UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
Departamento de Ciência da Computação

Relatório de Estudo
MAC5701 – Tópicos em Ciência da Computação

Tema: "Uma visão geral sobre Mineração de Dados"

Aluno: Andre Rodrigo Sanches
Orientador: Profa. Dr. Nina S. T. Hirata
Área de Concentração: Data Mining

Monografia apresentada como requisito parcial à aprovação na disciplina Tópicos em Ciência da Computação, do Programa de Pós-Graduação em Ciência da Computação.

São Paulo – Novembro de 2003

Sumário

Lista de Figuras

Capítulo 1

Introdução

A coleta de dados, seja por sistemas tradicionais, como por exemplo, transações bancárias, registros de compras ou por métodos inovadores, como através da Internet, tem atingido enormes proporções. A grande quantidade de dados assim coletados tornou-se um desafio para os gerentes, cuja função e a tomada de decisões. Os métodos tradicionais de transformação de dados em conhecimento dependem da análise e da interpretação pessoal dos mesmos, o que é um processo lento, caro e altamente subjetivo.

Neste contexto, faz-se necessária uma ferramenta capaz de extrair informações úteis para o suporte às decisões, estratégias de marketing e campanhas promocionais, dentre outras. A busca por estas informações é realizada utilizando sofisticadas técnicas de inteligência artificial na análise daqueles dados, a fim de encontrar padrões e regularidades nos mesmos. A esse processo dá-se o nome de Descoberta de Conhecimento em Banco de Dados *(KDD - Knowledge Discovery in Database)*.

Um passo particular do KDD é aquele denominado de *data mining* que consiste na aplicação de algoritmos específicos para a extração de padrões a partir da bases de dados. Do dicionário **Babylon**, temos a tradução destes termos:

mine extrair; explorar; minar;

mining escavação, desaterro; exploração de minas.

As ferramentas de *data mining* podem prever futuras tendências e comportamentos, permitindo às empresas um novo processo de tomada de decisão, baseado principalmente no conhecimento acumulado, e freqüentemente desprezado, contido em seus próprios bancos de dados.

A mudança de paradigma, causada por uma conjunção de fatores, como o grande acúmulo e coleta de dados, o relativo barateamento do processamento e dos computadores e o surgimento de novas oportunidades, como o marketing direto, trouxe um desenvolvimento ímpar para as técnicas de descoberta de conhecimento.

Ao longo deste trabalho serão apresentados os cenários onde KDD é comumente uti-

lizado, bem como, as bases para a correta e produtiva aplicação destas técnicas. O conhecimento sobre a forma de como os dados estão armazenados aumentam as chances de acerto na escolha das ferramentas de prospecção. Portanto, será traçado um perfil das formas mais comuns de bancos de dados existentes.

1.1 O *dado* como informação

Um *dado* é a estrutura fundamental sobre a qual um sistema de informação atua. A *informação* pode ser vista como uma representação ordenada e enxuta dos dados resultantes de uma consulta que permite a visualização e interpretação dos dados. O conhecimento provém da interpretação, geralmente pessoal, das informações apresentadas pelo sistema de banco de dados. Inteligência é o bom uso do conhecimento adquirido. A pirâmide do conhecimento, representada na figura 1.1, demonstra que a quantidade de dados existentes em um sistema é muito grande. Já o montante de informação é reduzido devido a dados errôneos ou sem expressão. Menor ainda é a quantidade de conhecimento que pode ser extraído desta informação através de técnicas de descoberta de conhecimento.

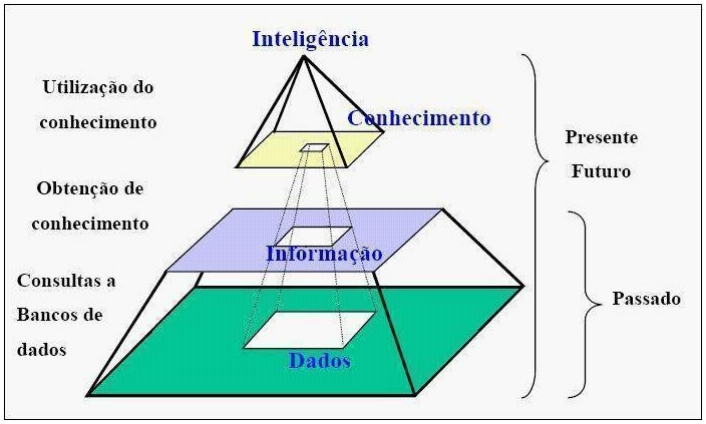

Figura 1.1: Pirâmide do conhecimento

1.2 As raízes da mineração de dados

Embora o mercado atual de mineração de dados seja caracterizado por uma série de novos produtos e companias, este assunto possui tradição de prática e de pesquisa de pouco mais de 30 anos. O primeiro nome utilizado para mineração de dados no início dos anos 60, era **análise estatística**. Os pioneiros da análise estatística foram SAS, SPSS, and IBM. Todas estas três companias são ativas no campo da mineração hoje em dia e oferecem produtos de muita credibilidade, baseados em seus longos anos de experiência. Originalmente, a análise estatística consistia em rotinas estatísticas clássicas tais como a correlação, a regressão, o chi-quadrado e a tabelação transversal. O SAS e SPSS oferecem

ainda estas aproximações clássicas. O processo de mineração foi além destas medidas estatísticas e evoluiu para as aproximações mais compreensíveis que tentam explicar ou predizer o que está sobre nos dados.

No final dos anos 80, a análise estatística clássica ganhou um conjunto mais eclético de técnicas com nomes tais como a lógica fuzzy, o raciocínio heurístico e redes neurais. Este era o auge da IA (inteligência artificial), em inglês *AI (Artificial Inteligence)*. Embora talvez seja uma áspera crítica, deve-se admitir que o IA era uma falha como o que foi apresentado e na década de 80. O que foi prometido era demasiadamente distante das possibilidades. Os sucessos de IA giravam em torno de domínios especiais do problema e requeriam freqüentemente um complicado investimento na codificação do conhecimento de um perito humano no sistema. E mais sério ainda: IA remanesceu para sempre uma caixa preta em que a maioria que de nós não conseguimos encontrar relação. Tente vender a um presidente de empresa um pacote caro que execute "a lógica fuzzy".

No final dos anos 90, aprendeu-se como tirar proveito das melhores abordagens da análise estatística clássica, das redes neurais, das árvores da decisão, da análise de mercado e de outras técnicas poderosas, de uma maneira muito mais convincente e eficaz. Adicionalmente, a chegada de sistemas sérios de *data warehouse* tem sido um ingrediente necessário que fêz a mineração de dados algo real e tangível.

Historicamente, a noção de encontrar padrões úteis em dados em seu estado bruto tem recebido diversos nomes, inclusive extração de conhecimento de informação, coleta de informação, arqueologia de dados ou padronização de dados [Ama01]. Esse processo surgiu em 1989 para encontrar o conhecimento existente em uma base de dados e enfatizar o alto nível das aplicações dos métodos de prospecção de dados. O crescente interesse por exploração e prospecção de dados culminou com a certeza de que os especialistas desta área nem sempre estavam em concordância com os especialistas de áreas afins. Para resolver essa discordância, foi organizada uma série de workshops sobre o processo KDD nos anos de 1989, 1991, 1993 e 1994. Com o aumento do interesse, realizou-se em 1995, na cidade de Montreal no Canadá, a primeira Conferência Internacional de Prospecção de Dados *(First International Conference on Knowledge Discovery and Data Mining)*, realizada durante a $14a$ Conferência Internacional de Inteligência Artificial (IJCAI-95). Segundo [FPSM91], KDD é um processo não trivial de identificacão válida do padrão dos dados. É um processo novo, potencialmente útil e fundamentalmente compreensível.

O KDD evoluiu, e continua evoluindo, da interseção de pesquisa em campos tais como bancos de dados, aprendizado de máquinas, reconhecimento de padrões, estatísticas, inteligência artificial, aquisição de conhecimento para sistemas especialistas, visualização de dados, descoberta de máquina, descoberta científica, recuperação de informação, e computação de altodesempenho. Os sistemas de software KDD incorporam teorias, algoritmos, e métodos de todos estes campos.

1.3 O processo de KDD

O termo processo implica que existem vários passos envolvendo preparação de dados, procura por padrões, avaliação de conhecimento e refinamento. Todos estes passos são interativos e iterativos [BL99] (veja figura 1.2), ou seja, dependem da constante interferência de um técnico especialista e se repetem de acordo com a necessidade.

1. **Conhecimento do domínio da aplicação:** inclui o conhecimento relevante anterior e as metas da aplicação, ou seja, a identificação do problema. Este passo utiliza o domínio do especialista para identificar problemas importantes e os itens necessários para resolvê-los. Entretanto, é importante que esta etapa seja realizada em conjunto com um engenheiro de conhecimento.

2. **Criação de um banco de dados alvo:** definir o local de armazenamento e selecionar um conjunto de dados ou dar ênfase para um subconjunto de dados nos quais o "descobrimento" será realizado.

3. **Pré-processamento:** inclui operações básicas como remover ruídos [1] ou subcamadas, se necessário, coletando informação necessária para modelar, decidindo estratégias para manusear (tratar) campos onde nota-se facilmente que não influenciam na solução das perguntas que se deseja responder.

4. **Transformação de dados e projeção:** inclui encontrar formas práticas para se representar dados, dependendo da meta do processo e o uso de redução dimensionável e métodos de transformação para reduzir o número efetivo de variáveis que deve ser levado em consideração; ou encontrar representações invariáveis para os dados.

5. **Mineração de dados:** inclui a decisão do propósito do modelo derivado do algoritmo de mineração. Além dessa decisão é necessário selecionar métodos para serem usados na procura por padrões nos dados, bem como decidir quais modelos e parâmetros podem ser apropriados, determinando um método de mineração particular a ser aplicado.

6. **Interpretação:** inclui a interpretação dos padrões descobertos e o possível retorno a algum passo anterior, além de uma possível visualização dos padrões extraídos, removendo aqueles redundantes ou irrelevantes e traduzindo os úteis em termos compreendidos pelos usuários.

7. **Utilização do conhecimento obtido:** inclui a necessidade de incorporar este conhecimento, para melhora de performance do sistema, adotando ações baseadas no conhecimento, ou simplesmente documentando e reportando este conhecimento para grupos interessados.

[1] Referem-se a dados que provavelmente contenham erros de digitação ou valores absurdos.

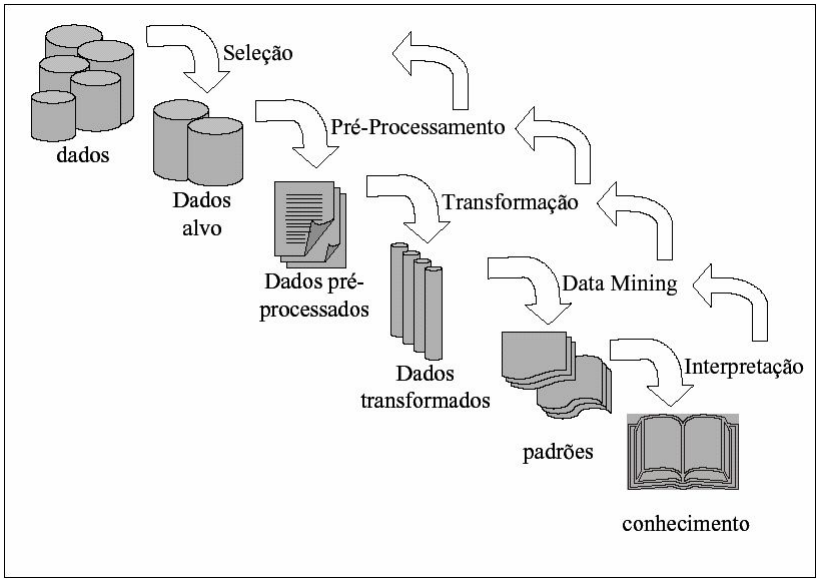

Figura 1.2: Processo de descoberta de conhecimento

1.4 Problemas e desafios

Os sistemas de mineração de dados impõem diversos problemas no seu incremento pois os bancos de dados geralmente apresentam tendência a serem dinâmicos, incompletos, grandes e repletos de informações irrelevantes. Alguns destes problemas são discutidos a seguir.

1.4.1 Informação limitada

Um banco de dados geralmente é projetado para propósitos diferentes de mineração de dados e em muitos casos as propriedades ou atributos que simplificariam a tarefa de aprendizado não estão presentes e nem podem ser requisitados do mundo real (adicionadas ao banco). Dados sem conclusão causam problemas quando alguns atributos essenciais sobre o domínio da aplicação não estão presentes nos dados, o que torna impossível descobrir conhecimento significativo sobre tal aplicação. Por exemplo, não se pode usar mineração de dados para diagnosticar malária de um banco de dados de pacientes, se esse banco não contém um campo determinando a contagem de células no sangue de cada paciente.

1.4.2 Valores perdidos

Grandes bases de dados normalmente estão repletas de erros originados: da modelagem, de dados inconsistentes ou de sistemas aplicativos mal concebidos. Nesse cenário não se pode assumir que os dados aqui contidos sejam confiáveis. Erros no valor de atributos ou

informação de classe são conhecidos como ruídos. É obviamente desejável a eliminação de qualquer ruído da informação a ser classificada pois eles afetam a precisão das regras e padrões gerados. Dados inválidos podem ser tratados através de sistemas de descoberta de vários modos:

- Desconsiderando os atributos ruidosos de cada registro;
- Omitindo os registros que contêm dados inválidos;
- Deduzindo valores inválidos a partir de valores conhecidos (predição);
- Tratando os valores inválidos como um valor especial;
- Calculando uma média sobre os valores inválidos.

1.4.3 Tamanho, atualização e campos irrelevantes

Os bancos de dados costumam ser dinâmicos, dado que seus conteúdos provêm de transações efetuadas e com isso informações são somadas, modificadas e removidas constantemente. O problema na perspectiva de mineração de dados está na forma de garantir que as regras estão sempre atualizadas e consistentes com a informação mais atual constante na base de dados. O sistema de aprendizado tem de ser sensível à passagem do tempo, pois alguns dados variam. O sistema de descoberta sempre é afetado pela atualização dos dados.

1.4.4 Variedade dos objetos

Hoje em dia os bancos de dados armazenam tipos de dados e estruturas cada vez mais complexos. Não são mais apenas valores numéricos e *strings* que constituem os registros dos bancos de dados, mas sim dados orientados a objetos, hipertexto, multimídia, sons, imagens, vídeos, mapas geográficos, dados temporais e espaciais e outros objetos que possuem operações mais complexa do que as informações mais rudimentares de anos atrás.

1.4.5 Eficiência e escalabilidade

Para extrair informação de modo eficaz, a partir de um banco de dados de grande porte, o algoritmo de mineração deve ser eficiente e escalável, ou seja, o tempo de execução do algoritmo deve ser previsível e aceitável em grandes bancos de dados. Algoritmos de ordem de complexidade exponencial ou mesmo de alta ordem polinomial não terão uso prático [CHY96].

1.4.6 Diferentes fontes de dados

A grande variedade de redes locais e *wide-area*, incluindo a Internet, conectam muitas fontes de dados de enormes e heterogêneos bancos de dados. Efetuar o processo de mineração em ambiente de diferentes fontes de dados, formatados ou não, com diversos significados semânticos, é ainda mais complexo. Por outro lado, a mineração pode ajudar a quebrar a barreira do alto nível de regularidade nos dados em bancos heterogêneos, o que dificilmente seria possível através de sistemas de simples consulta. Ainda, o enorme tamanho dos bancos, o abrangente distribuição dos dados e a complexidade computacional de alguns métodos de mineração motiva o desenvolvimento de algoritmos paralelos e distribuídos de mineração de dados [CHY96].

1.5 O processo de mineração

Serão descritas as tarefas específicas envolvidas no processo de descoberta de conhecimento. O processo é iterativo e volta comumente para fases anteriores com o objetivo de descobrir novos padrões e de melhorar a compreensão dos dados.

Os passos do processo de data mining são os seguintes:

1. **Identificação da fonte de dados**
   A tarefa de identificar os dados começa com a decisão sobre que dados serão necessários para se resolver um problema. Por exemplo, previsões sobre o comportamento dos clientes são freqüentemente uma meta necessária. Pensando em termos de um problema, o investigador tem que identificar os dados necessários para uma solução e outras possíveis fontes de dados.

   Os dados podem estar em uma difícil localização ou em um formato desconhecido. Às vezes há alguns bancos de dados iniciais que podem ser incompatíveis com novas bases de dados. Muitas vezes, se os dados estão escassos ou incompletos, pode ser preciso mais dados. A forma na qual os novos dados serão colecionados depende do formato da base de dados existente.

2. **Preparação dos Dados**
   Os dados muitas vezes necessitam de modificações antes de ser trabalhados ou minerados. Esse passo é geralmente chamado de *Cleaning*. Especificamente, os desafios seguintes são comuns:

   - Os dados podem estar em um formato incompatível com a representação do software de mineração que será utilizado;
   - Dados podem estar mal escritos ou escritos erroneamente, ou até mesmo ter valores incompletos, ou errôneos;

- Descrições de campo podem estar obscuras ou confusas, ou podem significar coisas diferentes que dependem da fonte. Por exemplo, data de ordem pode significar a data em que a ordem foi enviada, carimbada, recebida ou teclada;
- Dados podem estar obsoletos; por exemplo, os clientes podem ter mudado de endereço;
- Até mesmo dados bastante claros podem necessitar de uma transformação anterior para que sejam satisfatórios para mineração ou visualização.

A transformação pode melhorar em muito o desempenho do modelo. Se fosse o caso de analisar dados de uma companhia telefônica, por exemplo, poderia ser encontrada a taxa de ligação a longa distância (vendas dividido pelos minutos totais usados) e determinada uma previsão melhor do comportamento do cliente do que o estudo dos dados separadamente.

Transformações de dados estão no coração do desenvolvimento de um modelo de mineração de dados. Os dados podem ser transformados de diversas maneiras:

- Somando colunas, normalmente aplicando uma fórmula matemática para os dados existentes, criando assim um novo campo;
- Removendo colunas que não são pertinentes, são redundantes, ou contêm campos obvios e desinteressantes;
- Filtrando o banco de dados em uma expressão booleana usando valores de uma coluna para influenciar o modelo ou a visualização. Por exemplo, é possível visualizar apenas as regras mais fortes ou os segmentos de clientes mais lucrativos;
- Dados agregados, agrupando registros, e achando a soma, máximo, mínimo, ou valores comuns;
- Testando os dados para adquirir um subconjunto casual de dados (por porcentagem ou contagem);
- Aplicando um classificador, regressor, ou agrupando outros modelos que foram criados previamente.

A preparação dos dados pode ser em muito amenizada ao se aplicar mineração de dados em um *data warehouse - DW*. Toda e qualquer transformação que os dados necessitem podem ser ou já estão realizadas dentro do proprio sistema DW, aumentando em muito a performance do sistema. De outro modo, a aplicação de mineração em bases de dados desestruturadas exige um esforco computacional imenso, apesar de ser perfeitamente possível.

3. **Construção de um modelo**
   Durante o processo de mineração, será construído um modelo que é constituído das regras que descrevem os dados analisados no banco. Isso é feito automaticamente através de dados analíticos e de algoritmos de mineração de dados. É possível utilizar

todos os dados de que dispomos para a construção do modelo, ou pode-se reservar uma amostra dos dados para futuros testes de precisão do modelo. Estas escolhas não só influenciam o modo como a visualização é apresentada como tambem as decisões que o algoritmo toma na construção do classificador.

4. **Avaliação do modelo**
   Avaliar a precisão de um modelo refina sua compreensão e sua utilidade. Algumas estratégias, principalmente a de árvore de decisão e a de árvore de opção, avaliam diferentes partes do modelo e têm uma representação visual fácil dessa avaliação.

5. **Desdobramento do modelo**
   Um modelo pode ser desdobrado para ser aplicado a novos dados. Outra base de dados pode dar lugar ao surgimento de novas perguntas, trazendo um refinamento adicional às descobertas.

Um bom exemplo é o do modelo criado para determinar as causas pelas quais clientes estavam dispostos a deixar de optar pelos serviços de uma companhia. Poderiam ser avaliados registros de clientes pelo modelo para identificar os clientes específicos que provavelmente recusariam o serviço. A estes clientes poderiam ser oferecidas opções para incentivá-los a manter contrato com a companhia.

1.6 Classificação das técnicas utilizadas

Diferentes esquemas de classificação podem ser aplicados à área de mineração de dados, aos métodos utilizados, tipos de bancos de dados a serem analisados, tipos de técnicas, como mostra-se abaixo [CHY96]:

- **Tipo de banco de dados a ser analisado**
  Um sistema de mineração de dados pode ser classificado de acordo com os bancos de dados nas quais a mineração será realizada. Por exemplo, um sistema será um minerador de dados relacional se descobre conhecimenho a partir de dados relacionais, ou será orientado a objetos se o processo minera dados a partir de bancos de dados orientado a objetos. Em geral, mineração pode ser classificada de acordo coma mineração de conhecimento dos diferentes tipos de bancos de dados: bancos relacionais, transacionais, orientados a objetos, dedutivos, espaciais, temporais, multimídia, heterogêneo, ativo, legado e baseado em Internet.

  Dados estruturados são os arquivos de banco de dados, as tabelas, ou seja, estruturas fixas com conteúdo uniforme. Dados desestruturados são arquivos do tipo texto ou imagem e podem ser usados em projetos que têm por objetivo a identificação de padrões ou formas.

- **Tipo de conhecimento a ser minerado**
  Diversos tipos típicos de conhecimento podem ser descobertos pelos mineradores,

incluindo regras de associação, classificação, clusterização (aglomeração), evolução e análise de desvio.

Os mineradores também podem ser classificados/categorizados de acordo com o nível de abstração de conhecimento descoberto, sendo conhecimento geral, conhecimento primitivo e múltiplo nível de conhecimento. Um minerador flexível deve descobrir conhecimentos em múltiplos níveis.

- **Tipo de técnica a ser utilizada**
  Mineradores também podem ser categorizados de acordo com as técnicas de mineração utilizadas. Por exemplo, podem ser técnicas que utilizam autonomia, dados, consultas ou iteratição.

  Podem ser classificados também segundo a abordagem: generalização, busca de similaridade, baseado em estatísticas ou teorias matemáticas, abordagens integradas, etc.

Dentre as diversas classificações possíveis, seguir-se-á o esquema de classificação do tipo de conhecimento a ser minerado porque esta classificação apresenta uma clara visão dos requisitos e técnicas de mineração.

Métodos para diferentes tipos de conhecimento incluem regras de associação, caracterização, classificação, clusterização, etc. Para um determinado tipo de conhecimento, diferentes abordagens pode ser utilizadas, tais como máquinas de aprendizado.

Capítulo 2

Técnicas Utilizadas

Mineração de dados é totalmente dependente da aplicação (ou negócio) em questão e diferentes aplicações necessitarão de diferentes técnicas de mineração. Em geral, os tipos de conhecimento que podem ser descobertos em um banco de dados são classificados como segue.

2.1 Regras de associação

As regras de associação *association rules* em bancos de dados transacionais e relacionais têm atraido muita atenção na comunidade de banco de dados. A tarefa é produzir regras fortes (satisfazendo valores de sustentação, confiança e precisão acima de valores estabelecidos como parâmetro) de associação da forma "$A_1 \bigwedge \ldots \bigwedge A_m \Rightarrow B_1 \bigwedge \ldots \bigwedge B_n$", onde A_1 ($i \in \{1, \ldots, m\}$) e B_j ($j \in \{1,\ldots,n\}$) são conjuntos de atributo-valor, dos dados relevantes do banco. Um modelo matemático foi proposto por [AIS93].

O problema de mineração de dados utilizando regras de associação é decomposto em duas fases:

1. Descobrir os grandes conjuntos de itens, ou seja, os conjuntos de itens que possuem transação que satisfaça um mínimo valor de sustentação (quantidade de dados que satisfazem algum critério) pré-definido.
2. Utilizar o conjunto encontrado para gerar as regras de associação para o banco de dados.

É importante minerar regras de associação em múltiplos níveis relativos ao conteúdo e contexto do banco e em um nível geral mais abstrato. Com relação à precisão do resultado obtido, podemos efetuar uma troca (balanceamento) entre precisão e eficiência a fim de controlar os resultados obtidos e o tempo de execução, respectivamente.

Como exemplo, podemos encontrar um grande conjunto de dados transacionais uma regra de associação o tipo: "se um consumidor compra leite, ele usualmente compra pão na mesma transação". Como as regras de associação requerem percorrer o banco diversas vezes, o total de processamento pode ser enorme, e o aproveitamento eficaz é uma preocupação essencial minerando tais regras.

Alguns algoritmos eficientes são o Apriori [AS94] e DHP [PCY95b], considerados iterativos. Para reduzir a quantidade de leitura que deve ser feita na base de dados a fim de melhorar a eficiência destes algoritmos, pode-se aplicar a técnica de *scan-reduction*.

Como os dados no banco de transações são inseridos, atualiados e apagados, deve-se manter uma atualização incremental a fim de manterem válidas as regras de associação encontradas em minerações anteriores [CHNW96].

Hoje em dia muitos bancos de dados transacionais encontram-se distribuídos pelo mundo. Então, efetuar mineração nestes bancos, a fim de obter uma decisão global a partir de conhecimentos parciais coletados em cada banco de dados pode tornar-se uma tarefa problemática, tanto do ponto de vista do colume de informações para analisar quanto da comunicação na rede. Para contornar este problema, pesquisas são feitas em algoritmos paralelos de mineração e um algoritmo proposto pode ser encontrado em [PCY95a].

2.2 Generalização e sumarização

Dados e objetos nos bancos de dados freqüentemente contém informações detalhadas em níveis conceituais muito baixos. Por exemplo, um item em uma relação de vendas pode conter atributos como nome, data_fabricação, preço, etc. No processo de mineração de dados, é desejável sumarizar esses atributos, por meio de generalizações e da utilização somente dos atributos relevantes.

Generalização de dados é um processo que abstrai um grande conjunto de dados relevantes de um banco de dados a partir de conceitos de baixo nível para outros de mais alto nível. O conhecimento essencial aplicado nessa técnica é o conceito de hierarquia, que podem ter relações semânticas fortes ou não. Por exemplo, o esquema endereço(número, rua, cidade, estado, país), pode gerar uma estrutura hierárquica da forma {cidade, estado, país} $\in$ {estado, país}, que possui uma relação semântica forte. Já o esquema item(id, nome, categoria, fabricante, data_fabricação, preço) pode gerar a estrutura hierárquica categoria, fabricante, data_fabricação $\in$ {categoria, fabricante}, que não possui nenhuma relação semântica aparente.

Os métodos para uma generalização eficiente e flexível podem ser categorizados em duas abordagens:

2.2.1 Cubo de dados

Também conhecido como bancos de dados multidimensionais, *views* materializadas e *OLAP (On-Line Analytical Processing)*, a idéia geral desta abordagem é prover *views* flexíveis a partir de diferentes pontos-de-vista e níveis de abstração. Para isso, certos cálculos (funções agregadas) com alto custo computacional e que são freqüentemente requisitados, como *count, sum, average, etc*, sâo materializados e computados de acordo com o agrupamento de diferentes conjuntos de atributos. Sendo assim, as dimensões do cubo podem representar atributos ou agrupamentos de atributos e as células representam valores daquele atributo ou agrupamento.

Por exemplo, uma relação com o esquema vendas(vendedor, comprador,data_venda, valor_compra) pode ser materializada em um cubo de dados com as dimensões: (vendedor), (comprador), (data_venda, vendedor), (data_venda, comprador) e as células representam a função sum(valor_compra). Possíveis cisões seriam ValorTotal (computado pelos valores das vendas) agrupado por (vendedor, fornecedor e comprador) e ValorTotalAgrupadoPor (vendedor)

Generalização e especialização podem ser realizadas num cubo de dados com as operações de *roll-up* e *drill-down*, onde *roll-up* reduz o número de dimensões em um cubo de dados e generaliza valores de atributos para níveis conceituais mais altos, e *drill-down* faz o contrário.

Os cubos de dados podem ser muito esparsos em muitos casos porque nem toda célula em cada dimensão pode ter um dado correspondente no banco de dados. Por isto, são necessárias técnicas para a manipulação de cubos esparsos, indexação e atualização incremental se fazem necessárias [FH01, Wid95, ZGMHW95].

O armazenamento dos resultados pré-computados em forma de cubo de dados assegura um tempo de resposta mais curto e visões de dados a partir de diferentes ângulos e em diferentes níveis de abstração.

A aplicação desta técnica é feita na área de KDD e apoio a tomadas de decisão.

2.2.2 Indução orientada a atributos

A indução orientada a atributos é uma técnica para generalização de qualquer subconjunto de dados em um banco de dados relacional, e extração de conhecimento a partir dos dados generalizados. Esses dados podem ser armazenados em um banco de dados, na forma de relações generalizadas ou um cubo generalizado, e ser atualizado incrementalmente.

É uma técnica para generalização de qualquer subconjunto de dados on-line num BD e extração de conhecimento desses dados, aplicada em BD´s relacionais, orientados a objetos e espaciais. Pode ser atualizada incrementalmente pela atualização do BD, porém não é apropriada para minerar padrões específicos em níveis conceituais primitivos. Pode ajudar a guiar a mineração por encontrar alguns traços em níveis conceituais mais altos

e aprofunda progressivamente a mineração para níveis mais baixos.

Os passos utilizados nesta técnica são:

- Obter o conjunto de dados relevantes do BD (parte de uma *query* de mineração de dados expressada de forma semelhante a uma em SQL);
- Realizar generalizações (remoção de atributos, escalada de árvores conceituais, controle de *threshold* de atributos);
- Armazenar os dados generalizados em BD (através de relações generalizadas ou cubos de dados);
- Realizar operações e transformações para obter diferentes tipos de conhecimento (operações *roll-up* e *drill-down* fornecem uma visão dos dados em diferentes níveis de abstração).

A relação de generalização é mapeada em em tabelas de sumarização, gráficos, curvas, projeções e análises para visualização e apresentação. A extração de regras discriminantes, que comparam diferentes classes de dados em múltiplos níveis de abstração podem ser extraídas agrupando-se o conjunto de dados de comparação em classes contrastantes antes de efetuar a generalização. Isto para os analistas e gerentes que necessitam não somente dos dados em sua forma normal, mas de informações em forma estratégica.

A extração de regras de caracterização, que sumarizam as características dos dados generalizados em diferentes níveis de abstração de acordo com um ou mais atributos de classificação pode ser feita aplicando classificadores de árvores de decisão na dado generalizado. A derivação de regras de associação, que associam um conjunto de propriedades de atributos generalizados em uma regra de implicação lógica, utilizando de métodos para mineração de regras de associação também pode ser realizada.

Na hierarquia de atributos, deve-se ter um conhecimento prévio essencial aplicado na indução orientada a objetos. Porém os dados atributos podem não estar fortemente ligados semanticamente.

O conceito essencial utilizado na indução é o conceito da hierarquia associado a cada atributo. Por exemplos, um conjunto de atributos Endereço (logradouro, número, bairro, cidade, estado, país) no esquema de banco de dados representa o conceito de hierarquia do atributo Endereço. Logo infere-se que {cidade, estado, país} $\subset$ {estado, país}. No esquema Item (id, nome, categoria, produtor, data_fabr, preço) temos que {categoria, produtor, data_fabr} $\subset$ {categoria, data_fabr}.

2.3 Classificação

Classificação de dados (*data classification*) consiste no processo de encontrar propriedades comuns e um determinado conjunto de objetos de um banco de dados e classificá-los em

diferentes classes, de acordo com um modelo de classificação. Para construir um modelo de classificação, um banco de dados de exemplo é definido como o conjunto de treinamento, onde cada tupla consiste em um conjunto de múltiplos atributos comuns das tuplas de um grande bancos de dados e, adicionalmente, cada tupla contém um rótulo marcado com a identificação de uma classe conhecida associada a ela. O objetivo da classificação de dados é primeiro analisar o conjunto de treinamento e desenvolver uma apurada descrição ou modelo para futuros testes com os dados de um grande banco de dados. Aplicações de classificação incluem diagnósticos médicos, predição de performance, vendas seletivas, etc.

Classificação de dados tem sido estudado substancialmente pela aproximção estatística (regressão e discriminante lineares), máquinas de aprendizado escalar (através da combinação dos classificadores de base) e redes neurais. Os passos básicos são:

- Definição de um conjunto de exemplos conhecidos (treinamento);
- Treinamento sobre esse conjunto;
- Gerar regras de classificação ou descrição.

Uma tarefa de classifição consiste em associar um item a uma classe dentre diversas opções pré-definidas. A tarefa do analista passa a ser de selecionar qual classe melhor representa cada registro. Por exemplo, ao se deparar com uma base de dados de veículos, em que cada registro contém os atributos de cor, peso, combustível, número de portas, cilindradas e número de marchas, classificar cada veículo em esporte, utilitário ou passeio.

Estimação
Uma variação do problema de classificação envolve o processo de geração de pontuações através das várias dimensões nos dados. Ao invés de empregar um classificador binário para determinar se um pretendente de empréstimo por exemplo, é um risco bom ou mau, esta aproximação gera um crédito de mérito *credit-worthiness* baseado em uma pontuação dada pelo conjunto de treinamento.

2.4 Clusterização

Instintivamente as pessoas visualizam os dados segmentados em grupos discretos, como por exemplo, tipos de plantas ou animais. Na criação desses grupos discretos pode-se notar a similaridade dos objetos em cada grupo.

Enquanto a análise de grupos é freqüentemente feita de modo manual em pequenos conjuntos de dados, para grandes conjuntos um processo automático de clusterização *(data clustering)* através da tecnologia de mineração de dados é mais eficiente. Em adição, os cenários existentes são muito similares, tornando-os competitivos, requerendo a utilização de algoritmos complexos que determinem a segmentação mais apropriada.

Nesse método de mineração, considerado do tipo "divisão e conquista", o algoritmo deve criar as classes através da produção de partições do banco de dados em conjuntos de tuplas. Essa partição é feita de modo que tuplas com valores de atributos semelhantes, ou seja, propriedades de interesse comuns, sejam reunidas dentro de uma mesma classe. Essas classe encontradas podem ser mutuamente exclusivas e exaustivas ou consistir de uma representação rica tal como uma hierarquia ou sobreposição de categorias.

Uma vez que as classes sejam criadas, pode-se aplicar um algoritmo de classificação nessas classes, produzindo assim regras para as mesmas. Um bom agrupamento caracteriza-se pela produção de segmentos de alta qualidade, onde a similaridade intra-classe é alta e a inter-classe é baixa. A qualidade do resultado da clustrerização também depende da medida utilizada para medir a similaridade usada pelo método e de sua implementação, além de sua habilidade de descobrir algum ou todos os padrões escondidos.

As técnicas mais utilizadas para agrupar dados são baseadas em três categorias:

- Partição, basicamente enumera várias partições e então cria uma nota para cada uma delas segundo algum critério, ex: k-means (explicado logo abaixo).
- Hierarquia, cria uma decomposição hierárquica do conjunto de dados usando algum critério;
- Modelo, um modelo é hipoteticamente criado para cada cluster e a idéia é encontrar o que melhor se enquadra quando comparados entre si.

A maioria das ferramentas de clusterização trabalham em função de um número pré-definido de grupos especificado por um usuário. Isso requer um conhecimento detalhado do domínio, transformando assim a tarefa de descoberta de conhecimento menos atrativa. Tecnologias mais sofisticadas são capazes de procurar através de diferentes possibilidades de quantidades de grupos e avaliar cada configuração de acordo com a sua importância.

Técnicas baseadas em redes neurais auto organizáveis utilizando algoritmos Kohonen também são capazes de segmentar grupos de dados.

Tradicionalmente as técnicas de clusterização são divididas em **hierárquicas** e de **partição** [Ber02]. Clusterização hierárquica ainda pode ser subdividida em **aglomerativa** e **divisiva**. A base de clusterização hierárquica conta com a fórmula de cluster conceitual, algoritmos clássicos como SLINK, COBWEB, e os mais recentes algoritmos CURE [GRS98, HKT] and CHAMELEON [HKT].

Enquanto clusterização hierárquica constrói *clusters* de forma gradual, os algoritmos de particionamento os inferem diretamente. Com isso, eles tentam descobrir novos *clusters* realocando iterativamente pontos entre os subconjuntos, ou tentam identificar áreas densamente povoadas com dados. As técnicas de particionamento podem ser subdivididas em clusterização probabilística (*EM (Expectation Maximization) framework* [HKT], algoritmos SNOB, AUTOCLASS, MCLUST), métodos k-medoids [HKT] (algoritmos PAM, CLARA, CLARANS [RTN02], e suas extensões), e métodos k-means (esquemas diferentes,

inicialização, otimização, média harmônica e extensões). Estes métodos concentram em como os pontos encaixam-se em seus respectivos *clusters* e tentam construir *clusters* de formatos propriamente convexos. Dois algoritmos [HKT] que foram os precursores desta técnica foram AGNES (AGlomerative Nesting) e DIANA (DIvisia ANAlysis). O primeiro é um algoritmo do tipo *bottom-up* que inicia colocando todos os objetos em seu devido *cluster* e então inicia a fase de agrupamento destes *clusters* atômicos em maiores, e maiores, até que os objetos estejam em um único *cluster*, ou até que certa condição de término seja satisfeita. O segundo algoritmo adota uma abordagem *top-down*, que faz exatamente o reverso de AGNES: inicia com todos os objetos em um único *cluster* e subdivide-o em menores até cada objeto formar um *cluster* atômico ou satisfação certa condição de parada.

Algoritmos de particionamento tentam descobrir densos componentes de dados, flexíveis em termos de sua forma. Os algortimso utilizados são DBSCAN [EKSX96], OPTICS [HKT], DBCLASD, enquanto DENCLUE [HKT] explora as funções de densidade espaciais. Estes algoritmos são menos influenciáveis pelos ruídos[1] e descobrem *clusters* de formato irregular. usualmente trabalham com dados de baixo nível, de atributos numéricos, conhecidos como dados espaciais. Objetos espaciais incluem não somente pontos como objetos extensos (algoritmo GDBSCAN).

Alguns algoritmos trabalham indiretamente com os dados, construindo sumarizações de dados através do subconjunto do espaço de atributos. Efetuam segmentação do espaço, e depois agregam estes segmentos. São conhecidos como *Grid-Based Methods* [HKT]. Frequentemente eles utilizam aglomeração hierárquica como uma das fases de processamento. São os algoritmos BANG, STING [HKT], WaveCluster [HKT] e idéias de fractais. Estes algoritmos são muito rápidos e lidam bem com ruídos. A tecnologia *Grid-Based* é um passo intermediário em muitos outros algoritmos (por exemplo, CLIQUE [HKT], MAFIA).

Dados categóricos estão intimamente ligados com bancos de dados transacionais. O conceito de similaridade por si só não é suficiente para clusterização de dados. A idéia de co-ocorrência categorical dos dados vem para ajudar. Os algoritmos encontrados para este fim são ROCK, SNN, e CACTUS. A situação começa a complicar quando aumenta-se o número de itens envolvidos. Para contornar isto, um esforço é adicionado na tarefa de clusterização para uma pré-clusterização dos itens ou nos valores categóricos dos atributos. O desenvolvimento baseado no particionamento *hyper-graph* e o algoritmo STIRR exemplificam esta abordagem.

Muitas outras técnicas foram desenvolvidas, principalmente em máquinas de aprendizado, que possuem significado teórico, são usadas tradicionalmente fora da comunidade de mineração de dados, ou não encaixam-se nas categorias previamente abordadas. Este limite é errôneo devido às novidades em aprendizado supervisionado, *gradient descent* e a ANN (LKMA, SOM), métodos evolucionários (simulação enrigessida *(simulated annealing)*, algoritmos genéticos e o algoritmo AMIBA.

[1] Nesta seção vamos utilizar o termo ruído para denotar pontos ignorados pois afetam negativamente a análise de clusterização

Porém um dos problemas, são as restrições do mundo real impostas aos sistemas de mineração [Ber02]. Primeiramente, a mineração é efetuada em enormes bancos de dados, consequentemente, clusterizar grandes conjuntos de dados apresenta o problema de escalabilidade e estensões VLDB *(Very Large DataBases)*, onde encontra-se como solução algortimos como DIGNET, BIRCH [ZRL96] e outras técnicas de compressão de dados, até os limites de Hoffding ou Chernoff.

Outro problema são as numerosas dimensões. O problema vem de uma diminuição na separação métrica quando a dimensão cresce. Uma abordagem para redução da dimensionalidade utiliza transformações de atributos (DFT, PCA, *wavelets*). Uma outra maneira de enfrentar o problema é utilizar a aglomeração de subespaços (algoritmos CLIQUE, MAFIA, ENCLUS, OPTIGRID, PROCLUS, ORCLUS). Outra abordagem clusteriza atributos em grupos e usa um substituto para clusterizar os objetos. Esta técnica de dupla clusterização é conhecida como co-clusterização.

2.4.1 Técnica K-means

Uma das técnicas existentes de clusterização é o k-means, onde k seria o número de classes no qual deseja-se agrupar. Como o pesquisador não tem como saber um número ótimo para k, faz-se necessário a execução dessa técnica várias vezes, parando quando perceber-se que as classes são as desejadas.

Esta técnica consiste basicamente de 4 passos principais:

1. Partição dos objetos em k grupos não vazios;
2. Defina os centróides dos grupos da partição atual;
3. Para cada objeto, inclua-o ao grupo cujo centróide esteja mais próxima;
4. Retorne ao passo 2 enquanto houver objetos que estejam em grupos incorretos.

Esses passos podem ser observados na figura abaixo:

- Pontos Fortes: Relativamente eficiente: $O(tkn)$, onde n é número de objetos, k é número de grupos, e t é número de iterações. Normalmente, k, $t << n$.

  Freqüentemente termina em um ótimo local. O ótimo global pode ser encontrado usando técnicas tais como: *deterministic annealing* e algoritmos genéticos.

- Pontos Fracos: É necessário especificar a priori k, o número de grupos. É sensível a ruídos e valores aberrantes. Não é apropriado para a descoberta de grupos não esféricos.

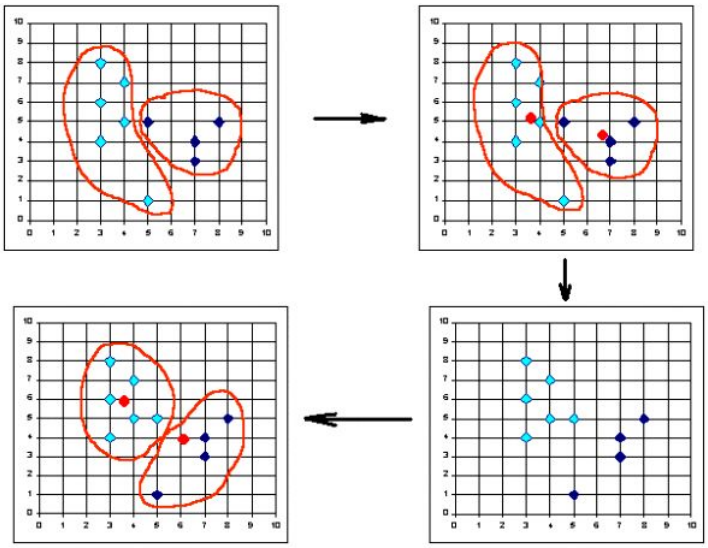

Figura 2.1: Ilustração da técnica K-mean

2.5 Busca baseada em reconhecimento de padrões

Quando se busca por padrões similares em bancos de dados temporal ou espacial-temporal, duas formas de perguntas são comumente encontradas em varias operações de mineração de dados, principalmente em bancos de dados temporais:

- **Pergunta por similaridade com um objeto de referência:**
  a busca é feita sobre uma coleção de objetos para encontrar os que possuem uma distância (similaridade) dentre a definida pelo usuário do objeto de referencia. É conhecida como *Object-relative similarity query*;
- **Pergunta por todos os pares similares:**
  O objetivo dessa busca é encontrar todos os pares de elementos que estão dentre a distância definida pelo usuário entre eles. É conhecida também como *All-pair similarity query*

A medida de similaridade é totalmente dependente do domínio dos dados: imagens, áudio, texto, hipertexto, mapas geográficos, etc.

Avanços significativos foram feitos tanto em busca de seqüência *sequence matching* em bancos de dados temporais quanto no reconhecimento de diálogo como *dynamic time warping*.

2.6 Busca por padrões de caminhos em ambiente "linkado"

Em um ambiente que possui informações distribuídas, como páginas de internet, por exemplo, os documentos e objetos são "linkados", principalmente através de *hyperlinks* e endereços URL, para facilitar a navegação do usuário. Exemplos de ambientes e aplicação desta técnica são sites de internet como a Amazon, America On Line, Submarino, etc. Entender os padrões de acesso dos usuários em sistemas como esses não só possibilita uma melhor projeção desses sistemas, mas também conduz a melhores decisões de marketing (por exemplo, colocar banner em locais adequados baseado em análises de comportamento do usuário).

A captura dos padrões de acesso dos usuários em alguns ambientes é conhecido como *Mining Path Traversal Patterns*. É uma área de pesquisa bem recente e por isso diz-se que está em sua infância. Isso pode ser explicado pelo fato de grande parte da análises relacionadas aos padrões de acesso dos usuários já realizadas se preocuparem com questões como freqüência de visita em determinada página ou dados pessoais dos usuários como idade ou ocupação, ao invés de tentar fazer uma análise sobre os caminhos percorridos durante a navegação.

Para o problema do reconhecimento de padrões de acesso foi explorado em foi proposta uma solução que consiste em dois passos.

- Primeiro, um algoritmo tem por objetivo converter as seqüências de acessos de um usuário em um conjunto de subseqüências. É interessante notar que esse primeiro passo faz uma filtragem dos dados, eliminando os dados originados do retorno a uma página anterior já acessada durante a navegação. Isso permite que os estudos sejam feitos sobre seqüências significantes de acesso.
- Em um segundo passo, algoritmos são usados para obter o conjunto de percursos máximos, sendo que cada percurso máximo consiste no último objeto ou página atingido através de um caminho a partir do ponto inicial de acesso (no caso de um site, a página principal). Deve-se escolher os percursos máximos que ocorrem um número significativo de vezes.

Por exemplo, suponha a seqüência de acessos da figura 10.1 para um determinado usuário: A, B, C, D, C, B, E, G, H, G, W, A, O, U, O, V. O conjunto de percursos máximos é dado por: ABCD, ABEGH, ABEGW, AOU, AOV.

Depois que os conjuntos de percursos máximos são obtidos para todos os usuários, deve-se obter o conjunto de seqüências máximas. Uma seqüência máxima é aquela que não está contida em nenhuma outra seqüência máxima. Isso é feito de uma maneira direta. Suponha, por exemplo, que AB, BE, AD, CG, GH, BG seja um conjunto de percursos máximos de tamanho 2 e que ABE, CGH seja um conjunto de percursos máximos de tamanho 3. Então o conjunto de seqüências máximas é dado por AD, BG, ABE e CGH. Uma seqüência máxima corresponde a um caminho freqüentemente acessado.

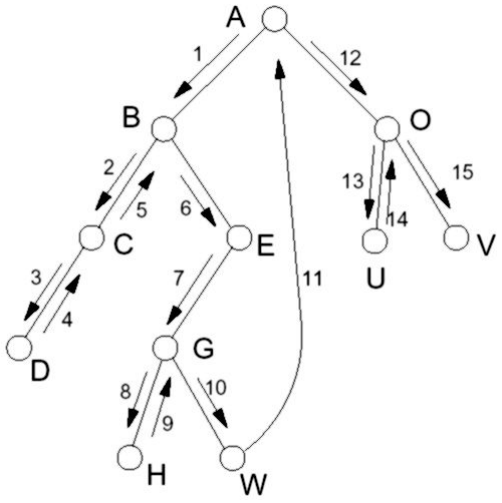

Figura 2.2: Exemplo ilustrativo de busca por padrões de caminhos em ambiente "linkado"

Poderíamos fazer uma analogia entre esse problema de encontrar o percurso máximo em um caminho e o de encontrar um itemset em uma regra de associação que apareça um suficiente número de vezes em uma transação. Entretanto, existe uma diferença entre os dois pelos fato do primeiro levar em consideração o caminho percorrido, ao passo que no segundo caso apenas é feita uma análise sobre uma combinação de itens em uma transação, não existindo um link entre dois itemsets. Assim, os algoritmos dos dois casos são diferentes.

2.7 Regressão

Esta tarefa é conceitualmente similar a tarefa de classificação. A maior diferença é que nessa tarefa o atributo meta, ou objetivo, é contínuo, isto é, pode tomar qualquer valor real ou qualquer número inteiro num intervalo arbitrário, ao invés de um valor discreto.

A regressão utiliza valores existentes para predizer o que outros valores de itens poderão assumir [Cro99]. Em um caso simples, regressão faz o uso de técnicas padrões de estatística tais como regressão linear. Infelizmente, muitos problemas do mundo real não são simples projeções lineares de valores encontrados anteriormente. Por exemplo, volume de vendas, *stock prices* e taxa de falha nos produtos são difíceis de predizer porque podem depender de complexas iterações de múltiplas variáveis. Conseqüentemente, técnicas mais complexas (por exemplo, regressão logística, árvores de decisão ou redes neurais) podem ser necessárias para prever os valores futuros.

Os mesmos tipos de modelo podem frequentemente ser usados para a regressão e a classificação. Por exemplo, o algoritmo de árvore da decisão CART *(Classification And Regression Trees)* pode ser usado para construir árvores de classificação (para classificar variáveis categóricas da resposta) e árvores da regressão (para prever variáveis contínuas

da resposta). As redes neurais podem criar modelos de classificação e de regressão.

2.8 Árvores de decisão

As árvores da decisão *desicion trees* são uma maneira de representar uma série de regras que conduzem a uma classe ou a um valor. Por exemplo, pode desejar-se classificar pretendentes do empréstimo como riscos de crédito bons ou maus. A figura abaixo [2] mostra uma árvore simplificada da decisão que resolva este problema ao ilustrar todos os componentes básicos de uma árvore da decisão: o nó da decisão, ramificações e folhas.

Figura 2.3: Uma árvore de decisão bem simples

O primeiro componente no topo de uma árvore de decisão, ou a raiz, especifica o teste a ser efetuado. O nó raiz do exemplo é "Income > $40,000". O resultado deste teste divide a árvore em duas ramificações, cada uma representando uma ou mais respostas possíveis. Neste caso, o teste "Income > $40,000" pode ser respondido tanto como "yes" ou "no" e então descemos para uma destas duas ramificações.

Dependendo do algoritmo, cada nó pode ter duas ou mais ramificações. Por exemplo, CART gera árvores de dedecisão com apenas duas ramificações por nó. Tal árvore é chamada de árvore binária. Quando mais de duas ramificações são permitidas, a árvore é chamada de multi-nível.

Os algoritmos mais utilizados na técnica de árvores de decisão são: CART, CHAID, C4.5, C5.0, Quest, etc. Eles diferem: no *binning requirements*, suporte a árvores binárias ou de múltiplas folhas e diferenças na seleção da próxima variável a ser escolhida para ramificação.

2.9 Algoritmos genéticos

Os algoritmos genéticos não são usados encontrar padrões por si só, mas para guiar o processo de aprendizagem de algoritmos mineradores de dados como redes neurais [Cro99].

[2] Exemplo extraído de [Cro99]

Essencialmente, estes algoritmos agem como um método para executar uma busca guiada de modelos adequados ao espaço da solução.

São chamados algoritmos genéticos porque seguem fracamente o padrão da evolução biológica em que os membros de uma geração (dos modelos) competem para passar suas características à geração seguinte, até que o melhor modelo seja encontrado. A informação a ser passada é contida em cromossomos que contêm os parâmetros para construir os próximos modelos.

Por exemplo, ao construir uma rede neural, os algoritmos genéticos podem substituir o *backpropagation* como uma maneira ajustar os pesos. O cromossomo, neste caso, conteria os pesos. Alternativamente, estes algoritmos podem ser usados para encontrar a melhor arquitetura de um sistema, e os cromossomos conteriam o número de camadas escondidas e o número dos nós em cada camada.

Enquando os algoritmos genéticos são uma bordagem interessante para otimização de modelos, eles adicionam um custo muito elevado de recusos computacionais.

2.10 Redes neurais

2.11 Outras técnicas de mineração

Existem outras técnicas de mineração que menos destacadas, são elas:

- Descoberta de regras através da semântica da otimização de *queries* - Esta tarefa transforma uma *query* do banco de dados em uma outra utilizando a semântica do conhecimento da base, tais como restrições de integridade e dependências funcionais para produzir uma *query* mais eficiente.
- Descoberta de dependências do banco de dados - No modelo de dados relacional, as definições das relações na base de dados não dizem nada sobre o relacionamento entre seus atributos. Esses relacionamentos são especificados através das dependências dos dados, ou das restrições de integridade, essa tarefa busca automaticamente descobrir tais dependências.

  Abordagens adicionais são utilizas em conjunto com as técnicas descritas anteriormente e com outras técnicas incluindo *case-based reasoning*, lógica fuzzy, algoritmos genéticos e transformações baseadas em fractais. Esta última é uma ferramemta relativamente recente na análise de dados e interessante devido à sua agressividade na compressão de dados sem perda *(lossless)*. A partir deste fato, há a possibilidade que o reconhecimento de padrões baseado nestas técnicas poderia explorar tamanhos substancialmente reduzidos de dados a fim de aumentar o desempenho. Cada uma destas técnicas possui sua própria força, fraqueza em termos dos problemas que devem ser enfrentados, potenciais problemas, performance e requisitos de treinamento.

Os algoritmos são freqüentemente ajustáveis usando-se uma variedade de parâmetros visados a fornecer o contrapeso adequado à precisão e ao desempenho.

Capítulo 3

Mineração de dados e *Data WareHouse*

Existe uma relação simbólica entre a atividade de mineração e *data warehouse (DW)*. Os DW organizam os dados para um efetivo processo de mineração, porém, a prospecção de dados pode ser aplicada onde não exista nenhum DW, mas este aumenta em muito as chances do sucesso do processo de prospecção. Cada uma das características dos DW, que incluem dados integrados, dados detalhados e resumidos, dados históricos e metadados, melhoram o desempenho e o resultado da prospecção do processo de mineração de dados.

Dados integrados permitem ao minerador visualizar de forma rápida e fácil os dados. Na ausência de integração entre os dados, o agente minerador (técnico humano) necessitaria de uma grande quantidade de tempo para condicionar e refinar os dados antes do processo de mineração. Chaves precisariam ser reconstituídas, dados codificados necessitariam de revisão e estruturas de dados deveriam ser padronizadas. Os DW são integrados e têm todas essas tarefas e muitas outras já realizadas e portanto, o agente minerador pode concentrar-se integralmente nos algoritmos de mineração.

Dados detalhados são necessários quando o agente minerador deseja examinar os dados de forma mais granular. Algumas vezes o nível de exploração dos dados requer a análise cuidadosa destes dados, mas geralmente os dados resumidos asseguram que uma prévia já foi feita e evitam muito processamento desnecessário e repetitivo. Dados históricos são importantes porque grande quantidade de informações fica implicitamente armazenada. Trabalhar somente com informações atuais pode impedir que se detecte tendências e padrões de comportamento ao longo do tempo. Informações históricas são cruciais para se entender o condicionamento dos negócios. Metadados ajudam a descrever não só o conteúdo dos dados, mas o contexto das informações. à medida em que a informação passa a ser examinada, o contexto passa a ser mais importante do que o conteúdo, revelando explicações a respeito do significado dos dados.

Esta importante relação entre mineração e *data warehouse* que, utilizados em conjunto, maximizam os resultados do processo de mineração de dados [Kim97].

Capítulo 4

Exemplos práticos de mineração de dados

4.1 Supermercado

Considerando um exemplo (que é muito clássico em mineração de dados) de um supermercado que utiliza *scanners* de código de barras no caixa de compras. O sistema de computadores é quem identifica o nome e preço do produto e atualiza a lista de estoque. Com isso, existe uma base para o gerente ordenar o reabastecimento das prateleiras. Geralmente esse é um dos únicos propositos a que se prestam esses dados e após algum tempo eles serão descartados. Entretanto, esses conjuntos de dados possuem muitas informações valiosas que podem ser utilizadas para outros propósitos além daquele para o qual foram originalmente coletados.

Estas informações podem ser usadas para se providenciar resumos de vendas, estudar as preferências dos clientes, conhecer quais itens ou combinações de itens devem ser colocados a venda ou simplesmente para adquirir diversos tipos de informações de marketing.

Nesse contexto, a aplicação de um sistema de mineração de dados pode apontar modelos tais como:

- Quais itens são freqüentemente comprados em combinação (por exemplo, cereais e leite; mostarda, pão e salsicha; fralda e comida para recem-nascido)?
- Quais itens são freqüentemente adquiridos numa compra em torno de R$ 100,00?
- Quais itens são freqüentemente comprados por famílias (uma família pode ser identificada através dos tipos de certos produtos que são tipicamente adquiridos por crianças)?
- Quais itens são freqüentemente comprados por pessoas fazendo pequenas compras?

Obviamente, correlações como uma relação entre compradores de fraldas e comida para bebê bem menos interessante, do ponto de vista da descoberta de conhecimento, do que uma correlação entre produtos derivados do leite e anti-ácidos. Modelos que sejam realmente interessantes normalmente se preocupam com relações que sejam totalmente inesperadas.

4.2 Lojas Brasileiras

Até meados de 1997, a rede varejista Lojas Brasileiras sofria com a dificuldade de dispor em suas prateleiras cerca de 51.000 produtos que mantinha em seu catálogo segundo [Com97]. O problema era meramente de espaco físico em suas lojas. Depois de um processo de automação que teve um custo de aproximadamente um milhão de dólares, a cadeia de lojas, que contava na época com setenta lojas espalhadas por todo o Brasil, descobriu que muitas dessas mercadorias não rendiam quase nenhum retorno em vendas. Entre os itens de pouca venda estavam guarda-chuvas, sombrinhas e malhas de lã. O motivo, descoberto mais tarde, era que tais produtos se encontravam expostos em lojas do nordeste, onde chuva e frio são raros. Outra descoberta foi o fato de estarem sendo vendidas batedeiras com voltagem de 110 Volts em Santa Catarina e no Rio Grande do Sul, onde a voltagem padrão é de 220 Volts.

Nos dias atuais, segundo informações [Com97], o grupo mantém 14.000 itens em exposição nas lojas. Em uma única operação, foram eliminados 37.000 produtos. Seus executivos utilizaram a mineração de dados para conseguirem estes resultados. Com base em relatórios a respeito dos hábitos de consumo dos clientes, seus hobbies e informações sobre suas transações comerciais e financeiras foi possível traçar associações que revelaram grandes nichos de mercado. Em conjunto foi utilizado um banco de dados baseado em *data warehouse*, modelado sobre as informações transacionais do conjunto das lojas da rede.

4.3 Wal-Mart

O caso mais divulgado pela mídia de utilização de mineração de dados por uma empresa é o da cadeia americana de supermercados Wal-Mart. Seus executivos identificaram um hábito curioso dos consumidores: ao procurar eventuais relações entre o volume de vendas e os dias da semana, o processo de mineração de dados identificou que, nas sextas-feiras, as vendas de cervejas cresciam na mesma proporção que as de fraldas. Uma investigação detalhada revelou que, ao comprar fraldas para seus bebês, os pais aproveitavam para abastecer o estoque de cerveja para o final de semana.

4.4 Bank of America

A detecção de fraudes é uma das aplicações mais visadas pelos gerentes que procuram por soluções em mineração de dados. Diversos bancos recorrem a esse recurso para avaliar a credibilidade de seus clientes. Perfis são traçados com o intuito de oferecer facilidades e serviços a clientes com maior possibilidade de retorno, além de aplicar limites de aplicações para clientes considerados negligentes.

O Bank of America utilizou essas técnicas para selecionar entre seus trinta e seis milhões de clientes aqueles com menor risco de não pagarem um empréstimo. A partir desses relatórios, foi possível criar uma campanha de marketing oferecendo linhas de crédito para os correntistas cujos filhos tivessem entre 18 e 21 anos e, portanto, se interessassem por um empréstimo em dinheiro para auxiliar os filhos na compra de um automóvel, uma casa própria ou arcar com os gastos da faculdade. Em menos de três anos o banco obteve um lucro de trinta milhões de dólares tendo um número muito abaixo do normal de clientes com problemas no acerto do empréstimo.

4.5 Banco Itau

O banco Itau é uma empresa pioneira no Brasil no uso de mineração de dados. Era comum o envio de mais de um milhão de malas diretas via correio para todos os correntistas com ofertas de serviços dos mais diversos. A correspondência era direcionada a todos os correntistas, mas no máximo 2% deles respondiam às promoções.

Hoje o banco mantém informações sobre toda a movimentação financeira de mais de três milhões de clientes e, através da mineração dessa base de dados, é possível que as cartas sejam direcionadas apenas àqueles clientes que demonstram maior chance de responder a oferta. A taxa de retorno aumentou de 2% para 30% e houve uma economia de aproximadamente 80% nas despesas com serviços de correio.

4.6 Outros exemplos

Uma rede varejista americana descobriu, com base na mineração dos dados contidos em sua informação coletada em um armazém de dados, que a venda de colírios sofre um notável aumento nas vésperas de feriados. O motivo dessa procura ainda e ignorado, mas a informação foi mais tarde comprovada através das vendas nos feriados seguintes. As lojas passaram a contar com estoques extra do produto, preparando-se para os feriados.

Capítulo 5

Softwares comerciais de apoio

Como o processo de mineração de dados é muito custoso, devidos a diversos fatores como a grande dimensão dos bancos de dados a serem minerados, disponibilidade do resultado apresentado, atualização deste resultado devido a alterações na base de dados, e outros descritos anteriormente, deve-se fazer uso de sofwares de apoio à tarefa de mineração de dados.

Podemos categorizar os softwares de apoio em duas classes:

- Pacotes pertencentes ao *RDBMS (Relational Database Management Systems)*
- Pacotes específicos

5.1 Pacotes integrados ao *RDBMS*

Um bom ponto de partida para a mineração de dados pode ser tão próximo quanto o próprio sistema de gerenciamento de banco de dados utilizado: a maioria dos fabricantes destes sistemas comerciais reconheceram o crescimento e a necessidade de aplicações de mineração e empacotaram junto ao sistema, ferramentas de mineração.

Abaixo tem-se a lista de alguns produtos de conhecidos fabricantes de sistemas gerenciadores de bancos de dados e respectivas ferramentas de apoio à mineração de dados:

5.1.1 Oracle 9i: Darwin

Utiliza redes neurais, classificação e regressão em àrvores de decisão, algoritmos de clusterização, entre outras;

5.1.2 DB2 - IBM: Intelligent Miner For Data

Inclui suporte a aplicações de mineração como classificação, regras de associação, seqüências e clusterização.

5.1.3 SQL Server 2000 - Microsoft

Provê suporte a duas classes de aplicação de mineração de dados: árvores de decisão e clusterização.

5.2 Pacotes específicos

Pacotes que não se encontram dentro do pacote de gerenciamento do sistema de banco de dados são encontrados em grande quantidade e variedade hoje em dia.

Uma lista completa destes pacotes pode ser encontrada em http://www.kdnuggets.com/software/suites.html. Em particular, serão descritos dois sofwares, a título de exemplificação:

MineSet: produzido pela empresa *Sillicon Graphics*;

WizRule: produzido pela empresa *Wiz Soft*.

Estes softwares auxiliam a geração o de regras de dependência entre os dados e, principalmente o *MineSet*, na visualizaçãoo gráfica de conjuntos de dados.

5.2.1 *MineSet*

O software MineSet é formado por um conjunto de ferramentas integradas, que permitem a realização de mineração e visualização de dados contidos em um banco de dados ou arquivos de texto com um formato específico. Essas ferramentas aplicam as técnicas de data mining para "garimpar" dados e mostrar os resultados de forma gráfica, de tal forma que permita ao usuário uma melhor visualização, compreensão e com isso descoberta de informações ocultas contidas nestes dados. Este software foi desenvolvido pela empresa americana *Silicon Graphics* e adquirido pela empresa *Purple Insight* e está atualmente na versão 3.1. Maiores informações podem ser obtidas em [Ins03].

5.2.2 *WizRule*

O *WizRule* é um software de auditoria, descrição e limpeza de dados que, de forma automática, revela todas as regras que modelam a base de dados e indica os casos de desvio encontrados com relação ao conjunto de regras geradas. Criado pela empresa *WizSoft*, pode ser adquirido através de download a partir do site da empresa [Wiz00].

O programa gera relatórios que descrevem a base de dados através de regras, dentre elas, regras do tipo se A então B, regras matemáticas e erros ortográficos de nomes e valores. Pode também calcular o nível de incerteza de cada desvio evitando assim os casos em que um registro é considerado um desvio a regra.

Capítulo 6

Trabalhos e direções futuras

Atualmente as pesquisas e desafios das aplicações em KDD e mineração de dados incluem:

6.1 Grandes conjuntos de dados e alta dimensionalidade

Bancos de dados de vários gigabytes a terabytes com milhões de registros e grande número de campos são comuns. Estes bancos de dados criam grandes espaços durante a busca e aumentam as chances do algoritmo de mineração encontrar um modelo que não seja genericamente válido. Possíveis soluções incluem algoritmos muito eficientes, amostragem, metodos de aproximação, técnicas de redução da dimensionalidade e incorporação de conhecimentos adquiridos anteriormente.

6.2 Interação com o usuário e conhecimento adquirido

Um analista normalmente não é um especialista em KDD mas uma pessoa responsável por perceber o sentido nos dados usando as técnicas de KDD. Sendo o KDD por definição interativo e iterativo, é um desafio criar com alta performance, um ambiente de resposta rápida que também ajude os usuários na seleção formal das ferramentas apropriadas e técnicas para alcancar seus objetivos. Há a necessidade de uma maior ênfase na interação homem-computador e menor ênfase na automação total - com o objetivo de dar suporte a ambos, especialistas e usuários novatos. Muitos dos atuais métodos e ferramentas de KDD não são realmente interativos e não incorporam facilmente os conhecimentos previamente adquiridos sobre o problema estudado, exceto em casos simples. O uso do domínio do conhecimento é importante em todos os passos do processo de KDD.

6.3 Dados perdidos

Este problema é especialmente encontrado em bancos de dados de negócios. Atributos importantes podem ser perdidos se a base de dados não foi criada tendo em vista a possível descoberta de conhecimento. Dados perdidos podem resultar de erros do operador, sistemas reais (dados não preenchidos pelo vendedor durante uma transação) e falhas de medidas, ou de uma revisão do processo de aquisição de dados ao longo do tempo, como por exemplo, novas variáveis são incluídas, mas eram consideradas sem importância poucos meses atrás. Possíveis soluções incluem estratégias sofisticadas de estatística para identificar variáveis escondidas e dependências.

6.4 Gerenciamento de mudanças de variáveis e conhecimento

Rápidas mudanças nos dados podem fazer conhecimentos previamente adquiridos tornarem-se inúteis. Além disso, as variáveis medidas numa determinada aplicação de banco de dados podem ser modificadas, apagadas, ou aumentadas com novas medidas ao longo do tempo. Possíveis soluções incluem métodos incrementais para atualizar os modelos e tratar mudancas como uma oportunidade de descoberta, usando isto como uma sugestão para procurar por novos modelos de mudanca.

6.5 Integração

Um sistema de descoberta isolado pode não ser muito útil. Integrações típicas incluem integração o com um Sistema de Gerenciamento de Banco de Dados - SGBD (por exemplo, via uma interface de consulta), integração com planilhas eletrônicas e ferramentas de visualização. Ambientes de forte integração homem-computador como esboçado pelo processo de KDD permitem tanto a descoberta humana assistida por computador como a descoberta computacional assistida por humanos. O desenvolvimento de ferramentas para visualização, interpretação e análise de modelos descobertos é de extrema importância. Tal ambiente interativo pode habilitar soluções práticas para vários problemas da vida real com um custo de tempo mais acessível comparado aos resultados obtidos por humanos ou computadores operando individualmente. Existe uma oportunidade potencial e um desafio em desenvolver técnicas para integrar as ferramentas OLAP da comunidade de base de dados e as ferramentas de mineração das comunidades de aprendizado de máquina e estatística.

6.6 Multimídia e dados orientados a objetos

Uma significante tendência é que os bancos de dados contenham não somente números, mas grande quantidade de dados incomuns (não-numericos, não-textuais, geométricos e

gráficos) e dados multimídia (texto falado de forma livre e imagens digitalizadas, vídeo e dados de áudio). Esses tipos de dados ainda estão em grande parte além do escopo da atual tecnologia de KDD.

Capítulo 7

Conclusão

As tecnologias para armazenamento de informação são tão comuns quanto numerosas. Junta-se a isso, a vontade dos empreendedores de extrair o máximo de vantagem de suas informações. Esses elementos tornam a mineração de dados e a busca de conhecimento a partir de banco de dados uma área de conhecimento em crescente expansão nos dias de hoje. Será raro, em um futuro próximo, uma empresa ou organização que não invista nas tecnologias do conhecimento. A busca pelo conhecimento nunca foi fácil. A utilização de equipamentos, computadores e de técnicas avançadas de inteligência artificial não substituem as habilidades abstratas humanas na interpretação de qualquer tipo de informação. Os softwares de mineração de dados auxiliam em muito e minimizam o trabalho exaustivo do homem na análise de imensas quantidades de dados, tornando a informacé o mais clara e a busca pelo conhecimento mais fácil.

No campo de apoio à decisão, a mineração de dados utilizada de forma consciente em conjunto com os gerentes e engenheiros da informação resulta em vasta gama de novos e totalmente inesperados conhecimentos que não seriam de forma alguma localizados por qualquer uma das partes isoladamente. Gerentes têm o conhecimento sobre o dia-a-dia e o universo de suas atividades, mas não e viável que analisem toda a informação colhida em suas atividades. Engenheiros da informação costumam trabalhar dados e transformá-los da forma que lhes convêm para torná-los mais amistosos. A mineração de dados realiza o trabalho pesado e exaustivo que seria impossível a qualquer agente humano ou ao menos não poderia ser realizado em tempo hábil. Pode-se dizer com relativa confiança que é fácil começar um projeto de mineração de dados, a dificuldade está na finalização de acordo com as expectativas. As promessas geradas, no início de um projeto, pela utilização de novas tecnologias que podem resolver problemas tradicionalmente difíceis, podem ser mal interpretadas ao se avaliar a expectativa de um novo projeto.

Dificuldades com a extração, preparação e validação dos dados extraídos e a alocação de recursos no cliente, freqüentemente são subestimadas durante o planejamento dos cronogramas para a execução dos projetos. As atividades de obtenção e limpeza dos dados geralmente consomem mais da metade do tempo dedicado ao trabalho.

Este problema multi-disciplinar, que envolve diversas áreas como banco de dados, inteligência artificial, estatística, computação gráfica e outras mais, é responsável hoje em dia pela redução de custos, otimização de recursos (tempo, dinheiro, pessoal), suporte a tomadas de decisões, melhorias na relação empresa/consumidor, etc.

Com relação à proposta inicial da monografia, entregue anteriormente, algumas alterações foram realizadas:

- Os pontos referentes à descrição dos algoritmos CLARANS [RTN02], BIRCH [ZRL96], DBSCAN [EKSX96] e CURE [GRS98] não foram abordados pois verificou-se a existência de uma enorme quantidade de algoritmos de mineração e por isso seria injusto citar apenas quatro, dentre as dezenas existentes. Assim, manteve-se o escopo de um trabalho mais geral.
- Parte da bibliografia inicial não foi utilizada neste trabalho devido a impossibilidade de acessar/comprar o material [HK00, Dun02].

Referências Bibliográficas

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 26–28 1993.

[Ama01] F. C. Amaral. *Tecnicas e Aplicações para o Marketing*. Berkeley Brasil, 2001.

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.

[Ber02] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, 2002.

[BL99] Michael J. A. Berry and Gordon Linoff. *Mastering Data Mining - Art and Science of Customer Relationship Management*. Wiley, 1999.

[CHNW96] David Wai-Lok Cheung, Jiawei Han, Vincent Ng, and C. Y. Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. In *ICDE*, pages 106–114, 1996.

[CHY96] Ming-Syan Chen, Jiawei Han, and Philip S. Yu. Data mining: an overview from a database perspective. *Ieee Trans. On Knowledge And Data Engineering*, 8:866–883, 1996.

[Com97] Revista Computerworld. Internet, Agosto 1997. Revista de Comunicação.

[Cro99] Two Crows. Introduction to data mining and knowledge discovery. Technical report, Two Crows Corporation, 1999. 3.a edição.

[Dun02] Margaret H. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice-Hall, Agosto 2002.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for dicovering clusters in large spatial databases with noise. In *Proceedings of 2nd Conference ok Knowledge Discovery and Data Mining*, München, Alemanha, 1996.

[FH01] Lixin Fu and Joachim Hammer. Cubist: A new approach to speeding up olap queries in data cubes, 2001.

[FPSM91] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus. *Knowledge Discovery in Databases: An overview.* AAAI Press, 1991.

[GRS98] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: an efficient clustering algorithm for large databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 73–84, Nova Iorque, EUA, 1998.

[HK00] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2000.

[HKT] J. Han, M. Kamber, and A. K. H. Tung. Spatial clustering methods in data mining: A survey.

[Ins03] Purple Insight. http://www.purpleinsight.com/products/mineset, 2003. EUA/Reino Unido.

[Kim97] Ralph Kimball. Digging into data mining - your data warehouse is your data mining platform, Outubro 1997. DBMS and Internet Systems (http://www.dbmsmag.com).

[PCY95a] J. Park, M. Chen, and P. Yu. Efficient parallel data mining for association rules, Novembro 1995.

[PCY95b] Jong Soo Park, Ming-Syan Chen, and Philip S. Yu. An effective hash based algorithm for mining association rules". In Michael J. Carey and Donovan A. Schneider, editors, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 175–186, San Jose, California, 22–25 1995.

[RTN02] Jiawei Han Raymond T. Ng. Clarans: A method for clustering objects for spatial data mining. In *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, volume 14. IEEE Computer Society, Setembro/Outubro 2002.

[Wid95] Jennifer Widom. Research problems in data warehousing. In *4th International Conference on Information and Knowledge Management*, pages 25–30, Baltimore, Maryland, Novembro 1995.

[Wiz00] WizSoft. http://www.wizsoft.com, 2000. Israel/EUA.

[ZGMHW95] Yue Zhuge, Héctor García-Molina, Joachim Hammer, and Jennifer Widom. View maintenance in a warehousing environment. In *View maintenance in a warehousing environment*, pages 316–327, Maio 1995.

[ZRL96] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 103–114, Montreal, Canada, 1996.