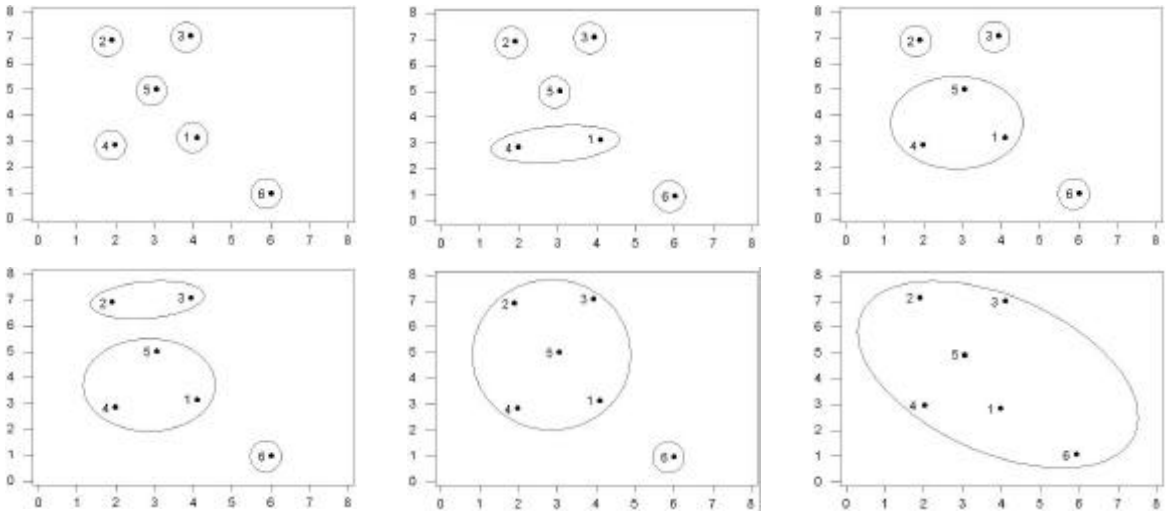

Figura 4.16: Seqüência de agrupamentos realizada no método de ligação por mediana.

A figura 4.17 mostra o dendograma gerado pelo do método de ligação por mediana:

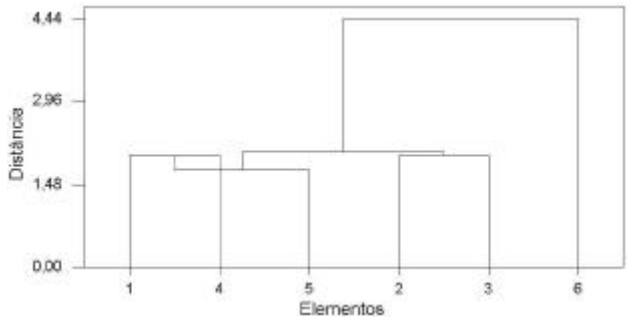

Figura 4.17: Dendograma aplicando o método de ligação por mediana.

Algumas características referentes a esse método são:

- Apresenta resultado satisfatório quando os grupos possuem tamanhos diferentes;
- Pode apresentar resultado diferente quando permutado os elementos na matriz de similaridade;
- Robustez à presença de *outliers*.

4.2.1.6 Método de ligação de *Ward*

Nesse método, a função distância é dada por:

$$d_{(UV)W} = \frac{((N_W + N_U).d_{UW} + (N_W + N_V).d_{VW} - N_W.d_{UV})}{N_W + N_U + N_V} \quad (4.10)$$

- onde: N_U e N_W são os números de elementos no grupo U e V, respectivamente;
d_{UW} e d_{VW} são as distâncias entre os elementos UW e VW, respectivamente.

Exemplo. Utilizando a mesma matriz do exemplo anterior, as distâncias são calculadas a seguir:

CÁLCULO DAS DISTÂNCIAS	VALOR
$d_{(1,4)2} = \frac{((1+1).4{,}47 + (1+1).4 - 1.1)}{3}$	5,31
$d_{(1,4)3} = \frac{((1+1).4 + (1+1).4{,}47 - 1.1)}{3}$	5,31
$d_{(1,4)5} = \frac{((1+1).2{,}24 + (1+1).2{,}24 - 1.1)}{3}$	2,65
$d_{(1,4)6} = \frac{((1+1).2{,}83 + (1+1).4{,}47 - 1.1)}{3}$	4,53

Assim, a matriz resultante será:

$$\begin{array}{c} (1,4) \\ 2 \\ 3 \\ 5 \\ 6 \end{array} \begin{bmatrix} 0 & 5{,}31 & 5{,}31 & 2{,}65 & 4{,}53 \\ 5{,}31 & 0 & 2 & 2{,}24 & 7{,}21 \\ 5{,}31 & 2 & 0 & 2{,}24 & 6{,}32 \\ 2{,}65 & 2{,}24 & 2{,}24 & 0 & 5 \\ 4{,}53 & 7{,}21 & 6{,}32 & 5 & 0 \end{bmatrix}$$

Os demais passos serão repetidos como descrito no algoritmo padrão, sempre calculando, para esse método, as distâncias entre os elementos ou grupos de acordo com a equação 4.10.

A figura 4.18 traz a seqüência dos grupos formados em cada iteração do algoritmo.

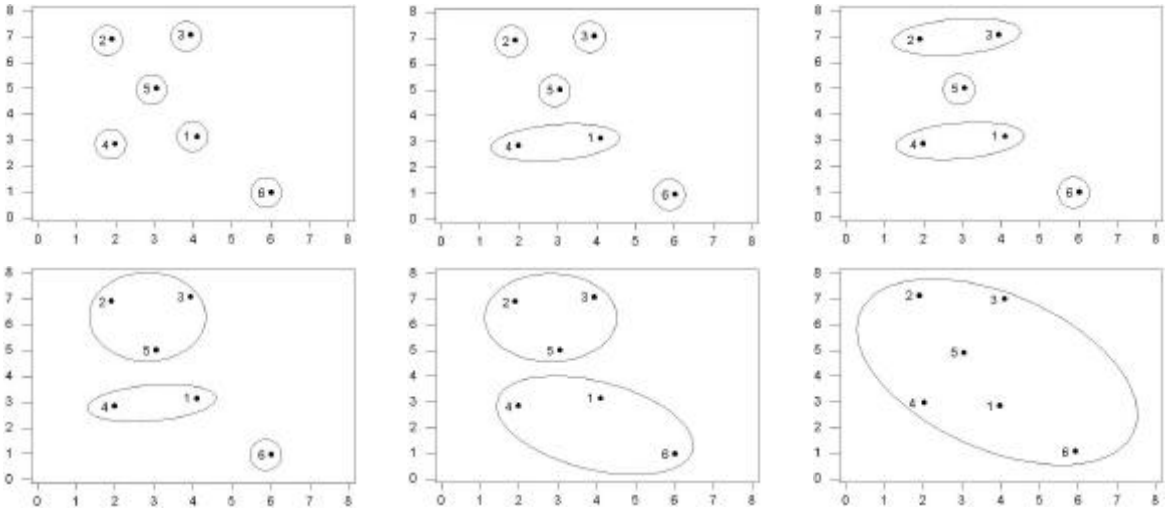

Figura 4.18: Seqüência de agrupamentos realizada no método de ligação de *Ward*.

A figura 4.19 mostra o dendograma gerado pelo do método de ligação de *Ward.*

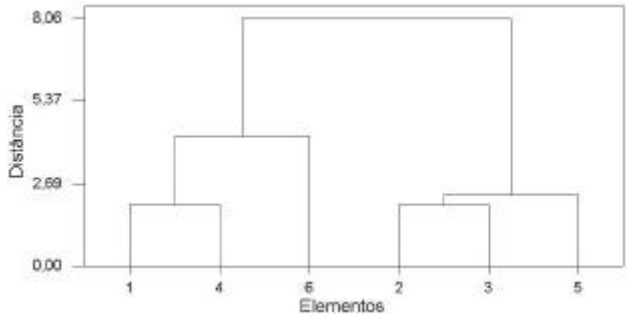

Figura 4.19: Dendograma aplicando o método de ligação de *Ward*.

Algumas características desse método são:

- Apresenta bons resultados tanto para distâncias euclidianas quanto para outras distâncias;
- Pode apresentar resultados insatisfatórios quando o número de elementos em cada grupo é praticamente igual;
- Tem tendência a combinar grupos com poucos de elementos;
- Sensível à presença de *outliers.*

A tabela 4.2 traz um resumo dos métodos aglomerativos.

MÉTODO	DISTÂNCIA	CARACTERÍSTICAS
Ligação por vizinho mais próximo	$d_{(UV)W} = \min(d_{UW}, d_{VW})$	Sensibilidade à ruídos. Encadeamento.
Ligação por vizinho mais distante	$d_{(UV)W} = \max(d_{UW}, d_{VW})$	Tendência a formar grupos compactos.
Ligação por média	$d_{(UV)W} = \frac{(N_u.d_{UW} + N_v.d_{VW})}{N_u + N_v}$	Tendência a formar grupos com número de elementos similares.
Ligação por centróide	$d_{(UV)W} = \frac{N_U.d_{UW} + N_V.d_{VW}}{N_U + N_V} - \frac{N_U.N_V.d_{UV}}{(N_U + N_V)^2}$	Robustez à ruídos. Reversão.
Ligação por mediana	$d_{(UV)W} = \frac{d_{UW} + d_{VW}}{2} - \frac{d_{UV}}{4}$	Robustez à ruídos.
***Ligação de* Ward**	$d_{(UV)W} = \frac{((N_W + N_U).d_{UW} + (N_W + N_V).d_{VW} - N_W.d_{UV})}{N_W + N_U + N_V}$	Sensibilidade à ruídos.

Tabela 4.2: Resumo dos métodos hierárquicos aglomerativos.

4.2.2 Métodos Divisivos

Os métodos divisivos trabalham na direção oposta dos métodos aglomerativos, ou seja, um grupo inicial contendo todos os elementos é dividido em dois subgrupos, de tal forma que os elementos em um subgrupo estejam distantes dos elementos do outro subgrupo.

Esses subgrupos são então divididos em subgrupos dissimilares e o processo continua até cada elemento formar um grupo (figura 4.20).

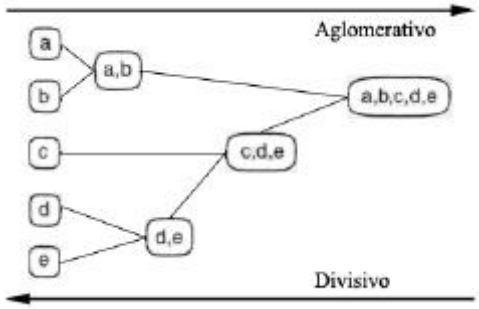

Figura 4.20: Relação entre o método aglomerativo e divisivo.

Os métodos divisivos são pouco mencionados na literatura, pois exigem uma maior capacidade computacional que os métodos aglomerativos (KAUFMAN, 1990).

Comparando apenas o primeiro passo dos métodos aglomerativos e divisivos, notamos que o método divisivo exige um maior número de iterações. No primeiro passo do método aglomerativo são consideradas todas as uniões possíveis de dois elementos, tendo a complexidade de tempo da ordem $O(N^2)$, sendo N o número de elementos. Este número cresce quadraticamente à medida que N aumenta, ou seja, seu crescimento é acelerado. Mesmo assim a implementação do algoritmo é computacionalmente viável.

Já o método divisivo, baseado no mesmo princípio, começará considerando todas as divisões dos elementos em dois grupos, com pelo menos um elemento em cada grupo, tendo a complexidade de tempo da ordem de $O(2^N)$. Este número cresce exponencialmente à medida que N aumenta, dessa forma, para um grande número de elementos, torna-se inviável sua implementação computacional.

Exemplo. Considerando-se 15 elementos no primeiro passo dos métodos aglomerativo e divisivo, observa-se uma grande diferença no número de possibilidades de agrupamento:

Método aglomerativo $\binom{N}{2} = \frac{N(N-1)}{2} = \frac{15(15-1)}{2} = 105$ possibilidades

Método divisivo - $2^{N-1} - 1 = 2^{15-1} - 1 = 16.383$ possibilidades

É possível, entretanto, construir métodos divisivos que não considerem o conjunto completo de divisões possíveis (MACNAUGHTON-SMITH, 1964). Um desses métodos é apresentado a seguir.

4.2.2.1 Método de *MACNAUGHTON-SMITH*

O algoritmo de *MacNaughton-Smith* consegue contornar esse problema do crescimento exponencial de iterações, exigindo, no pior caso, duas vezes mais iterações que os métodos aglomerativos.

A figura 4.21 traz a descrição desse algoritmo.

```
Entrada: A base de dados com N elementos.
Saída: Um conjunto de grupos.
1. j=1;
2. Repetir;
3.     Escolher o grupo Gj com maior número de elementos Nj;
4.     Iniciar uma matriz Dnj´nj;
5.     Calcular a similaridade média Sm de cada elemento do grupo Gj em
       relação aos demais;
6.           Fazer enquanto Sm > 0;
7.                 Retirar o elemento e com maior Sm do grupo Gj;
8.                 Armazenar o elemento e no grupo Fj;
9.                 (re)Calcular a similaridade média Si entre os
                   elementos que restaram no grupo Gj;
10.                (re)Calcular a similaridade média Sa entre cada
                   elemento do grupo Gj e o grupo Fj;
11.                Sm = Si - Sa;
12.          Fim;
13.    j=j+1;
14. Até restarem apenas grupos com dois elementos;
15. Repetir;
16.    Escolher o grupo H com maior similaridade média;
17.    Dividir o grupo H;
18. Até que todos grupos sejam divididos;
```

Figura 4.21: Algoritmo de *MacNaughton-Smith.*

Exemplo. Dada a matriz *D* abaixo e aplicando o algoritmo *MacNaughton-Smith*, temos:

$$D = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{bmatrix} 0 & 4,47 & 4 & 2 & 2,24 & 2,83 \\ 4,47 & 0 & 2 & 4 & 2,24 & 7,21 \\ 4 & 2 & 0 & 4,47 & 2,24 & 6,32 \\ 2 & 4 & 4,47 & 0 & 2,24 & 4,47 \\ 2,24 & 2,24 & 2,24 & 2,24 & 0 & 5 \\ 2,83 & 7,21 & 6,32 & 4,47 & 5 & 0 \end{bmatrix}$$

Calculando a similaridade média, obtém-se:

ELEMENTO	SIMILARIDADE MÉDIA RELATIVA AOS DEMAIS ELEMENTOS
1	$\frac{4,47 + 4 + 2 + 2,24 + 2,83}{5} = 3,11$
2	$\frac{4,47 + 2 + 4 + 2,24 + 7,21}{5} = 3,98$
3	$\frac{4 + 2 + 4,47 + 2,24 + 6,32}{5} = 3,81$
4	$\frac{2 + 4 + 4,47 + 2,24 + 4,47}{5} = 3,44$
5	$\frac{2,24 + 2,24 + 2,24 + 2,24 + 5}{5} = 2,79$
6	$\frac{2,83 + 7,21 + 6,32 + 4,47 + 5}{5} = 5,17$

O elemento 6 possui a maior similaridade, sendo então, retirado do grupo (1,2,3,4,5,6).

Calculando a similaridade média para os elementos que restaram no grupo, a similaridade média entre cada elemento do grupo e o elemento retirado e a diferenças desses valores, obtemos:

ELEMENTO	SIMILARIDADE MÉDIA RELATIVA AOS DEMAIS ELEMENTOS	SIMILARIDADE MÉDIA RELATIVA AO ELEMENTO RETIRADO (6)	DIFERENÇA
1	$\frac{4,47 + 4 + 2 + 2,24}{4} = 3,18$	$d_{1,6} = 2,83$	0,35
2	$\frac{4,47 + 2 + 4 + 2,24}{4} = 3,18$	$d_{2,6} = 7,21$	-4,03
3	$\frac{4 + 2 + 4,47 + 2,24}{4} = 3,18$	$d_{3,6} = 6,32$	-3,14
4	$\frac{2 + 4 + 4,47 + 2,24}{4} = 3,18$	$d_{4,6} = 4,47$	-1,29
5	$\frac{2,24 + 2,24 + 2,24 + 2,24}{4} = 2,24$	$d_{5,6} = 5$	-2,76

Na tabela acima, verifica-se que o elemento 1 possui a maior diferença positiva, sendo então retirado do grupo (1,2,3,4,5) e agrupado ao elemento (6).

Recalculando a similaridade média entre os elementos do grupo (2,3,4,5), a similaridade dos elementos desse grupo em relação ao grupo (1,6) e a diferença desses valores, obtemos:

ELEMENTO	SIMILARIDADE MÉDIA RELATIVA AOS DEMAIS ELEMENTOS	SIMILARIDADE MÉDIA RELATIVA AOS ELEMENTOS DO NOVO GRUPO (1,6)	DIFERENÇA
2	$\frac{2 + 4 + 2,24}{3} = 2,75$	$\frac{4,47 + 2,83}{2} = 3,65$	-0,90
3	$\frac{2 + 4,47 + 2,24}{3} = 2,90$	$\frac{4 + 6,32}{2} = 5,16$	-2,26
4	$\frac{4 + 4,47 + 2,24}{3} = 3,57$	$\frac{2 + 4,47}{2} = 3,24$	0,33
5	$\frac{2,24 + 2,24 + 2,24}{3} = 2,24$	$\frac{2,24 + 5}{2} = 3,62$	-1,38

Na tabela acima, verifica-se que o elemento 4 possui a maior diferença positiva, sendo então, retirado do grupo (2,3,4,5) e agrupado ao grupo (1,6).

Recalculando a similaridade média entre os elementos do grupo (2,3,5), a similaridade dos elementos desse grupo em relação ao grupo (1,4,6) e a diferença desses valores, obtemos:

ELEMENTO	SIMILARIDADE MÉDIA RELATIVA AOS DEMAIS ELEMENTOS	SIMILARIDADE MÉDIA RELATIVA AOS ELEMENTOS DO NOVO GRUPO (1,4,6)	DIFERENÇA
2	$\frac{2 + 2,24}{2} = 2,12$	$\frac{4,47 + 4 + 7,21}{3} = 5,21$	-3,10
3	$\frac{2 + 2,24}{2} = 2,12$	$\frac{4 + 4,47 + 6,32}{3} = 4,93$	-2,81
5	$\frac{2,24 + 2,24}{2} = 2,24$	$\frac{2,24 + 2,24 + 5}{3} = 3,16$	-0,92

Na tabela acima, verifica-se que todas diferenças são negativas, portanto, não há elementos a serem retirados do grupo (2,3,5).

Os dois grupos formados possuem três elementos, portanto, será escolhido o grupo (2,3,5) para prosseguir com as divisões.

Assim, temos a seguinte matriz de similaridade:

$$D = \begin{matrix} 2 \\ 3 \\ 5 \end{matrix}\begin{bmatrix} 0 & 2 & 2{,}24 \\ 2 & 0 & 2{,}24 \\ 2{,}24 & 2{,}24 & 0 \end{bmatrix}$$

Calculando a similaridade média de cada elemento em relação aos demais, obtém-se:

ELEMENTO	SIMILARIDADE MÉDIA EM RELATIVA AOS DEMAIS ELEMENTOS
2	$\frac{2 + 2{,}24}{2} = 2{,}12$
3	$\frac{2 + 2{,}24}{2} = 2{,}12$
5	$\frac{2{,}24 + 2{,}24}{2} = 2{,}24$

Na tabela acima, verifica-se que o elemento 5 possui a maior similaridade média, assim, ele será retirado do grupo (2,3,5).

Calculando a similaridade média entre os elementos do grupo (2,3), a similaridade dos elementos desse grupo em relação ao elemento (5) e a diferença desses valores, obtemos:

ELEMENTO	SIMILARIDADE MÉDIA RELATIVA AOS DEMAIS ELEMENTOS	SIMILARIDADE MÉDIA RELATIVA AO ELEMENTO RETIRADO (5)	DIFERENÇA
2	$d_{23} = 2$	$d_{25} = 2{,}24$	-0,24
3	$d_{32} = 2$	$d_{35} = 2{,}24$	-0,24

Como as diferenças são negativas, não é necessário agrupar elementos ao elemento 5.

Considerando o grupo (1,4,6), temos a seguinte matriz de similaridade:

$$D = \begin{matrix} 1 \\ 4 \\ 6 \end{matrix}\begin{bmatrix} 0 & 2 & 2{,}83 \\ 2 & 0 & 4{,}47 \\ 2{,}83 & 4{,}47 & 0 \end{bmatrix}$$

Calculando a similaridade média de cada elemento em relação aos demais, obtemos:

ELEMENTO	SIMILARIDADE MÉDIA RELATIVA AOS DEMAIS ELEMENTOS
1	$\frac{2 + 2,83}{2} = 2,42$
4	$\frac{2 + 4,47}{2} = 3,24$
6	$\frac{2,83 + 4,47}{2} = 3,65$

Como a similaridade média do elemento 6 é a maior, ele será retirado do grupo (1,4,6).

Calculando a similaridade média entre os elementos do grupo (1,4), a similaridade dos elementos desse grupo em relação ao elemento (6) e a diferença desses valores, obtemos:

ELEMENTO	SIMILARIDADE MÉDIA RELATIVA AOS DEMAIS ELEMENTOS	SIMILARIDADE MÉDIA RELATIVA AO ELEMENTO RETIRADO (6)	DIFERENÇA
1	$d_{14} = 2$	$d_{16} = 2,83$	-0,83
4	$d_{41} = 2$	$d_{46} = 4,47$	-2,47

Como as diferenças são negativas, não é necessário agrupar elementos ao elemento 6.

Restam grupos com apenas um ou dois elementos. Portanto, serão divididos os grupo (1,4) e (2,3).

Como d_{23} e d_{14} são iguais, o primeiro grupo, ou seja, (2,3), será dividido primeiro.

Como foi dito anteriormente, o método divisivo de *MacNaughton-Smith* realizou um maior número de iterações para a formação dos agrupamentos que os métodos aglomerativos.

A figura 4.22 traz a seqüência dos grupos formados em cada iteração do algoritmo.

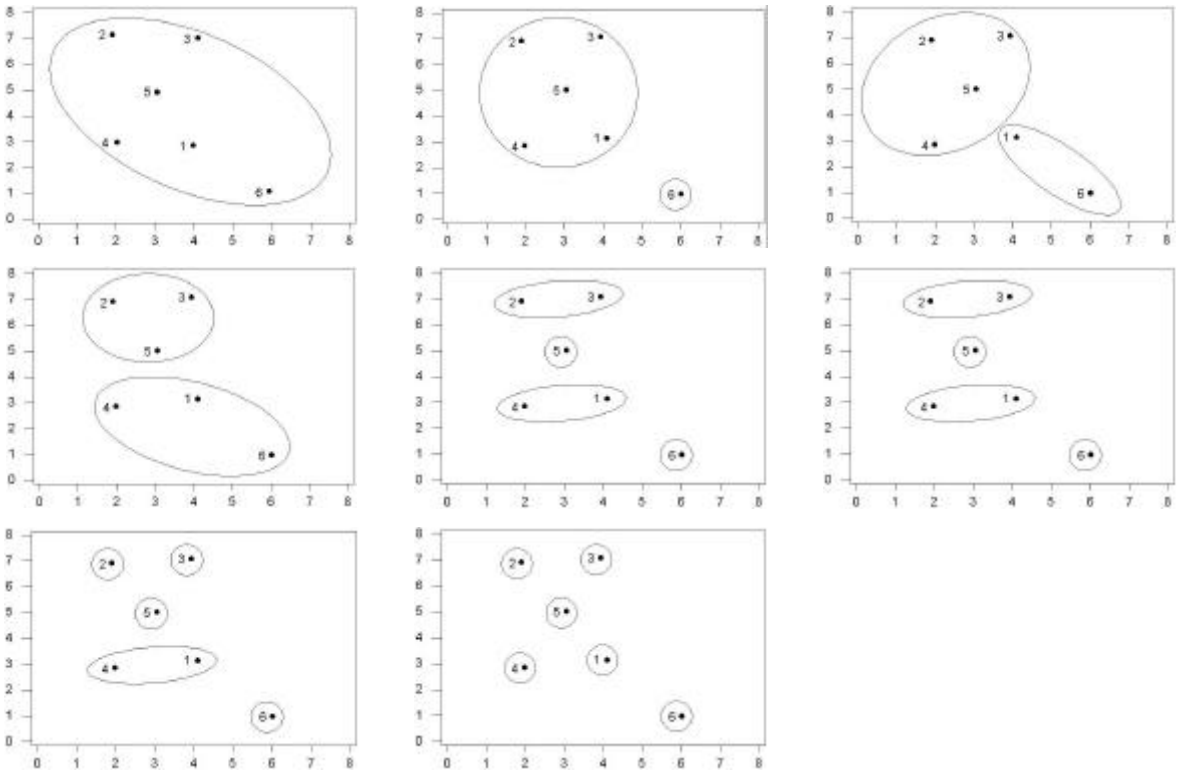

Figura 4.22: Seqüência de agrupamentos realizada no método *MacNaughton-Smith.*

A figura 4.23 mostra o dendograma gerado pelo do método de *MacNaughton-Smith*.

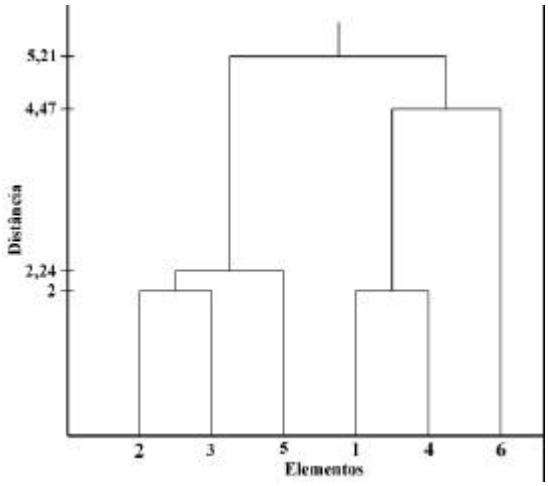

Figura 4.23: Dendograma aplicando o método de *MacNaughton-Smith*.

O algoritmo de *MacNaughton-Smith*, ao utilizar todas variáveis simultaneamente (matriz de similaridade), é considerado um método politético. Existem também os métodos divisivos que fazem cada divisão de acordo com uma única variável, esses são chamados de

monotéticos. O método politético possui a vantagem de ter implementação mais fácil que o monotético por utilizar a matriz de similaridade e, além disso, apresentam melhores resultados.

Comparando os métodos aglomerativos com os divisivos, verifica-se que o método divisivo possui vantagem ao considerar no primeiro estágio muitas divisões, diminuindo a probabilidade de uma decisão errada. Portanto, esse método torna-se mais seguro que o aglomerativo (KAUFMAN, 1990).

4.3 MÉTODOS NÃO-HIERÁRQUICOS OU POR PARTICIONAMENTO

Os métodos não-hierárquicos, ou por particionamento, foram desenvolvidos para agrupar elementos em *K* grupos, onde *K* é a quantidade de grupos definida previamente.

Nem todos valores de *K* apresentam grupos satisfatórios, sendo assim, aplica-se o método várias vezes para diferentes valores de *K*, escolhendo os resultados que apresentem melhor interpretação dos grupos ou uma melhor representação gráfica (BUSSAB, 1990).

A idéia central da maioria dos métodos por particionamento é escolher uma partição inicial dos elementos e, em seguida, alterar os membros dos grupos para obter-se a melhor partição (ANDERBERG, 1973).

Quando comparado com o método hierárquico, o método por particionamento é mais rápido porque não é necessário calcular e armazenar, durante o processamento, a matriz de similaridade.

Em geral, os métodos por particionamento diferem entre si pela maneira que constituem a melhor partição. Os métodos por particionamento mais conhecidos são o método *k-means* (k-médias) e o método *k-medoid* (k-medóides), e são descritos a seguir.

4.3.1 Método *k-means*

O método *k-means* toma um parâmetro de entrada, *K*, e particiona um conjunto de *N* elementos em *K* grupos, conforme figura 4.24:

```
Entrada: O número de grupos, K, e a base de dados com N
elementos.
Saída: Um conjunto de K grupos.
1. Escolher arbitrariamente K elementos da base de dados
como os centros iniciais dos grupos;
2. Repetir;
3.     (re)Atribua cada elemento ao grupo ao qual o
       elemento é mais similar, de acordo com o valor
       médio dos elementos no grupo;
4.     Atualizar as médias dos grupos, calculando o valor
       médio dos elementos para cada grupo;
5. Até que não haja mudanças de elementos de um grupo
   para outro.
```

Figura 4.24: Algoritmo do *k-means*.

Esse método possui uma complexidade de tempo da ordem de $O(nkl)$ e uma complexidade de espaço é da ordem de $O(k+n)$, onde *n* é o número de elementos, *k* é o número de grupos e *l* é o número de iterações do algoritmo (JAIN et al, 1999).

Exemplo. Considerando os elementos e variáveis da tabela 4.3, assumindo *K*=2, tais como (1,2,3) e (4,5,6), obtemos:

ELEMENTO	VARIÁVEIS	
	X	Y
1	4	3
2	2	7
3	4	7
4	2	3
5	3	5
6	6	1

Tabela 4.3: Conjunto de dados exemplo.

Calculando a média dos grupos (1,23) e (4,5,6), temos:

GRUPO	VARIÁVEIS	
	X	Y
(1,2,3)	$\frac{4+2+4}{3} = 3{,}33$	$\frac{3+7+7}{3} = 5{,}67$
(4,5,6)	$\frac{2+3+6}{3} = 3{,}67$	$\frac{3+5+1}{3} = 3$

Calculando a distância Euclidiana de cada objeto ao centróide dos grupos, obtém-se:

DISTÂNCIA DE CADA ELEMEMTO AO CENTRÓIDE DOS GRUPOS	VALOR
$de_{(1(1,2,3))} = \sqrt{(4-4)^2 + (3-3)^2}$	0
$de_{(1(4,5,6))} = \sqrt{(4-2)^2 + (3-7)^2}$	4,47
$de_{(2(1,2,3))} = \sqrt{(2-4)^2 + (7-3)^2}$	4,47
$de_{(2(4,5,6))} = \sqrt{(2-2)^2 + (7-7)^2}$	0
$de_{(3(1,2,3))} = \sqrt{(4-4)^2 + (7-3)^2}$	4
$de_{(3(4,5,6))} = \sqrt{(4-2)^2 + (7-7)^2}$	2

Na tabela acima, verifica-se que os elementos 2 e 3 estão mais próximos do grupo (4,5,6). Assim, eles serão retirado do grupo (1,2,3) e associados ao grupo (4,5,6).

Recalculando o centróide dos grupos, temos:

GRUPO	COORDENADAS	
	X	Y
1	4	3
(2,3,4,5,6)	$\frac{(2+4+2+3+6)}{5} = 3,4$	$\frac{(7+7+3+5+1)}{5} = 4,6$

Calculando a distância Euclidiana de cada objeto ao centróide dos grupos, temos:

DISTÂNCIA DE CADA ELEMEMTO AO CENTRÓIDE DOS GRUPOS	RESULTADO
$de_{(4(1))} = \sqrt{(2-4)^2 + (3-3)^2}$	2
$de_{(4(2,3,4,5,6))} = \sqrt{(2-3,4)^2 + (3-4,6)^2}$	2,17
$de_{(5(1))} = \sqrt{(3-4)^2 + (5-3)^2}$	2,24
$de_{(5(2,3,4,5,6))} = \sqrt{(3-3,4)^2 + (5-4,6)^2}$	0,57
$de_{(6(1))} = \sqrt{(6-4)^2 + (1-3)^2}$	2,83
$de_{(6(2,3,4,5,6))} = \sqrt{(6-3,4)^2 + (1-4,6)^2}$	4,44

Na tabela acima, verifica-se que os elementos 4 e 6 estão mais próximos do grupo (1). Assim, eles serão retirados do grupo (2,3,4,5,6) e associados ao grupo (1).

Recalculando o centróide dos grupos, obtém-se:

GRUPO	COORDENADAS	
	$\bar{x}$	$\bar{Y}$
(1,4,6)	$\frac{(4+2+6)}{3} = 4$	$\frac{(3+3+1)}{3} = 2{,}33$
(2,3,,5)	$\frac{(2+4+3)}{3} = 3$	$\frac{(7+7+5)}{3} = 6{,}33$

Calculando a distância Euclidiana de cada elemento ao centróide dos grupos, temos:

DISTÂNCIA DE CADA ELEMEMTO AO CENTRÓIDE DOS GRUPOS	RESULTADO
$de_{(1(1,4,6))} = \sqrt{(4-4)^2 + (3-2{,}33)^2}$	0,45
$de_{(1(2,3,5))} = \sqrt{(4-3)^2 + (3-6{,}33)^2}$	12,09
$de_{(2(1,4,6))} = \sqrt{(2-4)^2 + (7-2{,}33)^2}$	25,81
$de_{(2(2,3,5))} = \sqrt{(2-3)^2 + (7-6{,}33)^2}$	1,45
$de_{(3(1,4,6))} = \sqrt{(4-4)^2 + (7-2{,}33)^2}$	21,81
$de_{(3(2,3,5))} = \sqrt{(4-3)^2 + (7-6{,}33)^2}$	1,45
$de_{(4(1,4,6))} = \sqrt{(2-4)^2 + (3-2{,}33)^2}$	4,45
$de_{(4(2,3,5))} = \sqrt{(2-3)^2 + (3-6{,}33)^2}$	12,09
$de_{(5(1,4,6))} = \sqrt{(3-4)^2 + (5-2{,}33)^2}$	8,13
$de_{(5(2,3,5))} = \sqrt{(3-3)^2 + (5-6{,}33)^2}$	1,77
$de_{(6(1,4,6))} = \sqrt{(6-4)^2 + (1-2{,}33)^2}$	5,77
$de_{(6(2,3,5))} = \sqrt{(6-3)^2 + (1-6{,}33)^2}$	37,41

Verificando a necessidade de realocação, observa-se que cada elemento está corretamente associado ao grupo com o centróide mais próximo, ou seja, cada elemento possui menor distância em relação ao grupo no qual faz parte do que em relação ao outro grupo. Sendo assim, o processo encerra-se com os grupos (1,4,6) e (2,3,5) conforme figura 4.25.

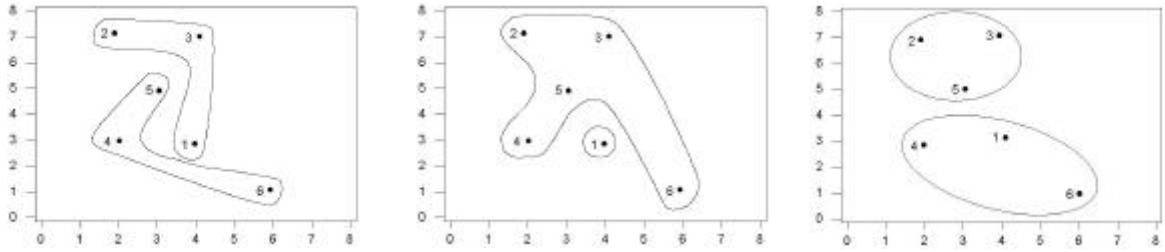

Figura 4.25: Seqüência de agrupamentos realizada no método *k-means*.

Algumas características desse método são:

- Sensibilidade a ruídos, uma vez que um elemento com um valor extremamente alto pode distorcer a distribuição dos dados;
- Tendência a formar grupos esféricos;
- O número de grupos é o mesmo durante todo o processo;
- Inadequado para descobrir grupos com formas não convexas ou de tamanhos muito diferentes.

4.3.2 Método *k-medoid*

O método *k-medoid* utiliza o valor médio dos elementos em um grupo como um ponto referência, chamado de medóide. Esse é o elemento mais centralmente localizado em um grupo.

A estratégia básica é encontrar *K* grupos em *N* elementos e, arbitrariamente, encontrar um elemento representativo (medóide) para cada grupo. Cada elemento remanescente é agrupado com o medóide ao qual ele é mais similar. A estratégia, então, iterativamente, troca um dos medóides por um dos não medóides enquanto a qualidade do agrupamento resultante é melhorada.

A figura 4.26 traz o algoritmo *k-medoid*:

Entrada: O número de grupos, ***K***, e a base de dados com ***N*** elementos.
Saída: Um conjunto de ***K*** grupos.
1. Escolher, arbitrariamente, ***K*** elementos da base de dados como os medóides iniciais dos grupos;
2. Repetir;
3. atribua cada elemento remanescente ao grupo com o medóide mais próximo;
4. aleatoriamente, selecione um elemento que não esteja como medóide, ***r***;
5. calcule o custo total, S, de trocar o medóide O_j pelo elemento ***r***;
6. se $S < 0$ então troque O_j por ***r*** para formar o novo conjunto de k-medóides;
7. Até que não haja mudança de objetos de um grupo para outro.

Figura 4.26: Algoritmo *k-medoid*.

Exemplo. Considerando os mesmos elementos do exemplo anterior, assumindo K=2 e escolhendo, aleatoriamente, os elementos 1 e 4 como medóides iniciais, temos:

ELEMENTOS (i)	d_{i1}	d_{i4}	$Min(d_{i1},d_{i4})$	MEDÓIDE MAIS PRÓXIMO
1	0	2	0	1
2	4,47	4	4	4
3	4	4,47	4	1
4	2	0	0	4
5	2,24	2,24	2,24	1
6	2,83	4,47	2,83	1
			$Média_{1,4}$ = 2,18	

Com base na tabela acima, verificamos que os elementos 2, 5 e 6 são agrupados ao elemento 1, pois estão mais próximos desse medóide. O único elemento mais próximo ao medóide 4 é o elemento 3, portanto será agrupado a esse.

A média das similaridades mínimas, calculada acima, representa a qualidade dos grupos encontrados. Quanto menor esse valor, melhor é a qualidade dos grupos. Essa média é utilizada para encontrar o custo, S, na mudança de medóide.

Para verificar a necessidade de mudança de medóide, selecionamos aleatoriamente o elemento 6 e calculamos o custo de trocar o medóide 1 por 6.

ELEMENTOS (i)	d_{i6}	d_{i4}	$min(d_{i6},d_{i4})$	MEDÓIDE MAIS PRÓXIMO
1	2,83	2	2	4
2	7,21	4	4	4
3	6,32	4,47	4,47	4
4	4,47	0	0	4
5	5	2,24	2,24	4
6	0	4,47	0	6
			$Média_{6,4} = 2,12$	

Na tabela acima, verificamos que os elementos 1, 2, 3 e 5 são agrupados ao elemento 4, pois estão mais próximos desse medóide. Nenhum elemento é agrupado ao medóide 6.

Calculando o custo de troca do medóide 1 pelo 6, temos:

$S_{1,6} = Média_{6,4} - Média_{1,4} = 2,12-2,18 = - 0,06$

Como o custo é menor que zero, o medóide 1 é substituído pelo medóide 6.

O algoritmo prossegue selecionando novos não-medóides verificando a necessidade de substituir os medóides. Essa análise é feita para todos os pares de elementos.

Na tabela abaixo, temos um resumo dos resultados:

MEDÓIDES	MÉDIAS
(1-4)	2,18
(4-6)	2,12
(2-6)	1,85
(1-2)	1,51
(1-5)	1,55
(1-6)	2,12
(1-3)	1,51
(2-3)	2,76
(2-4)	1,79
(2-5)	1,91
(3-5)	1,91
(3-4)	1,79
(4-5)	1,83
(4-6)	2,12
(3-6)	1,92
(5-6)	1,49

Verificando a tabela acima, os medóides 5 e 6 possuem a menor média e, portanto, são os medóides finais e serão utilizados para formar os grupos.

Os elementos 1, 2, 3 e 4 são agrupados ao medóide 5. Nenhum elemento é agrupado ao medóide 6.

Sendo assim, o processo encerra-se com os grupos (1,2,3,4,5) e (6), conforme figura 4.27.

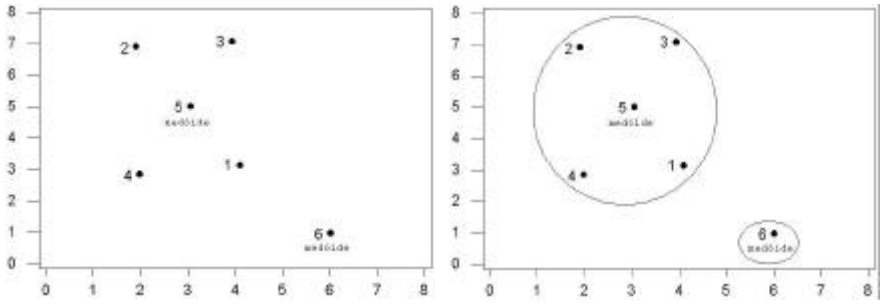

Figura 4.27: Agrupamentos realizados no método *k-medoid*.

Algumas características desse método são:

- Independente da ordem, os resultados serão os mesmos;
- Tendência a encontrar grupos esféricos;
- Processamento mais custoso que o *k-means*;
- Não aplicável à grandes bases de dados, pois o custo de processamento é alto;
- Mais robusto do que o *k-means* na presença de ruídos porque o medóide é menos influenciado pelos ruídos do que a média.

Uma forma de otimizar o método *k-medoid* para grandes bases de dados é considerar uma porção dos dados como uma amostra representativa, e escolher os medóides dessa amostra. Se a amostra é selecionada aleatoriamente, ela deverá representar bem o conjunto de dados originais, apresentando bons resultados (KAUFMAN, 1990).

4.4 OUTROS MÉTODOS

Além das técnicas estatísticas de análise de *cluster* hierárquica e não-hierárquica, outras técnicas como algoritmos evolutivos (JAIN, 1999), agrupamentos *fuzzy*, redes neurais (mapas de Kohonen), entre outras, podem ser empregadas para formação de agrupamentos. Aqui, ilustramos, brevemente, as técnicas de agrupamento *fuzzy* e mapas de Kohonen, indicando referências mais adequadas a essas técnicas.

4.4.1 Agrupamentos *fuzzy*

O agrupamento *fuzzy* é uma generalização dos métodos por particionamento e, assim como nos métodos por particionamento, também é necessário indicar o número inicial de grupos.

Nos métodos por particionamento, definem-se claramente em qual grupo ficará cada elemento, ou seja, são definidos agrupamentos rígidos (*crisp cluster*). Já os agrupamentos *fuzzy* permitem visualizar a grau de associação de cada elemento em cada grupo, que geralmente se verifica em domínios de dados reais, onde um elemento pertence a diferentes grupos, com diferentes graus de associação.

A principal vantagem dos agrupamentos *fuzzy* em relação aos outros métodos por particionamento, é que ele fornece informações mais detalhadas sobre a estrutura dos dados, pois são apresentados os graus de associação de cada elemento a cada grupo, não tendo, portanto, a formação de agrupamentos rígidos. A desvantagem desse método é que a quantidade de coeficientes de associação cresce rapidamente com o aumento do número de elementos e de grupos. Entretanto, trata-se de uma técnica válida, pois ela associa graus de incerteza aos elementos nos grupos e, essa situação, em geral, se aproxima mais das características reais dos dados (KAUFMAN, 1990).

Um algoritmo de agrupamentos *fuzzy* bastante utilizado é o *fuzzy c-means*. Trata-se de um algoritmo iterativo que inicia com *c* valores arbitrários, e com base nesses valores, associa cada elemento ao valor ao qual possui menor distância, formando *c* grupos. Em seguida, calcula-se o centro de cada grupo formado, e os elementos são reassociados ao centro mais próximo. Assim, os cálculos prosseguem, iterativamente, até que as diferenças entre os centros do passo atual e do anterior sejam mínimas.

Exemplo. Utilizando o software *Matlab*, aplicamos o algoritmo *fuzzy c-means* aos elementos da tabela 4.4, considerando a divisão dos elementos em dois grupos. A seguir, mostramos, o resultado apresentado pelo algoritmo.

ELEMENTOS	X	Y
1	4	3
2	2	7
3	4	7
4	2	3
5	3	5
6	6	1

Tabela 4.4: Elementos do exemplo *fuzzy c-means*.

Após a definição dos centros, foram calculados os graus de associação de cada elemento em cada grupo, como mostra a tabela 4.5.

ELEMENTO	GRUPO 1	GRUPO 2
1	0.9515	0.0485
2	0.0518	0.9482
3	0.0700	0.9300
4	0.6605	0.3395
5	0.1451	0.8549
6	0.8902	0.1098

Tabela 4.5: Graus de associação dos elementos aos grupos.

Os centros obtidos para os grupos foram:

CENTRO	X	Y
Grupo 1	4,3168	2,3009
Grupo 2	2,9578	6,2364

Observa-se na tabela 4.5, que os elementos 1, 4 e 6 apresentam maior grau de associação em relação ao grupo 1, enquanto os elementos 2, 3 e 5 apresentam maior grau de associação em relação ao grupo 2. Então, pode-se afirmar que:

- O elemento 1 pertence 95% ao grupo 1 e 5% ao grupo 2;
- O elemento 2 pertence 5% ao grupo 1 e 95% ao grupo 2;
- O elemento 3 pertence 7% ao grupo 1 e 93% ao grupo 2;
- O elemento 4 pertence 66% ao grupo 1 e 33% ao grupo 2;
- O elemento 5 pertence 15% ao grupo 1 e 85% ao grupo 2;
- O elemento 6 pertence 89% ao grupo 1 e 11% ao grupo 2.

Portanto, o algoritmo *fuzzy c-means* apresenta, como resultado os grupos (1,4,6) e (2,3,5), representados na figura 4.28, onde ´ é o centro de cada grupo:

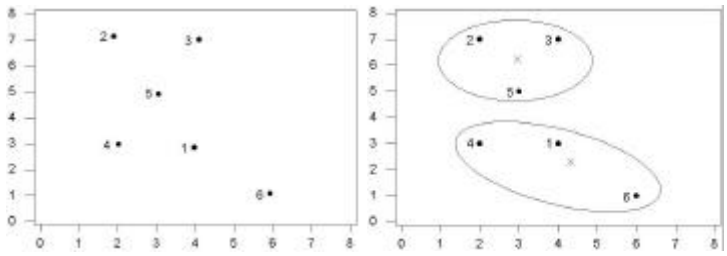

Figura 4.28: Agrupamentos realizados no algoritmo *fuzzy c-means*.

Maiores detalhes a respeito de agrupamentos *fuzzy* podem ser obtidos em (KAUFMANN, 1990) e (BEZDEK,1992).

4.4.2 Mapas Auto-Organizáveis de Kohonen

Os mapas auto-organizáveis de Kohonen constituem uma classe de redes neurais artificiais baseadas em aprendizado competitivo, em que os neurônios tendem a aprender a distribuição estatística dos dados de entrada.

A topologia da rede de Kohonen possui duas camadas, na qual todas as unidades de entrada encontram-se conectadas a todas as unidades de saída através de conexões sinápticas.

Essa rede utiliza regras de aprendizado competitivo, onde os neurônios de uma camada competem entre si pelo privilégio de permanecerem ativos, tal que o neurônio com maior atividade seja o principal participante do processo de aprendizado (*Winner-takes-all*).

O algoritmo de auto-organização é composto por quatro etapas, sendo a inicialização do mapa, o processo competitivo, o processo cooperativo e a adaptação sináptica.

A inicialização do mapa consiste na atribuição de um vetor de pesos aleatórios iniciais às conexões entre neurônios das camadas de entrada e saída. É necessário que este vetor de pesos seja normalizado, para que o somatório dos quadrados dos valores de pesos das conexões de entrada seja idêntico para cada neurônio de saída.

Assim, um vetor padrão de entrada é apresentado à rede, e este calcula o valor da distância quadrática para cada neurônio de saída em relação ao vetor de entrada. Este valor de proximidade entre o vetor de entrada e cada neurônio de saída *j* da rede é medido através do parâmetro d_j de distância Euclidiana, dado por:

$$d_j = \sum_{i=0}^{N-1} (u_i(t) - w_{ij}(t))^2 \tag{4.11}$$

- onde: $u_i(t)$ é a entrada ao neurônio *i*, no instante de tempo *t*;
$w_{ij}(t)$ é o peso entre o neurônio de entrada *i* e o de saída *j*, no instante de tempo *t*.

No processo competitivo, mediante a apresentação de um padrão de entrada, os neurônios competem entre si através de níveis de ativação, sendo que apenas um neurônio

será o vencedor. Assim, é selecionado o neurônio da camada de saída mais próximo ao padrão apresentado, tendo um d_j de valor mínimo.

O processo cooperativo é inspirado em um mecanismo neurobiológico, no qual o neurônio vencedor do processo competitivo tende a influenciar o estado dos neurônios vizinhos. Assim, a vizinhança é inicializada com um valor de largura d_0 e vai diminuindo a cada iteração, segundo a relação:

$$d_t = d_0\left(1 - \frac{t}{T}\right) \tag{4.12}$$

- onde: d_t é a largura da vizinhança para a iteração *t* atual e *T* é o número total de iterações.

O neurônio na posição (x, y) da matriz de saída será considerado como pertencente à vizinhança de (x_v, x_y), se:

$$(v - d_t) < x < (v + d_t) \text{ e } (v - d_t) < y < (v + d_t) \tag{4.13}$$

Na fase da adaptação sináptica, os pesos do neurônio vencedor e seus vizinhos são atualizados de modo a se aproximarem espacialmente do padrão de entrada, dado por:

$$w_{ij}(t+1) = w_{ij}(t) + h(t)(e_j(t) - w_{ij}(t)) \tag{4.14}$$

- para todo *j* na vizinhança de *v*.

O termo $h(t)$ corresponde à taxa de aprendizagem do algoritmo e também tem seu valor alterado a cada iteração, segundo a relação:

$$h(t) = h(0)\left(1 - \frac{t}{T}\right) \tag{4.15}$$

- onde: $h(0)$ é o valor inicial de h que decresce ao longo das iterações *t*.

Os elementos a serem agrupados são apresentados, um por vez, aos neurônios de entrada. A cada apresentação, os estímulos gerados pelo elemento são capturados pela camada de entrada e transmitidos igualmente a todos os neurônios da camada do mapa. O

neurônio que reagir mais fortemente aos estímulos do elemento apresentado ganha-o para si. Além disso, reforça suas ligações com os vizinhos próximos, sensibilizando-os um pouco mais às características do elemento capturado.

Numa próxima iteração, quando um elemento parecido for apresentado ao mapa, toda a região sensibilizada reagirá um pouco mais intensamente. Por outro lado, como os neurônios vizinhos são diferentes do neurônio ganhador, cada um reagirá mais intensamente a um elemento um pouco diferente.

A cada nova apresentação de um elemento ao mapa, o perfil de sensibilidade dos neurônios vai se alterando, isto é chamado de treinamento da rede. Estas alterações, no entanto, são cada vez menores, de forma que a configuração do mapa converge para uma disposição estável. Quando isto ocorre, o mapa aprendeu a classificar indivíduos.

Essas redes são úteis principalmente em reconhecimento de padrões, quando as classes a que devem pertencer os elementos a serem reconhecidos não são conhecidas inicialmente.

O resultado do processamento de uma rede treinada é que cada neurônio torna-se dono de um certo número de elementos, parecidos com os capturados pelos neurônios vizinhos. Desta maneira, elementos semelhantes vão sendo posicionados próximos entre si, formando um gradiente de características. Uma revisão detalhada deste tópico podem ser encontrada em (KOHONEN, 1997).

4.5 CONCLUSÃO

Neste capítulo, tratamos diversas técnicas de análise de *cluster* hierárquicas e não-hierárquicas, trazendo seus algoritmos, características, exemplos e funções distância empregadas. Maiores detalhes sobre as técnicas estatísticas de análise de *cluster* podem ser obtidos em (SNEATH, 1973), (JOHNSON, 1992), (KAUFMAN, 1990), (ROMESBURG, 1984) e (ANDERBERG, 1973). Além dessas, apresentamos, brevemente, outras técnicas

empregadas em análise de *cluster*, como mapas de Kohonen e agrupamentos *fuzzy*. Uma revisão detalhada desses métodos pode ser obtida em (JAIN, 1999), (KOHONEN, 1997), (KAUFMANN, 1990) e (BEZDEK,1992).

A seguir, aplicamos, em três conjuntos de dados, algumas técnicas hierárquicas e não-hierárquicas tratadas nesse capítulo.

5 ESTUDOS DE CASO

Nesse capítulo, realizaremos três estudos de caso, com diferentes conjuntos de dados, utilizando alguns métodos de análise de *cluster* apresentados nos capítulos anteriores.

Para essa análise, utilizaremos o *software* estatístico Minitab versão 13.2, da empresa *Minitab Inc*.. Esse *software* possui a maioria dos métodos de análise de *cluster* citados no trabalho. A figura 5.1 traz a tela inicial do *software*.

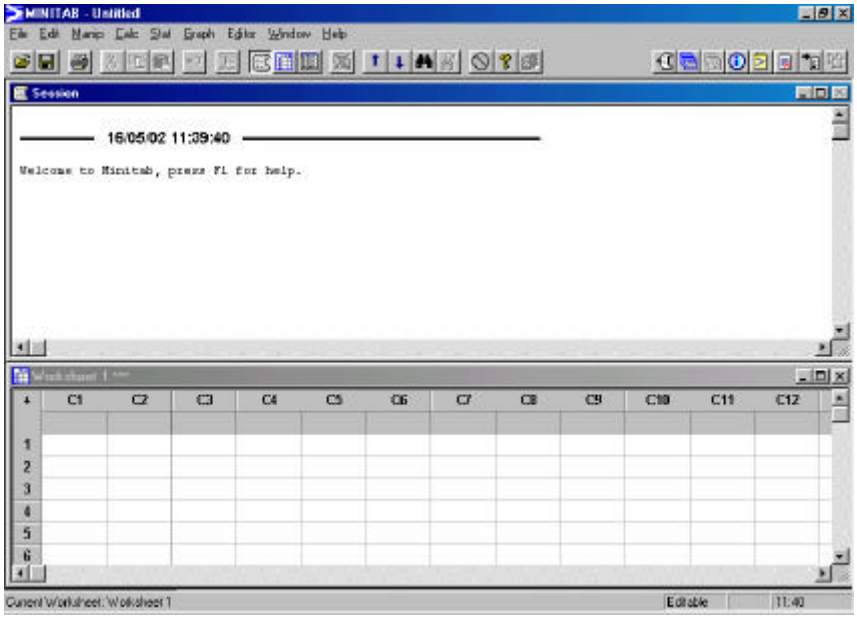

Figura 5.1: Tela inicial do Minitab.

Como podemos observar na figura 5.1, o software é dividido basicamente em três partes, sendo: o menu superior, o *log* (*Session*) e a planilha de dados (*Worksheet*). O menu superior possui todos os comandos do *software*, o *log* armazena o registro de todas as operações realizadas e a planilha armazena os dados a serem analisados.

A seguir, temos uma breve apresentação das funcionalidades do Minitab utilizadas no trabalho.

a) Análise descritiva dos dados e histograma - para realizar uma análise descritiva dos dados armazenados na planilha, devemos seguir, a partir do menu superior, os comandos *Stat* *Basic Statistics* *Display Descriptive Statistics*. Após isso, será exibida uma tela, conforme a figura 5.2.

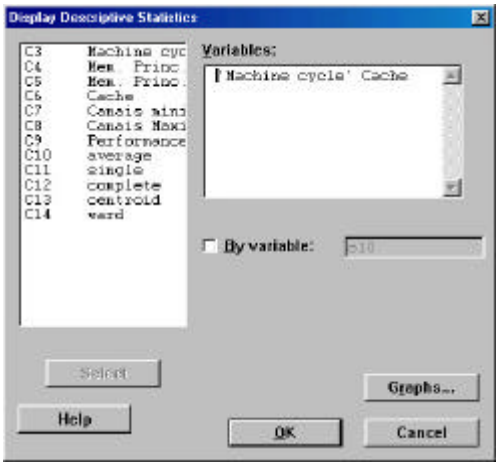

Figura 5.2: Tela de análise descritiva.

Selecionando as variáveis desejadas e clicando em *OK*, teremos, para cada variável, as seguintes estatísticas descritivas: número de elementos, média, mediana, média aparada, desvio padrão, erro padrão, mínimo, máximo, primeiro quartil e terceiro quartil. Também nessa tela, temos a opção de histograma, clicando em *Graphs* e selecionando *Histogram of data*.

b) Diagrama de dispersão – para criar um diagrama de dispersão dos dados, devemos seguir os comandos *Graphs* *Plot*. A seguir será exibida uma tela, conforme a figura 5.3.

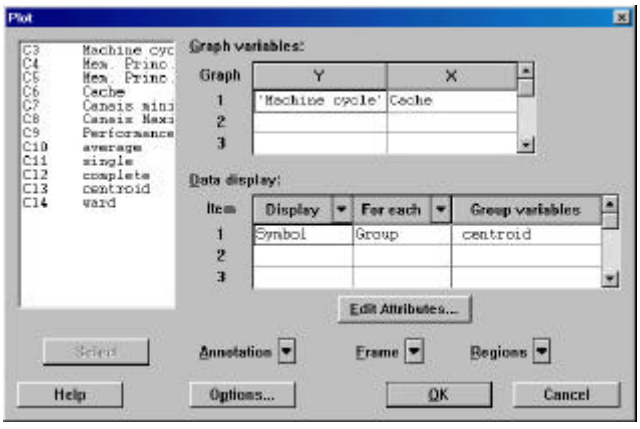

Figura 5.3: Tela de histograma.

c) Análise de *cluster* hierárquica - para realizar uma análise de *cluster* hierárquica, devemos seguir os comandos *Stat Multivariate Cluster Observations*. Em seguida, aparecerá uma tela (figura 5.4), onde temos a opção de selecionar as variáveis, o método, a função distância e a visualização do dendograma.

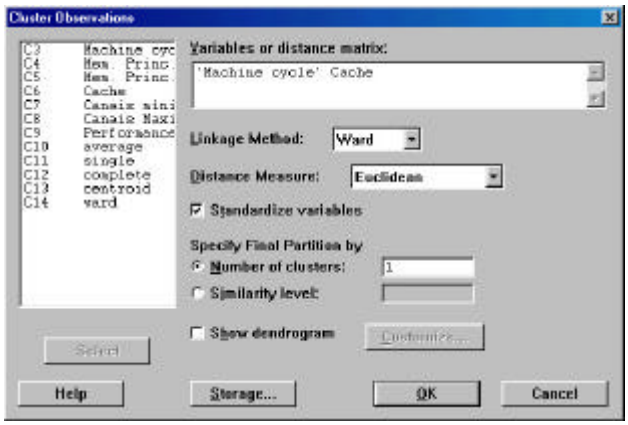

Figura 5.4: Tela de análise de *cluster* hierárquica.

d) Método do *k-means* - para aplicar o método do *k-means*, devemos seguir os comandos *Stat Multivariate Cluster k-means*. Em seguida, aparecerá uma tela (figura 5.5), onde devemos escolher as variáveis para análise. Clicando em *Storage* e selecionando uma variável em branco, os grupos ao qual foram associados os elementos serão armazenados.

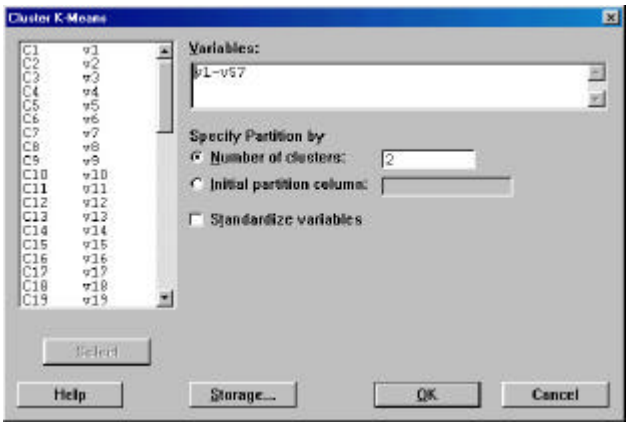

Figura 5.5: Tela do método *k-means*.

O Minitab possui outras funcionalidades, e para maiores detalhes sobre o *software* consulte (MINITAB, 2004).

5.1 ESTUDO DE CASO 1

Nesse estudo de caso, faremos uma análise de *cluster* utilizando um conjunto de dados de processadores de computador. Esse conjunto possui 209 registros, de 30 diferentes fabricantes obtidos de (FELDMESSER, 1987).

O objetivo dessa análise é agrupar os processadores que possuem características comuns entre as variáveis ciclo de máquina, medida em nanosegundos, e memória cache, medida em *kilobytes*.

A seguir, serão efetuadas simulações aplicando-se os métodos de ligação por vizinho mais próximo, ligação por vizinho mais distante, ligação de *Ward*, ligação por centróide e ligação por médias.

5.1.1 Simulações

Primeiramente, faremos uma análise descritiva das variáveis. O resultado é apresentado na tabela a seguir.

VARIÁVEL	MÉDIA	DESVIO PADRÃO	MÍNIMO	MÁXIMO
Ciclo de máquina	203,8	260,3	17	1500
Memória cache	25,21	40,63	0	256

Com base na tabela, observamos que o ciclo de máquina está entre 17 e 1500 nanosegundos, e apresenta um alto desvio padrão em relação à média. A variável memória cache está distribuída entre 0 e 256 *kilobytes*, também com um alto desvio padrão.

Para visualizar a distribuição das variáveis, temos o histograma da variável memória cache (figura 5.6).

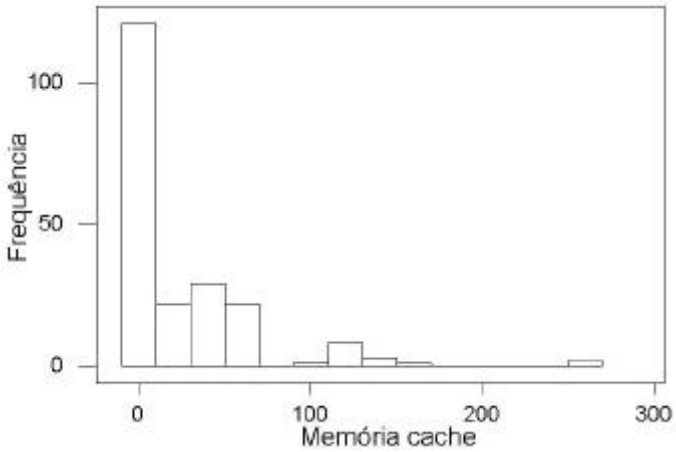

Figura 5.6: Histograma da variável memória cache.

Na figura 5.6, observamos que a maioria dos processadores possui memória cache entre 0 e 100. Existe ainda, um grupo menor com memória cache entre 100 e 200, e dois processadores (*outliers*) com memória de 256 *kilobytes*.

A figura 5.7 traz o histograma da variável ciclo de máquina.

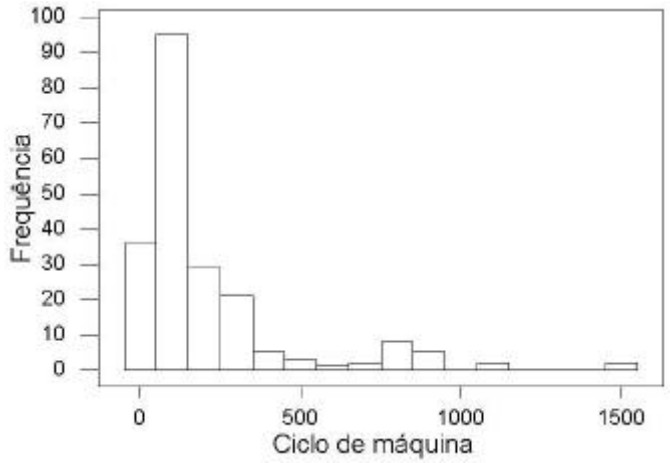

Figura 5.7: Histograma da variável ciclo de máquina.

Observamos, na figura 5.7, que a maioria dos processadores possui ciclo de máquina entre 0 e 500. Além disso, existe um grupo menor de processadores com ciclo de máquina entre 500 e 1000, e dois processadores (*outliers*) com ciclo de máquina de 1500.

Analisando as duas variáveis em conjunto, temos o diagrama de dispersão da figura 5.8.

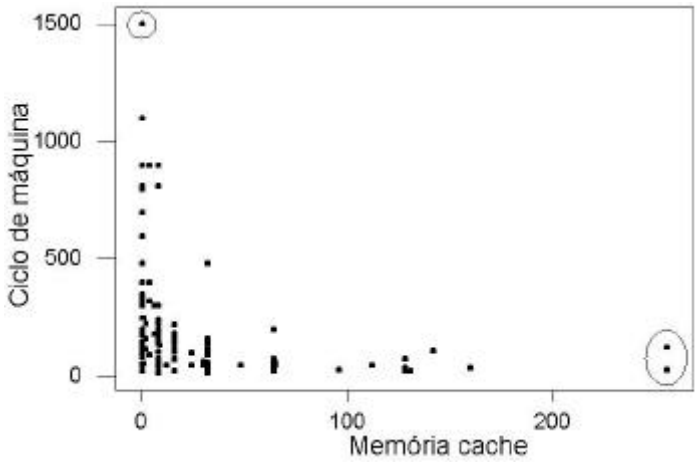

Figura 5.8: Diagrama de dispersão das variáveis ciclo de máquina e memória cache.

Na figura 5.8, observamos a presença de *outliers* (elementos circulados), os mesmos observados nos histogramas.

A seguir, aplicaremos os métodos de análise *cluster* hierárquicos. Devido à presença de *outliers* e conforme descrito no capítulo 4, não é adequado aplicar os métodos de ligação por vizinho mais próximo, ligação por vizinho mais distante e método de *Ward* a esse tipo de dado. Para demonstrar a ineficiência desses métodos aos dados a serem analisados, realizamos, também, os agrupamentos por esses três métodos.

5.1.1.1 Simulação 1: Aplicando o método de ligação por vizinho mais próximo

A figura 5.9 traz o dendograma aplicando-se o método de ligação por vizinho mais próximo ao conjunto de dados de processadores.

Figura 5.9: Dendograma utilizando o método de ligação por vizinho mais próximo.

Na figura 5.9, podemos observar que foram formados alguns grupos pequenos, e que esses grupos foram encadeados em um grupo único (grupo com a maioria dos elementos).

Com base no corte do dendograma da figura 5.9 (linha pontilhada), verificamos a divisão dos dados em 6 grupos.

A figura 5.10 traz o diagrama de dispersão, identificando os grupos por cores diferentes.

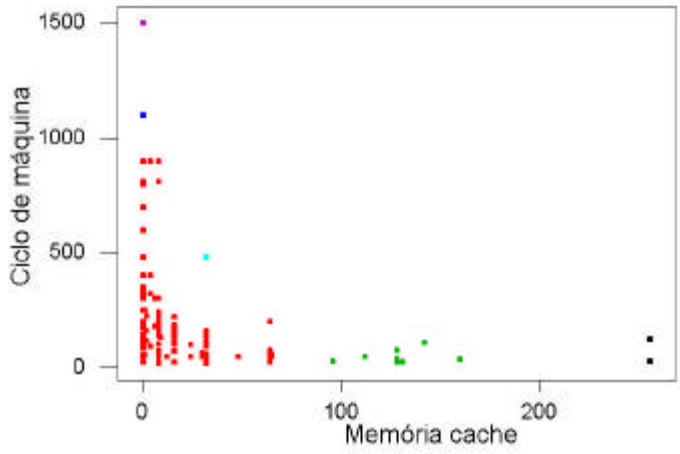

Figura 5.10: Diagrama de dispersão do método de ligação por vizinho mais próximo.

O resultado apresentado pelo método de ligação por vizinho mais próximo não foi satisfatório devido à presença de encadeamento, que ocorreu com o grupo dos elementos em vermelho, unindo processadores bem diferentes. Nesse grupo, foram agrupados desde

processadores com ciclo de máquina próximo de 0, até processadores com ciclo de máquina próximo de 1000.

5.1.1.2 Simulação 2: Aplicando o método de ligação por vizinho mais distante

A figura 5.11 traz o dendograma aplicando-se o método de ligação por vizinho mais distante.

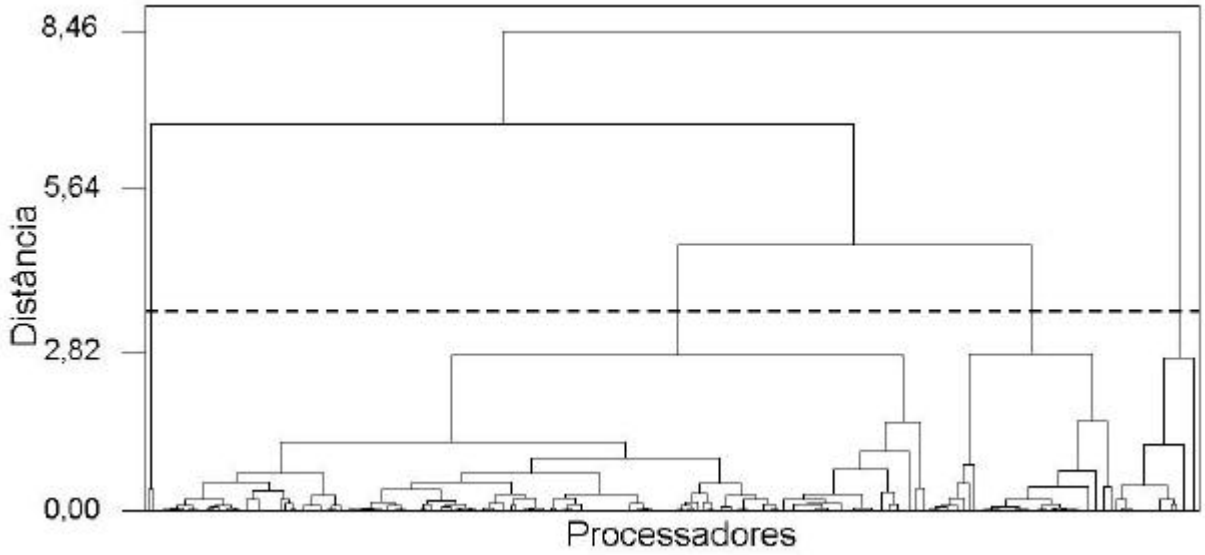

Figura 5.11: Dendograma do método de ligação por vizinho mais distante.

Com base no corte do dendograma da figura 5.11, verificamos a divisão dos dados em quatro grupos. A figura 5.12 traz o diagrama de dispersão, identificando os quatro grupos em cores diferentes.

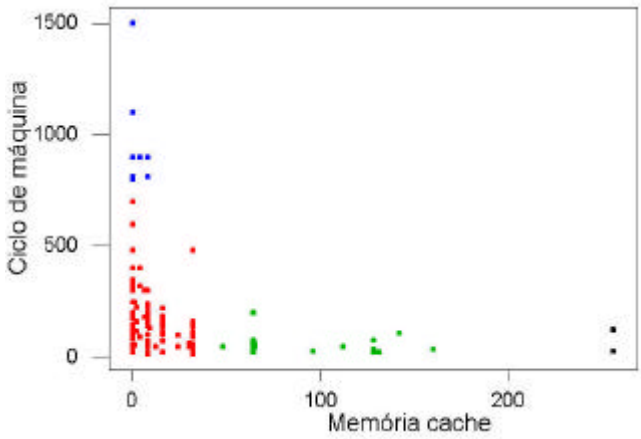

Figura 5.12: Diagrama de dispersão do método de ligação por vizinho mais distante.

O resultado apresentado pelo método de ligação por vizinho mais distante também não é satisfatório, devido à incorporação de dois dos *outliers* ao grupo identificado em azul.

5.1.1.3 Simulação 3: Aplicando o método de ligação de *Ward*

A figura 5.13 traz o dendograma aplicando-se o método de ligação de *Ward*.

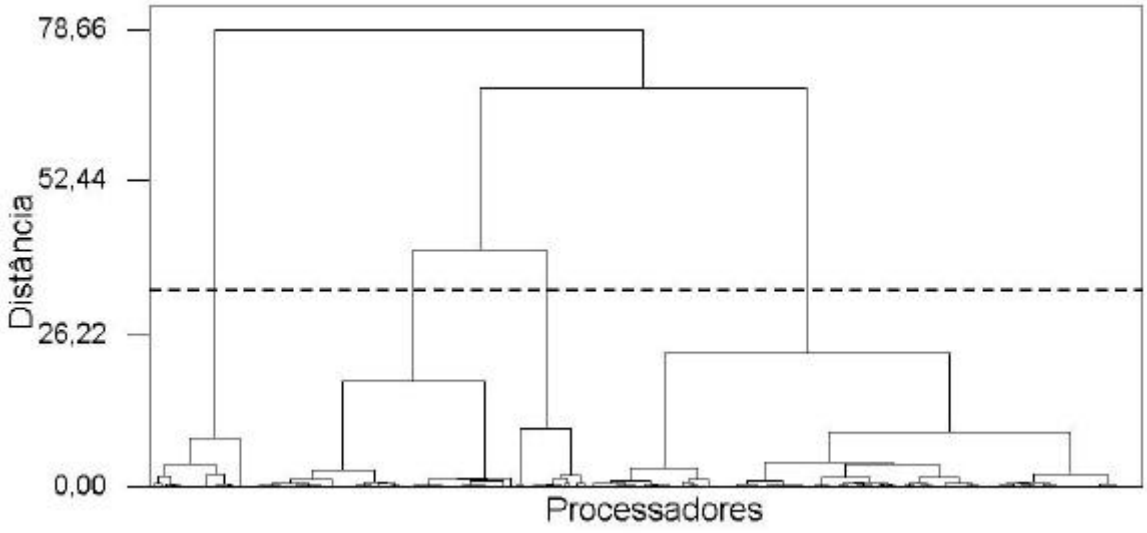

Figura 5.13: Dendograma do método de ligação de *Ward*.

Na figura 5.13, verificamos a divisão dos elementos em quatro grupos. A figura 5.14 traz o diagrama de dispersão, identificando os grupos em diferentes cores.

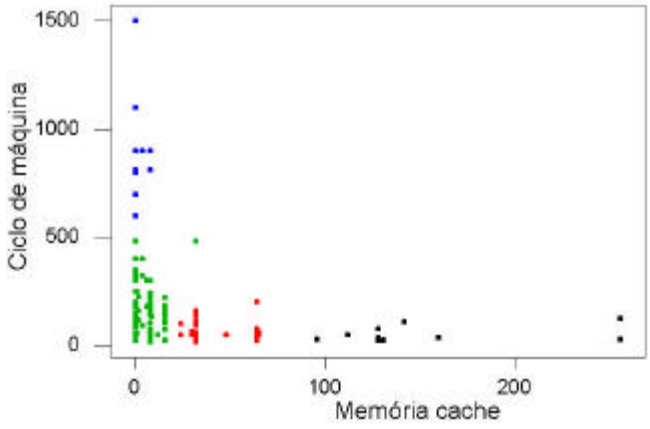

Figura 5.14: Diagrama de dispersão do método de ligação de *Ward*.

Os resultados apresentados pelo método de ligação de *Ward* não foram satisfatórios devido à incorporação de dois dos *outliers* ao grupo identificado em azul, e de outros dois ao grupo em preto.

5.1.1.4 Simulação 4: Aplicando os métodos de ligação por centróide e ligação por médias

Baseado na teoria do capítulo 4, observamos, a seguir, que para os dados analisados, os métodos de ligação por centróide e ligação por médias apresentam melhores resultados.

As figuras 5.15 e 5.16 trazem os dendogramas dos métodos de ligação por centróide e de ligação por médias, respectivamente.

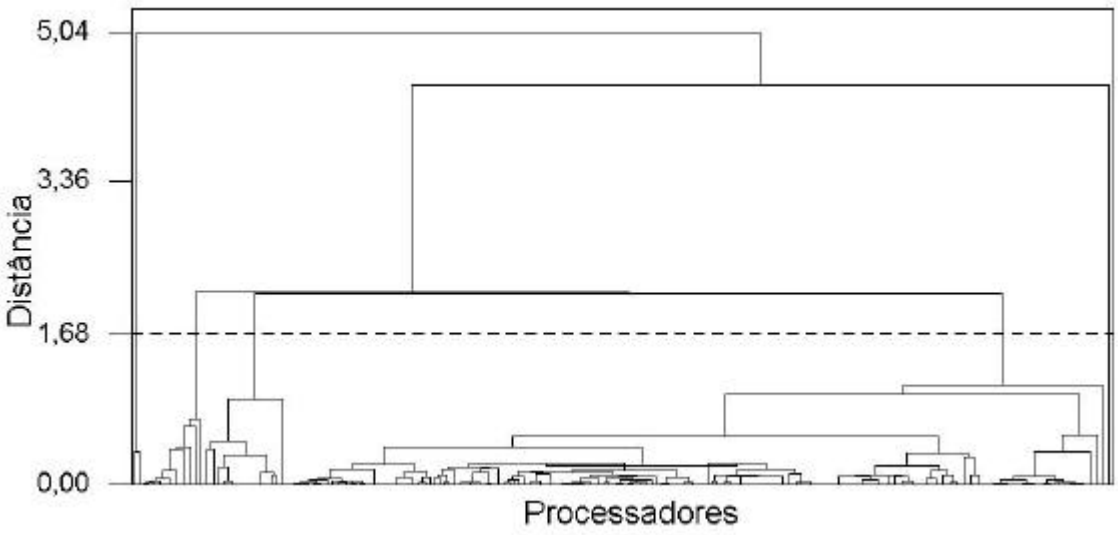

Figura 5.15: Dendograma do método de ligação por centróide.

Observamos, na figura 5.15, a divisão dos processadores em cinco grupos no método de ligação por centróide. No dendograma do método de ligação por média, ocorreu o mesmo, os dados foram divididos em cinco grupos, conforme figura 5.16.

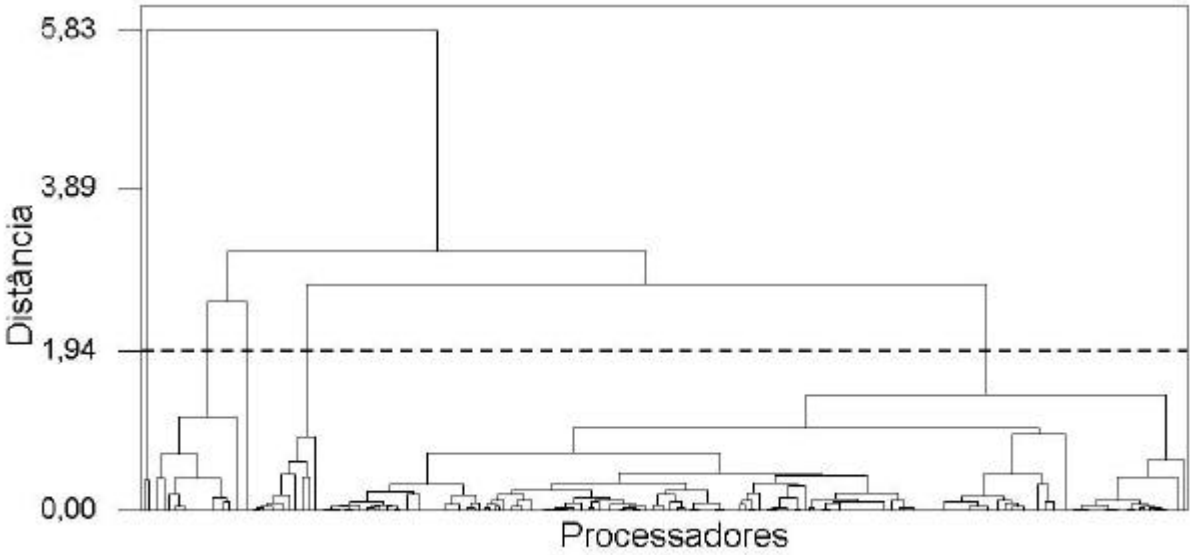

Figura 5.16: Dendograma do método de ligação por média.

Nos dois métodos, os grupos formados são os mesmos. Portanto, o gráfico de dispersão da figura 5.17 é válido para os dois métodos.

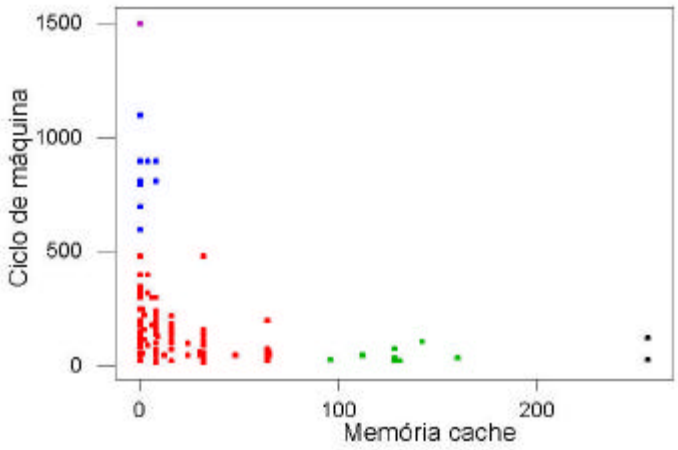

Figura 5.17: Diagrama de dispersão dos métodos de ligação por centróide e por média.

5.1.2 Conclusão

Os métodos de ligação por centróide e por média apresentaram melhores resultados aos dados analisados, assim, consideraremos os grupos obtidos por esses métodos o resultado final dos agrupamentos.

Os grupos formados pelo diagrama de dispersão da figura 5.17 estão distribuídos de acordo com a tabela 5.1.

GRUPO	QUANTIDADE DE PROCESSADORES	CICLO DE MAQUINA	MEMÓRIA CACHE
Preto	2	Baixo	Alta
Vermelho	174	Baixo	Baixa
Verde	13	Baixo	Intermediária
Azul	18	Intermediário	Baixa
Roxo	2	Alto	Baixa

Tabela 5.1: Grupos finais.

A tabela 5.2 traz a divisão dos valores das variáveis em três faixas.

CICLO DE MÁQUINA	AMPLITUDE	MEMÓRIA CACHÊ	AMPLITUDE
Baixo	17-500	Baixa	0-80
Intermediário	500-1100	Intermediária	80-160
Alto	1100-1500	Alta	160-256

Tabela 5.2: Classificação das variáveis.

A tabela 5.3 traz a descrição da quantidade de processadores por grupo, dividido por fabricante.

GRUPO	FABRICANTE	QUANTIDADE	FABRICANTE	QUANTIDADE
Preto	adviser	1	-	-
	nas	1	-	-
Vermelho	amdahl	8	hp	7
	apollo	2	ibm	21
	basf	2	ipl	6
	bti	2	magnuson	6
	burroughs	7	microdata	1
	c.r.d	6	nas	17
	cambex	5	ncr	11
	cdc	7	nixdorf	3
	dec	4	perkin-elmer	3
	dg	5	prime	5
	four-phase	1	siemens	11
	gould	2	sperry	9
	harris	7	sratus	1
	honeywell	13	wang	2
Verde	amdahl	1	-	-
	burroughs	1	-	-
	cdc	2	-	-
	gould	1	-	-
	nas	1	-	-
	ncr	2	-	-
	siemens	1	-	-
	sperry	4	-	-
Azul	dec	2	-	-
	dg	2	-	-
	formation	5	-	-
	ibm	9	-	-
Roxo	ibm	2	-	-

Tabela 5.3: Identificação dos fabricantes e quantidade de processadores por grupo.

A figura 5.18 traz a divisão final dos processadores em cinco grupos diferentes, classificados em três faixas de valores. Com essa divisão, podemos observar que os fabricantes de processadores produzem uma variedade maior de modelos de processadores de menor poder computacional, com utilização voltada às aplicações comerciais.

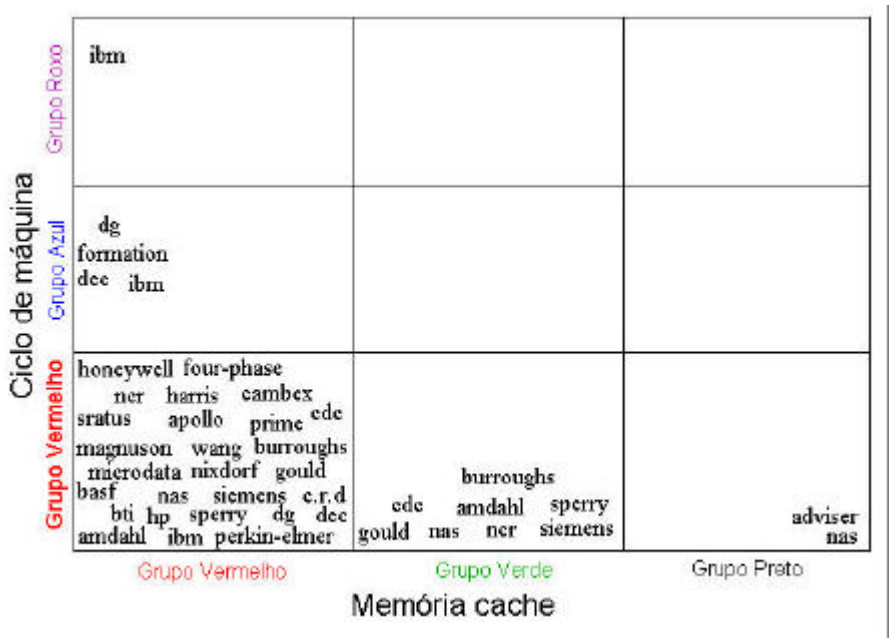

Figura 5.18: Grupos finais dos processadores.

5.2 ESTUDO DE CASO 2

Neste estudo de caso, utilizamos um conjunto de dados de 4601 e-mails classificados, previamente, como *spam* ou *não-spam* (HOPKINS, 1999). É considerado *spam* mensagens comerciais não solicitadas, entre outras.

As variáveis empregadas são descritas na tabela 5.4.

VARIÁVEIS	DESCRIÇÃO
1 a 48	Freqüência com que as palavras aparecem
49-54	Freqüência com que os caracteres aparecem
55	Média do número de letras em seqüência de letras maiúsculas
56	Número de letras na maior seqüência de letras maiúsculas
57	Soma do número de letras em seqüência de letras maiúsculas
58	Identificação: *spam* ou *não-spam*

Tabela 5.4: Descrição das variáveis do conjunto de dados de *e-mails*.

O objetivo, aqui, é verificar se o método *k-means* apresenta bons resultados na divisão dos e-mails em spam e não-spam. Para isso, aplicaremos o método *k-means* às 57 primeiras variáveis e, verificaremos se os grupos são os mesmos classificados conforme a variável 58.

5.2.1 Simulação aplicando o método *k-means*

Aplicando o método *k-means*, tendo $K = 2$, os e-mails foram divididos em um grupo com 3102 e-mails, considerados *não-spam* e, 1499 considerados *spam*.

Comparando o resultado encontrado pelo método *k-means* com a classificação real, temos a tabela 5.5.

REAL / K-MÉDIAS	SPAM	NÃO-SPAM	TOTAL
SPAM	2622 (56,99%)	166 (3,61%)	**2788 (60,60%)**
NÃO-SPAM	480 (10,43%)	1333 (28,97%)	**1813 (39,40%)**
TOTAL	**3102 (67,42%)**	**1499 (32,58%)**	**4601 (100,00%)**

Tabela 5.5: Identificação dos e-mails e fabricantes.

Observamos, na tabela 5.5, uma baixa porcentagem (3,61%) de e-mails *não-spam* que o método *k-means* classificou como *spam*. Porém, a porcentagem de e-mails *spam* classificados como *não-spam* é maior, 10,43%.

As porcentagens apresentadas na tabela 5.5 foram obtidas comparando-se cada *e-mail* classificado pelo método *k-means* ao respectivo *e-mail* previamente classificado.

5.2.2 Conclusão

Concluímos que se um usuário com o conjunto dos 3102 *e-mails* acima, utilizasse o método *k-means* como um filtro de sua caixa de *e-mails*, ele teria poucos problemas com *e-mails* desejados (166 mensagens) que foram direcionados para a caixa de *e-mail spam*, mas maiores problemas com *e-mails* indesejados (480 mensagens) que chegariam à sua caixa de entrada. Entretanto, um conjunto de variáveis mais adequado poderia permitir uma melhor detecção dos *e-mails*.

5.3 ESTUDO DE CASO 3

Neste estudo de caso, empregamos um conjunto de dados com 325.729 sites, indicando para cada página quais são seus *links*, obtidos de (ALBERT, 2002). Esse conjunto foi extraído do domínio *http://nd.edu*, da Universidade de Notre Dame, para estudo da estrutura de redes *internet*.

5.3.1 Simulação aplicando o método *k-means*

Manipulamos esse conjunto de dados e criamos duas variáveis; a primeira (ENTRADA), é o número de sites que apontam para um determinado site; a segunda (SAÍDA), é o número de sites para os quais cada página aponta (número de *links* por página).

A tabela 5.6 traz uma análise descritiva das variáveis.

VARIÁVEL	MÉDIA	DESVIO PADRÃO	MÍNIMO	MÁXIMO
ENTRADA	4,6	39	1	10721
SAÍDA	4,6	21,48	0	3445

Tabela 5.6: Análise descritiva das variáveis ENTRADA e SAÍDA.

Observamos que a variável ENTRADA está distribuída entre 1 e 10.721, e a variável SAÍDA está distribuída entre 0 e 3.445.

A figura 5.19 traz o histograma da variável ENTRADA.

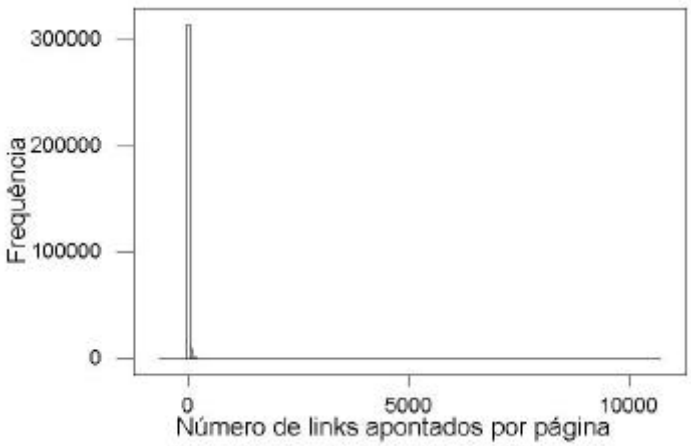

Figura 5.19: Histograma da variável ENTRADA.

Observamos, na figura 5.19, que a variável apresenta uma alta concentração nos valores próximos de zero, tendo um decréscimo exponencial extremamente rápido.

A figura 5.20 traz o histograma da variável SAÍDA.

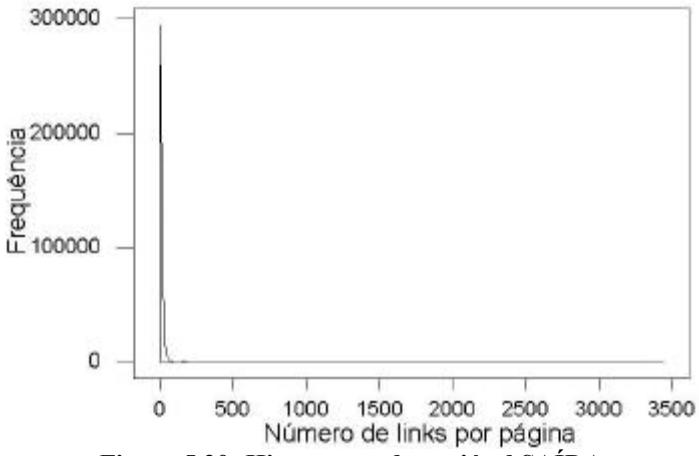

Figura 5.20: Histograma da variável SAÍDA.

Observamos, na figura 5.20, que a variável SAÍDA apresenta um comportamento semelhante ao da variável ENTRADA. A queda exponencial dessa variável já era esperada, por se tratar de um comportamento de redes sem escala (ALBERT, 2002).

Devido ao comportamento sem escala da rede, torna-se difícil visualizar os dados em sua escala original, tornando-se apropriado utilizar um gráfico na escala log-log, ou seja, com a freqüência e o número de *links* por página em escala logarítmica na base 10.

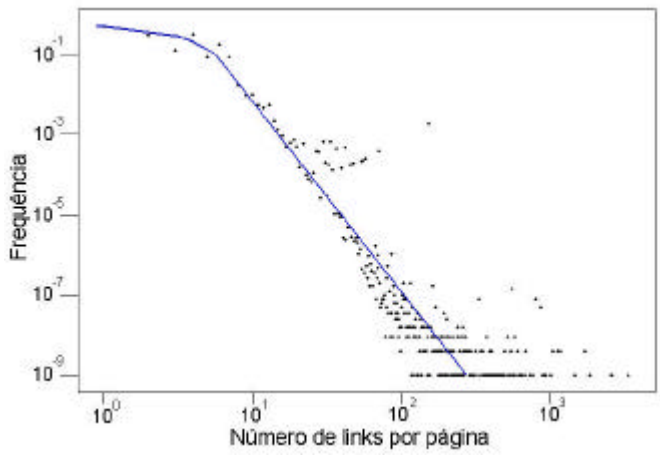

Figura 5.21: Gráfico log-log da distribuição do número de *links* por página.

Observamos, na figura 5.21, que os pontos (páginas) se distribuem ao longo de uma reta, caracterizando o comportamento exponencial, sem escala, do número de *links* na rede.

Uma rede sem escala é caracterizada segundo uma lei exponencial, onde a maior parte dos nós possui um pequeno número de conexões e, alguns, têm uma quantidade imensa de *links* (ALBERT, 2002). A rede *WWW* é considerada uma rede sem escala, pois obedece uma distribuição segundo a figura 5.22.

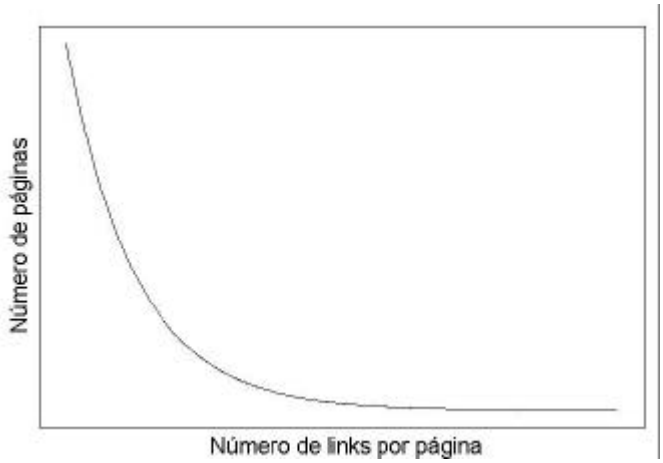

Figura 5.22: Distribuição exponencial dos *links*.

Devido ao grande número de páginas, seria impossível visualizar a rede de conexões entre elas de maneira gráfica e fazer inferências sobre o comportamento dessa rede. Utilizamos o método *k-means* para agrupar essas páginas e conseguir uma rede de grupos que resuma o comportamento das conexões.

Aplicando o método *k-means,* assumindo K=15, temos os grupos, conforme tabela 5.7. Observamos que se formaram grupos bastante heterogêneos entre si. Desde grupos, com páginas com poucos *links* e com poucas páginas apontando para elas, até grupos com páginas com mais de 7026 páginas que apontam para elas.

GRUPO	QUANTIDADE	MÍNIMA ENTRADA	MÁXIMA ENTRADA	MÍNIMO SAÍDA	MÁXIMA SAÍDA
1	3	7026	10721	0	17
2	8	3562	4300	2	21
3	237447	1	13	0	4
4	3262	1	23	23	106
5	1391	105	364	94	365
6	66330	1	13	1	22
7	5868	17	82	0	52
8	3100	1	105	24	142
9	335	79	322	0	61
10	238	1	47	140	486
11	6	1	7	1736	3445
12	58	1	11	497	1478
13	106	346	863	0	642
14	7528	7	52	0	28
15	49	863	2347	0	1058

Tabela 5.7: Resultado dos agrupamentos dos dados de páginas *web*.

Com base nestes grupos, utilizamos o *software* Pajek (PAJEK, 2004) e obtemos uma representação gráfica simplificada dessa rede, conforme as figuras 5.22 e 5.23.

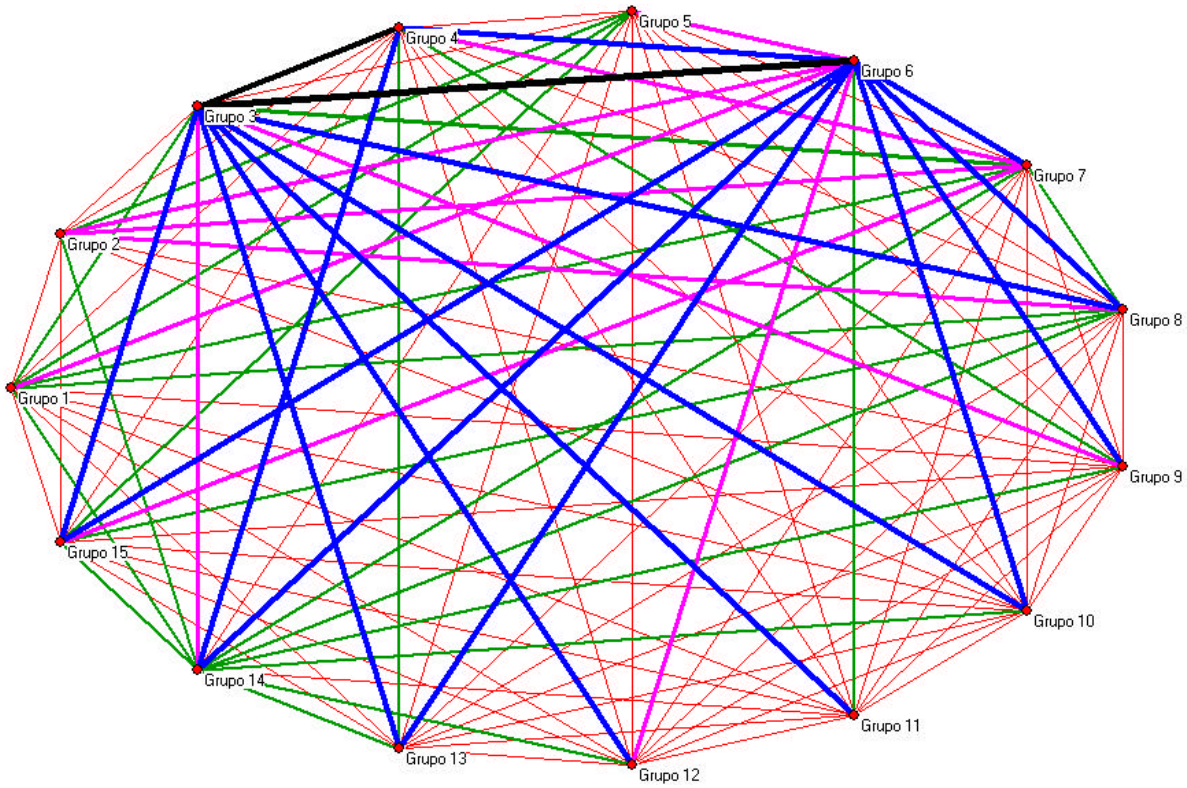

Figura 5.23: Desenho circular da rede após os agrupamentos.

Na figura 5.22, as cores das conexões indicam os graus de conectividade entre os grupos, onde a cor preta representa os grupos com maior conectividade, seguida do azul, magenta, verde e por fim, a cor vermelha, indicando menor conectividade.

A figura 5.23 traz as direções das conexões entre os grupos.

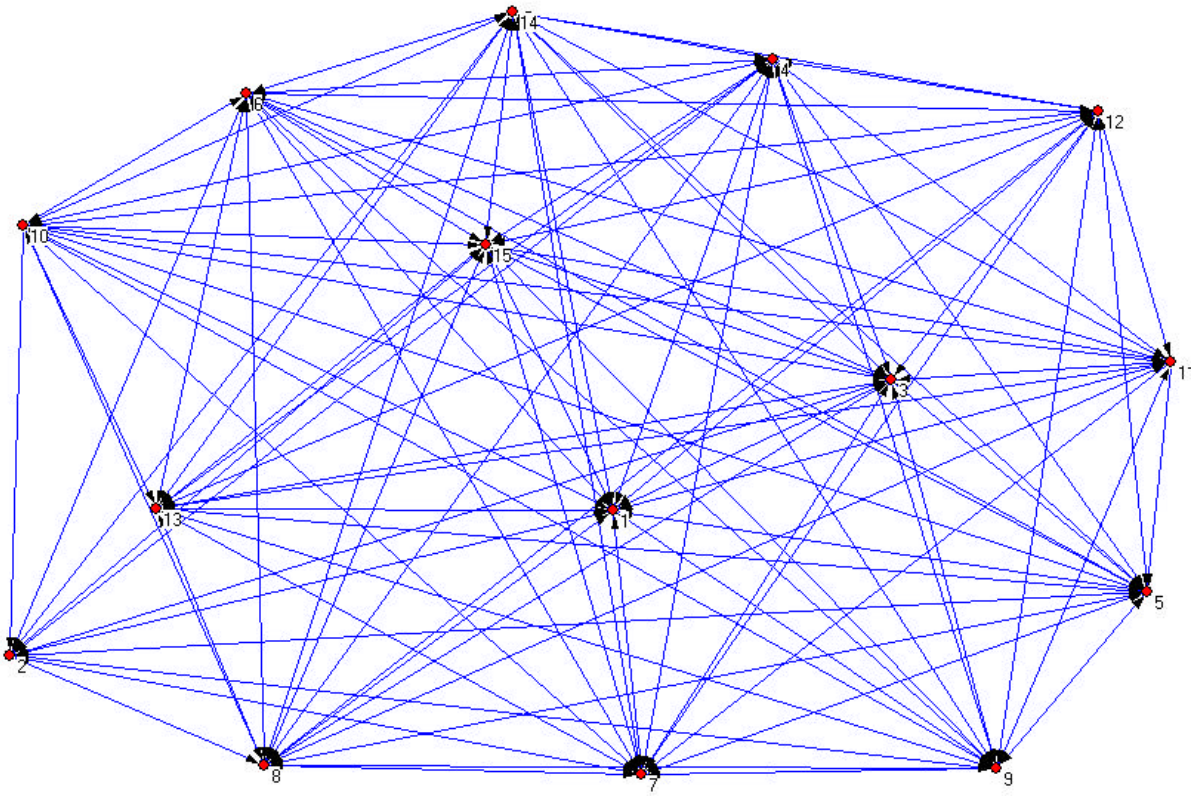

Figura 5.24: Desenho tridimensional da rede indicando as direções das conexões.

5.3.2 Conclusão

Nessa análise, observamos existência de alguns grupos que possuem um grande número de *sites* que apontam para eles, apesar de possuírem poucos elementos, como os grupos 1, 2. As páginas desses grupos são importantes para a rede, e caso haja alguma falha nesses grupos, a rede perderia diversas conexões. Outro grupo importante para a formação da rede, é o grupo 11, pois concentra grande quantidade de *links* em suas poucas páginas.

Verificamos, neste estudo de caso, a importância de técnicas de mineração de dados para a exploração de estruturas e características de redes complexas, como a rede *internet*.

6 CONCLUSÃO

O emprego da mineração de dados em grandes *Data Warehouses* é cada dia mais comum, seja para prover o suporte à decisão em sistemas empresariais, como em sistemas de relacionamento com clientes; seja no âmbito científico, na busca de estruturas e relações ocultas em grandes massas de dados, como no caso das informações sobre o genoma de uma espécie.

As técnicas de análise de *cluster* têm grande importância dentro das técnicas disponíveis de mineração de dados. Devido à complexidade e a escalabilidade das bases de dados, essas técnicas são eficientes, trazendo redução da complexidade, para uma melhor interpretação em processos decisórios. Neste estudo, damos destaque às técnicas de análise de *cluster* hierárquicas e de particionamento, nas quais uma função de similaridade desempenha um papel essencial. Esses métodos diferenciam-se de outras técnicas de agrupamento, muitas delas associadas a sistemas inteligentes como redes neurais e algoritmos genéticos.

Aplicando essas técnicas a alguns conjuntos de dados, verificamos sua aplicabilidade, sendo possível a extração de informações sobre a estrutura de dados relativamente complexos e volumosos, as quais eram previamente desconhecidas e difíceis de serem observadas sem uma redução de sua complexidade através da análise de *clusters*.

Embora técnicas de classificação (ou categorização) e análise de *cluster* tenham um resultado final similar, com a divisão de diferentes elementos em classes, ou agrupamentos, as técnicas de análise de *cluster* são mais poderosas e complexas, uma vez que as categorias, ou agrupamentos, não são previamente determinados.

Em uma primeira análise de *cluster*, realizada sobre um conjunto de dados sobre características de processadores computacionais, empregamos diversas técnicas hierárquicas de agrupamento descritas no capítulo 4, utilizando a distância Euclidiana como medida de

similaridade. Nas primeiras simulações, onde foram utilizadas as técnicas de ligação por vizinho mais próximo, ligação por vizinho mais distante, ligação de *Ward*, os resultados não foram satisfatórios, devido à presença de *outliers* e a forma como as funções distância definem os agrupamentos. Os métodos de ligação por médias e ligação por centróide obtiveram melhores resultados, dividindo, de forma correta, os diversos processadores.

Em uma segunda análise, utilizamos um conjunto de informações sobre *e-mails* para agrupa-los em *e-mails spam* e *não-spam,* resultado que apresenta um grande valor prático. Aqui, foi empregada a técnica não-hierárquica do *k-means*, utilizando a função de similaridade de distância Euclidiana. Essa análise apresentou um resultado bastante satisfatório, observando-se uma pequena porcentagem de erro, similar a outras técnicas classificação automática.

Em uma última análise realizada, um conjunto de dados de páginas *web* foi utilizado, e aplicamos a técnica de *cluster* não-hierárquica para particionar essas páginas em grupos similares, e obter uma representação gráfica e simplificada da rede.

Tendo em vista os resultados aqui obtidos, observamos a efetiva aplicabilidade das técnicas de *cluster,* hierárquica e não-hierárquica*,* as quais se mostraram eficientes no tratamento de dados complexos com recursos computacionais relativamente modestos. O sucesso, aqui verificado, do uso dessas técnicas sobre dados associados à estrutura *internet* (como *e-mails* e *websites*), indicam como promissores novos estudos e o aprofundamento do uso da análise de *cluster* em dados ligados a estrutura de redes complexas e à *internet*, para seu melhor entendimento.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBERT, Réka; JEONG, Hawoong; BARABÁSI, Albert-László. Diameter of the World Wide Web. Disponível em: <http://www.nd.edu/~networks/database/www/www.dat.gz>. Acesso em: 15 abr. 2004.

ANDERBERG, Michael R. *Cluster analysis for applications*. New York: Academic Press, 1973.

AURÉLIO, Marco; VELLASCO, Marley; LOPES, Carlos Henrique. *Descoberta de conhecimento e mineração de dados*. Pontifícia Universidade Católica, Laboratório de Inteligência Computacional Aplicada, 1999.

BARBARÁ, Daniel; CHEN, Ping. Using self-similarity to cluster large data sets. *Data Mining and Knowledge Discovery*, v. 7, n. 2, p. 123-152, Apr. 2003.

BEZDEK, James C.; PAL, Sankar K. *Fuzzy models for pattern recognition*: methods that search for structures in data. Piscataway: IEEE Press, 1992.

BRAGA, Antônio de Pádua et al. Redes neurais artificiais. In: REZENDE, Solange Oliveira (Org.). *Sistemas inteligentes*: fundamentos e aplicações. São Paulo: Malone, 2003, p. 141-168.

BUSSAB, Wilton de Oliveira; MIAZAKI, Édina Shizue; ANDRADE, Dalton Francisco de. *Introdução à análise de agrupamentos*. São Paulo: Associação Brasileira de Estatística, 1990.

DINIZ, Carlos Alberto R.; LOUZADA NETO, Francisco. *Data mining*: uma introdução. São Paulo: Associação Brasileira de Estatística, 2000.

FAYYAD, Usama M. et al. *Advances in knowledge discovery and data mining*. Massachusetts: MIT Press, 1996.

FAUSETT, Laurence. *Fundamentals of neural networks*: architectures, algorithms and applications. New Jersey: Prentice Hall, 1994.

FELDMESSER, Jacob; EIN-DOR, Phillip. Relative CPU performance data. Tel Aviv, Tel Aviv University, 1987. Disponível em: <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/cpu-performance>. Acesso em: 15 abr. 2004.

FREITAS, Alex A.; LAVINGTON, Simon H. Mining very large databases with parallel processing. Kluwer Academic Publishers. 1998.

HECKERMAN, David. Bayesian networks for knowledge discovery. In: FAYYAD, Usama M. et al. (Ed.). A*dvances in knowledge discovery and data mining*. Massachusetts: MIT Press, 1996, p. 273-305.

HOPKINS, Mark; REEBER, Erik; FORMAN, George. SPAM e-mail database. Palo Alto: Jaap Suermondt Hewlett-Packard Labs., 1999. Disponível em: <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/spambase>. Acesso em: 15 abr. 2004.

INMON, William H. *Como construir o Data Warehouse*. Rio de Janeiro: Campus, 1997.

JACKSON, Joyce. Data mining: a conceptual overview. *Communications of the Association for Information Systems*. v. 8, p. 267-296, Mar. 2002.

JOHNSON, Richard. A.; WICHERN, Dean W. *Applied multivariate statistical analysis.* 4th ed. New Jersey: Prentice Hall, 1992.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. *ACM Computing Surveys*, New York, v. 31, n. 3, p. 265-323, Sept., 1999.

KAUFMAN, Leonard; ROUSSEEUW, Peter J. *Finding groups in data*: an introduction to cluster analysis. New York: Wiley, 1990.

KOHONEN, T. *Self-organizing maps*. 2nd ed. Heidelberg: Springer, 1997.

MINITAB. *Tutorials Homepage*. Disponível em: <http://www.minitab.com/resources/tutorials>. Acesso em: 20 abr. 2004.

PAJEK. *Program for large network analysis*. University of Ljubljana. 2004. Disponível em: <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>. Acesso em: 15 abr. 2004.

REZENDE, Solange Oliveira et al. Mineração de dados. In: REZENDE, Solange Oliveira (Org.). *Sistemas inteligentes*: fundamentos e aplicações. São Paulo: Malone, 2003, p. 307-333.

ROMESBURG, Charles H. *Cluster analysis for researchers.* Belmont: Lifetime Learning Publications, 1984.

RUSSEL, Stuart J.; NORVIG, Peter. *Artificial intelligence*: a modern approach. Upper Saddle River: Prentice Hall, 1995.

SNEATH, Peter H.; SOKAL, Robert R. *Numerical taxonomy*: the principles and practice of numerical classification. San Francisco: W. H. Freeman, 1973.

ZAIANE, Osmar R. et al. *On data clustering analysis*: scalability, constraints and validation. Edmonton Alberta, University of Alberta, 2003.

WEISS, Sholon M.; KULIKOWSKY, Casimir A. *Computer systems that learn*: classification and prediction methods from statistics. Morgan Kaufman, 1991.

BIBLIOGRAFIA COMPLEMENTAR

CABENA, Peter et al. *Discovering data mining*: from concept to implementation. New Jersey: Prentice Hall, 1998.

CADEZ, Igor et al. Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery*, v. 7, n. 4, p. 399-424, Oct. 2003.

EVERITT, Brian S. *Cluster analysis*. 3rd ed. London: Edward Arnold, 1993.

HALKIDI, Maria; BATISTAKIS, Yannis; VAZIRGIANNIS, Michalis. On clustering validation techniques. *Journal of Intelligent Information Systems*, v. 17, n. 2-3, p. 107-145, Dec. 2001.

MACKINNON, Murray J; GLICK, Ned. Data mining and knowledge discovery in databases: an overview. *Australian New Zealand Journal of Statistics*, v. 41, p. 255-275, Sep. 1999.

MINERAÇÃO de Dados. Grupo de Sistemas Inteligentes. Maringá: Universidade Estadual de Maringá. Disponível em: <http://www.din.uem.br/ia/mineracao/geral/index.html>. Acesso em: 10 out. 2003.

STATLIB. *Data, software and news from the statistics community*. Disponível em: <http://lib.stat.cmu.edu/>. Acesso em: 27 set. 2003.

STATISTICA. *Data mining, data analysis, quality control, and web analytics software.* Disponível em: <http://www.statsoftinc.com/>. Acesso em: 10 nov. 2003.

WITTEN, Ian H.; FRANK, Eibe. *Data mining*: practical machine learning tools and techniques with Java implementations. San Francisco: Morgan Kaufmann, 1999.