Visualização de Operações de Junção em Sistemas de Bases de Dados para Mineração de Dados

MARIA CAMILA NARDINI BARIONI

ORIENTADOR: PROF. DR. CAETANO TRAINA JUNIOR

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação, como parte dos requisitos para a obtenção do título de Mestre em Ciências – Área de Ciências de Computação e Matemática Computacional.

Apoio financeiro FAPESP Nº 00/04183-2

USP - São Carlos

Abril de 2002

Agradecimentos

A Deus por estar sempre zelando por mim.

A minha família, especialmente a minha mãe e ao meu pai, pelas orações e pela confiança que depositaram em mim.

Ao Caetano pela orientação e incentivo.

Ao Humberto pelo carinho e pelo apoio técnico e emocional.

A Daniela e a Elisângela pela amizade e companheirismo.

A Agma, ao Enzo e a todos do GBDI pela atenção e pela contribuição crítica a este trabalho.

Aos funcionários do ICMC pelo apoio técnico e administrativo.

A todos os amigos que conheci nesses dois anos em São Carlos, pela amizade e pelos bons momentos compartilhados.

A FAPESP pelo suporte financeiro que possibilitou o desenvolvimento deste trabalho.

Sumário

Lista de Figuras

Lista de Tabelas

Barioni, M. C. N. *Visualização de Operações de Junção em Sistemas de Bases de Dados para Mineração de Dados*. São Carlos, 2002. 64 p. Dissertação de Mestrado – Instituto de Ciências Matemáticas e de Computação, USP.

Resumo

Nas últimas décadas, a capacidade das empresas de gerar e coletar informações aumentou rapidamente. Essa explosão no volume de dados gerou a necessidade do desenvolvimento de novas técnicas e ferramentas que pudessem, além de processar essa enorme quantidade de dados, permitir sua análise para a descoberta de informações úteis, de maneira inteligente e automática. Isso fez surgir um proeminente campo de pesquisa para a extração de informação em bases de dados denominado *Knowledge Discovery in Databases* – KDD, no geral técnicas de mineração de dados – DM – têm um papel preponderante. A obtenção de bons resultados na etapa de mineração de dados depende fortemente de quão adequadamente o preparo dos dados é realizado. Sendo assim, a etapa de extração de conhecimento (DM) no processo de KDD, é normalmente precedida de uma etapa de pré-processamento, onde os dados que porventura devam ser submetidos à etapa de DM são integrados em uma única relação. Um problema importante enfrentado nessa etapa é que, na maioria das vezes, o usuário ainda não tem uma idéia muito precisa dos dados que devem ser extraídos. Levando em consideração a grande habilidade de exploração da mente humana, este trabalho propõe uma técnica de visualização de dados armazenados em múltiplas relações de uma base de dados relacional, com o intuito de auxiliar o usuário na preparação dos dados a serem minerados. Esta técnica permite que a etapa de DM seja aplicada sobre múltiplas relações simultaneamente, trazendo as operações de junção para serem parte desta etapa. De uma maneira geral, a adoção de junções em ferramentas de DM não é prática, devido ao alto custo computacional associado às operações de junção. Entretanto, os resultados obtidos nas avaliações de desempenho da técnica proposta neste trabalho mostraram que ela reduz esse custo significativamente, tornando possível a exploração visual de múltiplas relações de uma maneira interativa.

Abstract

In the last decades the capacity of information generation and accumulation increased quickly. With the explosive growth in the volume of data, new techniques and tools are being sought to process it and to automatically discover useful information from it, leading to techniques known as Knowledge Discovery in Databases – KDD – where, in general, data mining – DM – techniques play an important role. The results of applying data mining techniques on datasets are highly dependent on proper data preparation. Therefore, in traditional DM processes, data goes through a pre-processing step that results in just one table that is submitted to mining. An important problem faced during this step is that, most of the times, the analyst doesn't have a clear idea of what portions of data should be mined. This work reckons the strong ability of human beings to interpret data represented in graphical format, to develop a technique to visualize data from multiple tables, helping human analysts when preparing data to DM. This technique allows the data mining process to be applied over multiple relations at once, bringing the join operations to become part of this process. In general, the use of multiple tables in DM tools is not practical, due to the high computational cost required to explore them. Experimental evaluation of the proposed technique shows that it reduces this cost significantly, turning it possible to visually explore data from multiple tables in an interactive way.

Capítulo 1

1. Introdução

A mineração de dados (*Data Mining* – DM) e a descoberta de conhecimento em bases de dados (*Knowledge Discovery in Databases* – KDD) se tornaram proeminentes campos de pesquisa que estão se desenvolvendo rapidamente. Atualmente, muitas técnicas e algoritmos têm sido desenvolvidos e utilizados para a mineração de dados em bases de dados. Entretanto, todas as técnicas de DM desenvolvidas até agora recebem como entrada uma única tabela. Se os dados disponíveis para análise estão armazenados em múltiplas tabelas, eles têm que ser integrados em uma etapa de preparação, ou seja, as operações de junção não fazem parte da etapa de DM. Dessa maneira, o usuário tem que preparar vários conjuntos de dados para serem submetidos, um de cada vez, às ferramentas de DM. Esses conjuntos de dados são preparados através da escolha de um conjunto de tabelas e/ou seleções de dados de mais de uma tabela, que são "ligadas" e enviadas às etapas seguintes de mineração e visualização. Essa etapa de preparação dos dados não é automatizada por quase nenhuma ferramenta de DM.

Um problema importante enfrentado no processo de descoberta de conhecimento em dados armazenados em bases de dados é que o volume de dados a ser analisado é muito grande, e na maioria das vezes não se sabe por onde começar essa análise.

A utilização de técnicas de visualização nos processos de DM e *Visual Data Mining* (VDM) veio possibilitar a integração do ser humano no processo de análise dos dados permitindo, entre outras coisas que, através da visualização da distribuição dos dados, o usuário possa focalizar a realização das operações em porções dos dados de interesse, diminuindo assim a pressão exercida pelo enorme volume de dados no processo de mineração.

1.1 Objetivos do Trabalho

Este trabalho teve como objetivo a exploração de recursos de visualização de dados armazenados em múltiplas relações de uma base de dados relacional, através da utilização de informações que sumarizam cada relação, evitando a necessidade de efetuar repetidamente as caras operações de junção. Esse é um recurso inédito em ferramentas para *data mining*, uma vez que as técnicas e ferramentas disponíveis permitem trabalhar em apenas uma tabela. De uma maneira geral, este trabalho considera que as relações envolvidas no processo de descoberta de conhecimento podem ser entendidas como apresentando uma estrutura hierárquica orientada por assunto, ou em estrela. Utilizando esse esquema, a técnica proposta neste trabalho permite que o usuário selecione uma relação principal e prossiga incluindo relações adicionais para compor a relação operacional que poderá então ser enviada para mineração. Essa relação operacional é formada pelos atributos da relação principal e pela seleção de funções de sumarização sobre os atributos das relações adicionais. O alto custo da junção e das operações de sumarização é reduzido através do pré-processamento dos atributos agregados que sumarizam cada relação adicional.

Essa técnica tem potencial para visualizar informações originadas em várias relações, sem necessitar do volume e tempo de processamento usualmente associados com a operação de junção, e apresenta resultados bem melhores do que a aplicação da operação de junção.

1.2 Organização do Trabalho

Este trabalho está organizado em seis capítulos da seguinte maneira:

- O capítulo 1 introduz o trabalho proposto, apresentando a motivação para a sua realização bem como a descrição de seus objetivos.
- No capítulo 2 é apresentada uma visão geral dos assuntos relacionados a este trabalho, onde são descritos conceitos gerais relacionados ao processo de descoberta de conhecimento em bases de dados, mineração de dados, *data warehouse* (DW) e visualização de dados dentro do contexto de *visual data mining*.
- O capítulo 3 aborda o estado da arte sobre as linhas de pesquisa que estão relacionadas a este trabalho, ou seja, o processo de descoberta de conhecimento em bases de dados, mineração de dados e visualização de dados. Este capítulo descreve a evolução no surgimento dessas linhas de pesquisa e os trabalhos que estão sendo realizados, apresentando o contexto onde este trabalho está inserido.

- No capítulo 4 é apresentada a técnica desenvolvida neste trabalho, denominada *wagging*. Além da definição dos conceitos relativos a essa técnica, e da descrição de problemas que incentivaram a sua criação, também é apresentado um exemplo comparando o processo *wagging* e o processo de DW.
- O capítulo 5 apresenta um histórico de desenvolvimento da ferramenta de visualização de dados armazenados em bases de dados (*FastMapDB*) onde foram integrados os resultados desta pesquisa, descrevendo o protótipo original e as etapas de trabalho realizadas para a implementação da técnica proposta. Este capítulo também apresenta os resultados da avaliação de desempenho da ferramenta.
- E, finalmente, o capítulo 6 apresenta as contribuições deste trabalho e algumas propostas de trabalhos futuros.

Capítulo 2

2. Conceitos

Neste capítulo são descritos conceitos gerais relacionados ao processo de descoberta de conhecimento em bases de dados, mineração de dados, visualização de dados dentro do contexto de *visual data mining* e *data warehouse*. Além dos conceitos relativos a cada uma das áreas citadas, outros aspectos também são abordados. Para:

- KDD: são enumeradas e especificadas todas as etapas do processo.
- DM: algumas das principais tarefas de mineração são descritas.
- DW: são apresentados o modelo multidimensional dos dados e os esquemas mais comuns para esse modelo, além de algumas operações típicas para dados multidimensionais.
- Visualização de dados: há uma caracterização das maneiras de integração de visualização de dados e *data mining*, bem como uma enumeração de alguns princípios que a representação gráfica proporcionada por um sistema VDM deve seguir.

2.1 Descoberta de Conhecimento em Bases de Dados

O processo de descoberta de conhecimento em bases de dados (*Knowledge Discovery in Databases*) tem como objetivo a identificação de padrões em conjuntos de dados, que representem informação válida, inédita, potencialmente útil e essencialmente compreensível em grandes coleções de dados. Freqüentemente o termo *data mining* vem sendo usado como sinônimo para tal processo, mas tipicamente KDD engloba mais do que *data mining*. O termo KDD se refere ao processo de descoberta de conhecimento útil como um todo enquanto *data mining* é apenas uma das etapas desse processo [Fayyad, 1997].

Segundo Han e Kamber [Han & Kamber, 2000] o processo de KDD segue uma seqüência iterativa de 7 etapas que são descritas a seguir (Figura 2.1):

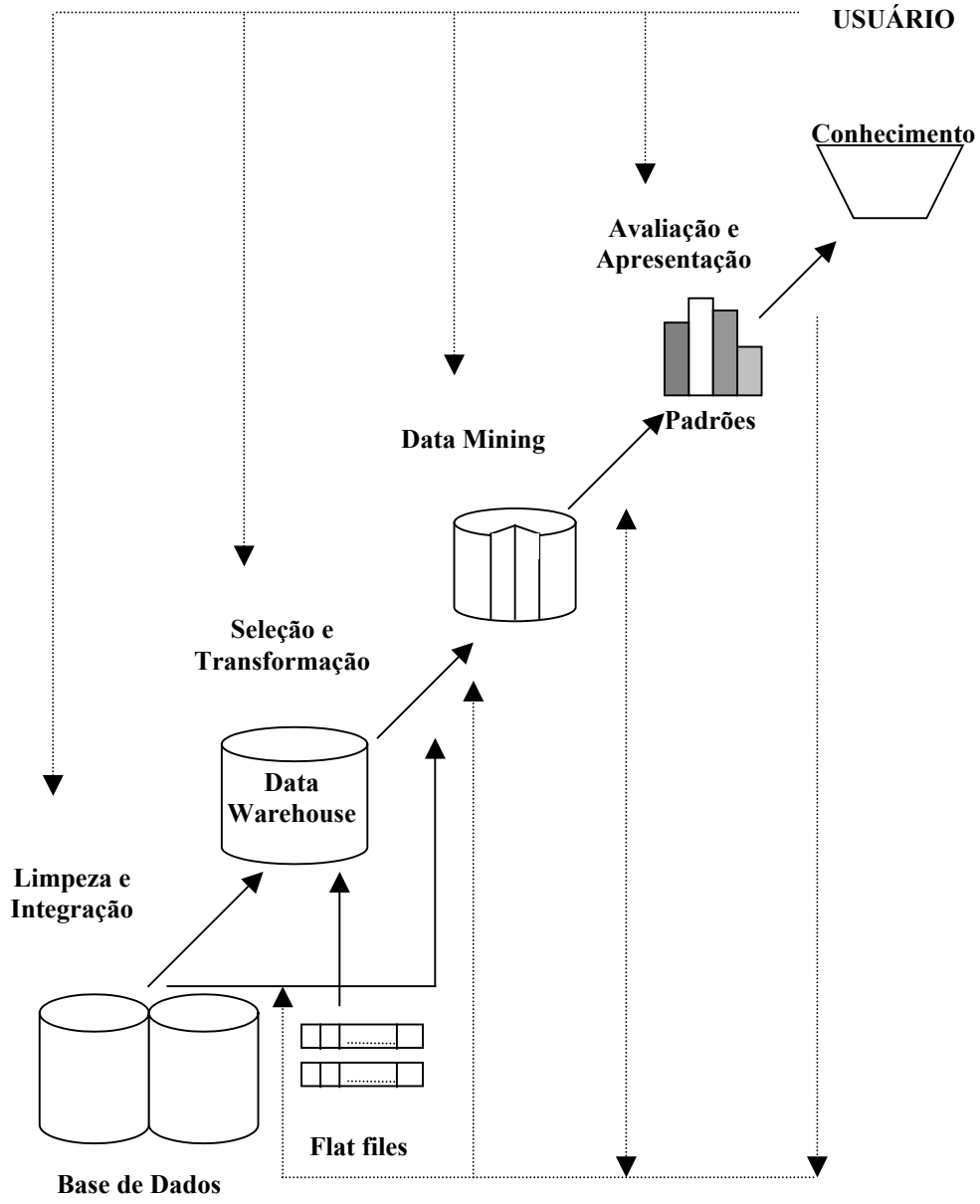


Figura 2.1 - Uma visão geral do processo de KDD [Han & Kamber, 2000].

1) Limpeza dos dados: nessa etapa, rotinas são utilizadas para tratar valores nulos, remover ruídos e corrigir dados inconsistentes.

2) Integração dos dados: nessa etapa, caso seja necessário, é feita a integração de múltiplas fontes de dados em uma única fonte coerente de dados, como um *data warehouse*. As múltiplas fontes de dados a serem integradas podem incluir múltiplas bases de dados, cubos de dados ou *flat files*. Durante essa etapa os seguintes aspectos são considerados: a integração de esquema; a detecção de redundância (duplicação) tanto de atributos quanto de tuplas; e a existência de conflitos de dados.

3) Seleção dos dados: essa etapa consiste em recuperar, da base de dados, dados relevantes à tarefa de análise.

4) Transformação dos dados: nessa etapa é realizada a transformação dos dados em formas apropriadas para a execução da mineração utilizando, por exemplo, operações de sumarização ou agregação.[1]

5) *Data Mining*: nessa etapa são aplicadas técnicas específicas para a extração de padrões de acordo com o tipo de conhecimento a ser minerado, como associação, classificação, detecção de agrupamentos (*clustering*), entre outros.

6) Avaliação dos Padrões: essa etapa tem como objetivo avaliar a utilidade dos padrões extraídos, medindo o quão interessante eles são de acordo com uma medida de interesse.

7) Apresentação do Conhecimento: nessa etapa o conhecimento minerado é apresentado através de técnicas de representação de conhecimento e visualização, como regras, tabelas, gráficos, cubos, grafos, entre outras.

É importante ressaltar que, uma tarefa de DM pressupõe a existência dos atributos em apenas uma tabela. Assim, as etapas 3) e 4) devem gerar essa tabela através de operações de seleção e junção múltiplas sobre os dados originais. Se essas fontes de dados são muitas, o próprio processo de DM envolve repetir as etapas 3) e 4) inúmeras vezes, na maior parte deles de maneira manual.

2.2 Mineração de Dados

O termo mineração de dados (*Data Mining* – DM) se refere ao processo de extração de conhecimento de grandes coleções de dados, que podem ser bancos de dados relacionais, transacionais, *data warehouses*, sistemas de banco de dados avançados, *flat files*, e até a *World Wide Web* [Han & Kamber, 2000]. Esse conhecimento descoberto se refere principalmente à extração de novos padrões e regras dos dados através da aplicação de técnicas específicas, de acordo com o tipo de conhecimento a ser minerado [Fayyad, 1997].

Segundo Chen [Chen et al., 1996] *data mining* é um extenso campo de pesquisa, que associa técnicas e conceitos de diversas áreas como sistemas de banco de dados, sistemas baseados em conhecimento, inteligência artificial, aprendizado de máquina, aquisição de conhecimento, estatística, banco de dados espaciais e visualização de dados. A contribuição da área de banco de dados é especialmente vital no que se refere a disponibilizar algoritmos e técnicas

[1] Algumas vezes a consolidação e transformação dos dados são realizadas antes da etapa de seleção dos dados, particularmente no caso de *data warehousing*.

escaláveis, para dar suporte às demais áreas envolvidas.

A seguir é feita uma descrição de algumas das tarefas de *data mining* mais significativas: detecção de agrupamentos (*clustering*), classificação e associação.

2.2.1 Detecção de Agrupamentos (*Clustering*)

Segundo Han e Kamber [Han & Kamber, 2000] a tarefa de agrupamento identifica a classe de cada objeto de maneira que, os objetos dentro de uma mesma classe apresentem alta similaridade entre si, e ao mesmo tempo, baixa similaridade em relação aos objetos das outras classes.

A medida de similaridade é feita baseada nos valores dos atributos que descrevem os objetos do banco de dados através de métodos de agrupamento, que segundo Han e Kamber [Han & Kamber, 2000], podem ser divididos em cinco categorias:

1) métodos baseados em partição: a idéia básica desse método baseia-se na construção de uma partição de um banco de dados em k *clusters* representados pelo valor médio dos objetos no *cluster* (*k-means*) ou por um objeto representativo do *cluster* que esteja localizado perto do centro do mesmo (*k-medoid*).
2) métodos hierárquicos: esse método cria uma decomposição hierárquica de um dado conjunto de objetos e pode ser classificado como sendo aglomerativo (*bottom-up*) ou divisivo (*top-down*) de acordo com a maneira como a decomposição hierárquica é realizada.
3) métodos baseados em densidade: a idéia geral desse método consiste no crescimento contínuo de um dado *cluster* até que a densidade (número de objetos) na vizinhança exceda um determinado limiar (*threshold*).
4) métodos baseados em grade ou retícula (*grid-based*): esse método quantifica o espaço do objeto em um número finito de células que formam uma estrutura de grade, onde todas as operações de agrupamento (*clustering*) são realizadas.
5) métodos baseados em modelos: cria um modelo hipotético para cada *cluster* e a idéia geral é encontrar os objetos que mais se adaptem a esse modelo.

Essa técnica pode ser usada, por exemplo, para identificar grupos de clientes em um banco de dados bancário com base em informações, como renda mensal.

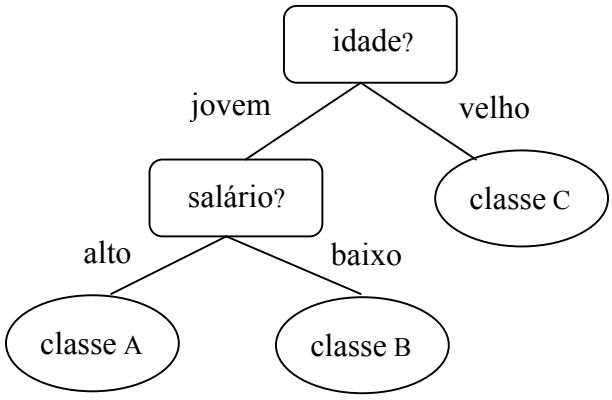
2.2.2 Classificação

É um processo que envolve dois passos, no qual o objetivo é avaliar as propriedades de um objeto, ou conjunto de objetos, em um banco de dados e classificá-los em classes diferentes, de acordo com um modelo de classificação.

No primeiro passo, o modelo de classificação é construído usando as características disponíveis nos dados. Para a construção de tal modelo, é usada uma amostra do banco de dados que é tratada como um conjunto de treinamento para a qual se conhece a classificação de cada objeto, a priori. A tupla de atributos que representa cada objeto dessa amostra é formada pelo conjunto de atributos do banco de dados e, adicionalmente, por um identificador [Chen et al, 1996]. Segundo Han e Kamber [Han & Kamber, 2000], o modelo de classificação pode ser representado, dentre outras maneiras, na forma de regras de classificação (do tipo IF-THEN) e árvores de decisão que podem ser vistas como uma representação gráfica em forma de árvore, onde cada nó interno representa um teste em um atributo, cada "caminho" representa um resultado do teste, e os nós folha representam as classes ou distribuições de classes. Um exemplo de uma árvore de decisão e de uma possível regra de classificação extraída dela pode ser vista na Figura 2.2.

No segundo passo, o modelo de classificação criado é usado, então, para a classificação de novos objetos.

A técnica de classificação pode ser aplicada, por exemplo, para extrair regras de classificação de um conjunto de dados sobre consumidores e usá-las para prever a taxa de crédito de um novo consumidor.



IF idade = "jovem" AND salário = "alto" THEN classe = "classe A"

Figura 2.2 - Árvore de decisão e regra de classificação [Han & Kamber, 2000].

2.2.3 Associação

A tarefa dessa técnica envolve a descoberta de regras de associação que indiquem correlações interessantes entre objetos de um dado banco de dados. Segundo Chen [Chen et al, 1996], uma regra de associação é uma implicação da forma X => Y, onde X e Y são subconjuntos dos atributos de um banco de dados.

Essa técnica pode ser utilizada em diversas aplicações, como a análise do carrinho de compras (*market basket analysis*). Nesse tipo de aplicação, há o interesse em descobrir associações entre os itens comprados pelos consumidores, ou seja, quais produtos são comprados junto com outros produtos. Um exemplo típico de uma regra de associação que pode ser extraída de uma base de dados de um supermercado é o seguinte: quando um consumidor compra pão ele também compra leite (pão => leite). Essas informações podem ser utilizadas, por exemplo, para dispor os itens, que são freqüentemente comprados juntos nas prateleiras, de maneira a encorajar a venda dos mesmos.

Segundo Adriaans e Zantinge [Adriaans & Zantinge, 1996], o número de regras de associação que podem ser encontradas quando se aplica a associação em um banco de dados é praticamente infinito e muitas dessas regras podem não ser interessantes. Para contornar esse problema foram introduzidas duas medidas de interesse que distinguem as regras que são interessantes das que não são. Essas medidas são: suporte que indica a freqüência com que uma regra aparece no banco de dados e confiança que indica o grau de acerto da regra.

Considerando regras de associação da forma X => Y, as medidas de suporte e confiança podem ser definidas da seguinte maneira:

$$suporte(X => Y) = \frac{nro\_tuplas\_contendo\_X\_e\_Y}{total\_de\_tuplas}$$

$$confiança(X => Y) = \frac{nro\_tuplas\_contendo\_X\_e\_Y}{nro\_tuplas\_contendo\_X}$$

2.3 Data Warehouse

De acordo com Inmon [Inmon, 1996] apud [Elmasri & Navathe, 2000], um *data warehouse* é "uma coleção de dados orientada a assunto, integrada, não volátil, variante no tempo, para suportar gerenciamento de decisão".

O processo de *data warehousing,* de uma maneira geral, envolve: a extração e pré-processamento de dados de múltiplas fontes diferentes, que passam por uma limpeza e reformatação antes de serem armazenados no *data warehouse*; e a geração de novas informações através de aplicações como OLAP (*On-Line Analytical Processing*), *data mining* e DSS (*Decision Support Systems*), que são armazenadas de volta no *data warehouse* [Elmasri & Navathe, 2000]. O processo completo de *data warehousing* é ilustrado na Figura 2.3.

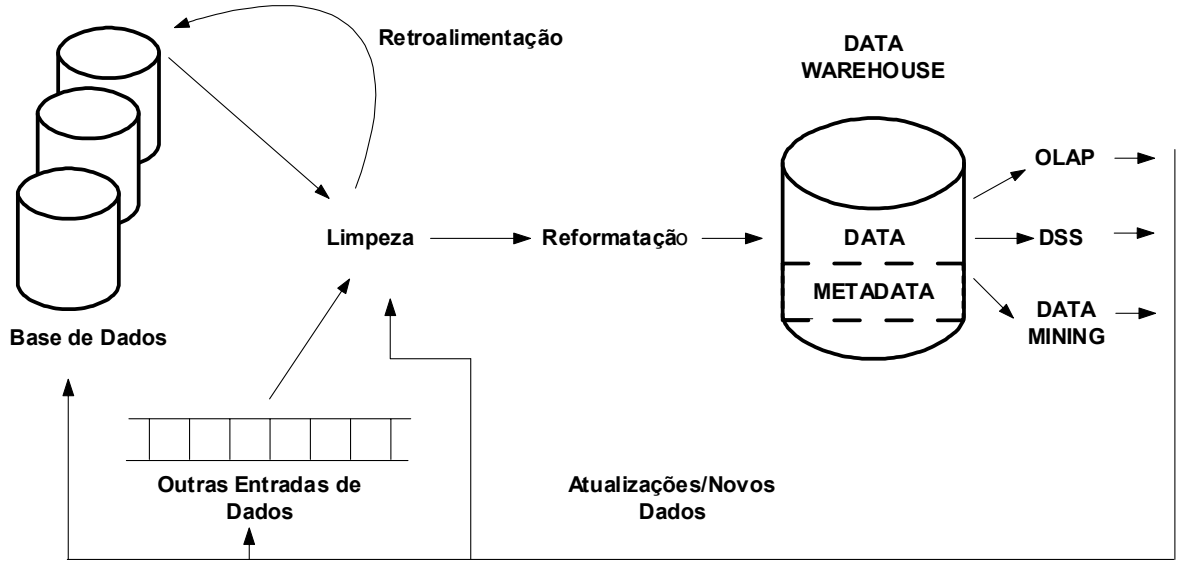


Figura 2.3 - O processo completo de data warehousing [Elmasri & Navathe, 2000].

2.3.1 Modelo Multidimensional dos Dados

Segundo Han e Kamber [Han & Kamber, 2000], um *data warehouse* é geralmente modelado através de um modelo multidimensional de dados. Para uma melhor compreensão, esse modelo considera os dados como na forma de um cubo de dados multidimensional. Esse cubo de dados é definido por dimensões e fatos, onde cada dimensão é uma unidade de análise que representa um eixo principal no estudo dos dados e corresponde a um atributo ou a um conjunto de atributos do banco de dados, enquanto os fatos são medidas numéricas que são usadas para analisar relacionamentos entre as dimensões. Para um *data warehouse* Vendas, por exemplo, pode-se armazenar informações históricas sobre dimensões como Produto, Período e Região, levando-se em consideração medidas (fatos) como NroProdutosVendidos, NroRegiõesAtendidas, e etc.

A Figura 2.4 apresenta um exemplo de uma representação multidimensional, na forma de um cubo de dados, do número de produtos vendidos em algumas regiões em certos períodos.

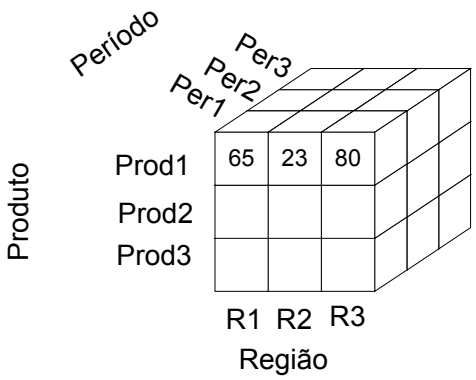


Figura 2.4 - Um Cubo de dados.

O modelo multidimensional de dados envolve dois tipos de tabelas: a tabela dimensão e a tabela fato. A tabela dimensão descreve uma dimensão associada a ela. Para o *data warehouse* Vendas, por exemplo, a tabela dimensão para Produto pode conter os atributos Número, Nome, Descrição, entre outros. Já a tabela fato representa o tema (assunto) geral ao redor do qual o modelo multidimensional é organizado. Ela possui os nomes dos fatos (ou medidas) e chaves para cada uma das tabelas dimensão relacionadas [Elmasri & Navathe, 2000] [Han & Kamber, 2000].

Os dois esquemas mais comuns para o modelo multidimensional de dados são: o esquema estrela e o esquema floco de neve [Elmasri & Navathe, 2000].

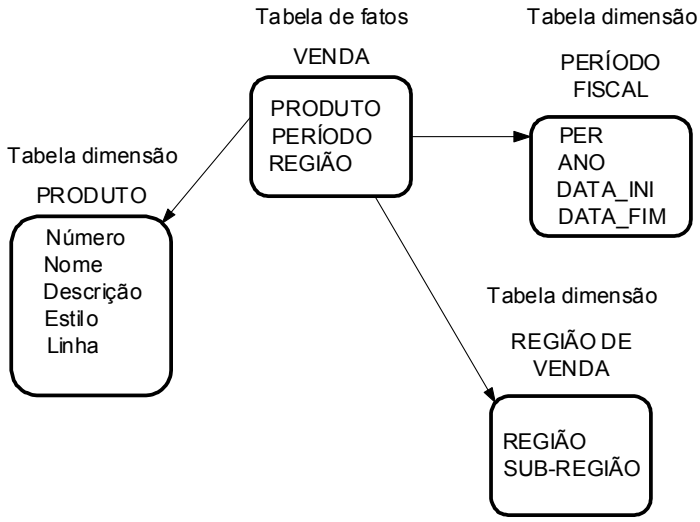


Figura 2.5 - Um esquema estrela [Elmasri & Navathe, 2000].

No esquema estrela o *data warehouse* consiste de uma tabela de fatos central contendo a massa de dados, sem redundância e de um conjunto de tabelas dimensão auxiliares, uma para cada dimensão [Han & Kamber, 2000]. Segundo Campos e Vieira [Campos & Vieira, 1998], esse esquema é denominado esquema estrela, pois apresenta a tabela fato "dominante" no centro do esquema, mantendo as tabelas dimensão nas extremidades. A ligação da tabela fato

às demais tabelas ocorre através de múltiplas junções, enquanto as tabelas dimensão se ligam apenas à tabela central através de uma única junção. Um exemplo desse esquema para Vendas considerando as dimensões Produto, Período e Região é apresentado na Figura 2.5.

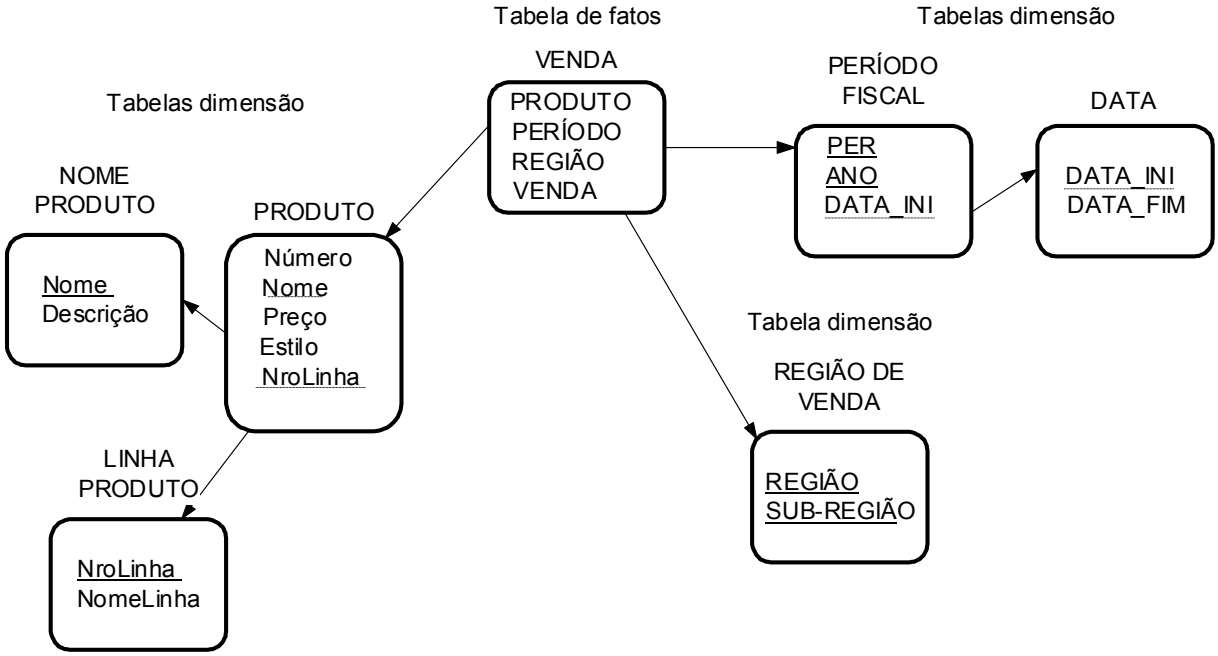


Figura 2.6 - Um esquema floco de neve [Elmasri & Navathe, 2000].

O esquema floco de neve é uma variação do esquema estrela onde as tabelas dimensão do esquema estrela são normalizadas e dessa forma organizadas em uma hierarquia, ou seja, nesse esquema uma dimensão pode ser composta de mais de uma tabela dimensão. Segundo Han e Kamber [Han & Kamber, 2000], o esquema floco de neve não é tão popular quanto o esquema estrela pois reduz a performance de varredura devido à necessidade da realização de mais junções para executar uma consulta. Na Figura 2.6 é apresentado um exemplo desse esquema para Venda.

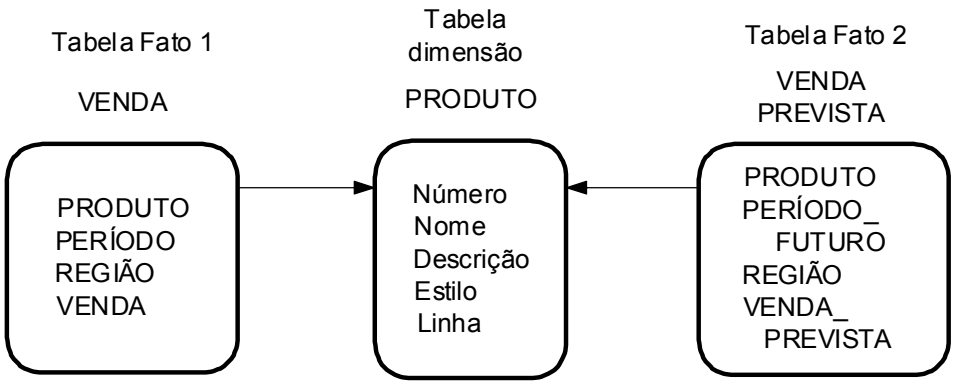


Figura 2.7 - Uma constelação de fatos [Elmasri & Navathe, 2000].

Outro esquema também utilizado para o modelo multidimensional de dados é o esquema constelação de fatos, que consiste de um conjunto de tabelas fatos que compartilham tabelas dimensão. Um exemplo desse esquema, para Vendas, é ilustrado na Figura 2.7.

2.3.2 OLAP

Segundo Han e Kamber [Han & Kamber, 2000], a organização do modelo multidimensional de dados em múltiplas dimensões, cada uma podendo também conter múltiplos níveis de abstrações, permite aos usuários uma maior flexibilidade para visualizar os dados sob diferentes perspectivas. Para materializar essas diferentes visões podem ser utilizadas um número de operações OLAP. De uma maneira geral, OLAP provê um ambiente amigável para a análise iterativa de dados.

Dentre as operações OLAP típicas, para dados multidimensionais estão [Elmasri & Navathe, 2000] [Han & Kamber, 2000]:

- **Roll-up**: permite ao usuário "subir" pelas hierarquias das dimensões, realizando agrupamentos em unidades maiores ao longo de uma dimensão. Por exemplo, agregando dados semanais, em meses ou anos.
- **Drill-down**: é o inverso da operação anterior, de maneira que permite a navegação a partir de dados menos detalhados para dados com um maior nível de detalhes. Um exemplo dessa operação seria a desagregação das informações de vendas de um país, por regiões, sub-regiões, e etc.
- **Pivoteamento**: é uma operação de visualização, que permite uma apresentação alternativa dos dados através do rotacionamento do eixo dos mesmos(Figura 2.8).

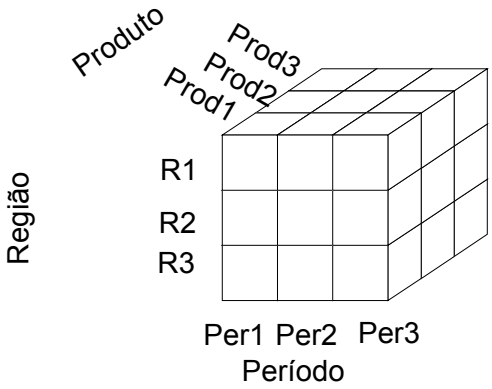


Figura 2.8 - Operação de pivoteamento: rotacionando os eixos do subcubo da Figura 2.4.

2.4 Visualização de Dados

Segundo Oliveira e Minghim [Oliveira & Minghim, 1997], "Visualização é o processo de transformar informação para uma forma visual, permitindo aos usuários observar a informação. A apresentação visual resultante permite ao cientista ou engenheiro perceber visualmente características que estão escondidas nos dados, mas que são necessárias para tarefas de exploração e análise". A visualização liga os dois sistemas de processamento de informação mais poderosos – a mente humana e o computador moderno [Eick & Fyock, 1996].

Um sistema de mineração visual de dados integra técnicas de visualização à mineração de dados procurando, dessa maneira, combinar as habilidades de exploração da mente humana com o enorme poder de processamento dos computadores e formando uma ferramenta atraente e eficiente para a compreensão da distribuição dos dados, padrões, agrupamentos (*clusters*) e etc.

Segundo [Han & Kamber, 2000] de maneira geral, visualização de dados e *data mining* são processos que podem ser integrados das seguintes maneiras:

- Visualização dos dados armazenados em banco de dados: os dados em banco de dados podem ser vistos sob diferentes níveis de granularidade ou abstração, ou como diferentes combinações de atributos ou dimensões e podem ser apresentados através de várias formas visuais, como cubos 3D, curvas, superfícies, grafos ligados e outras.
- Visualização dos resultados do processo de mineração de dados: é a apresentação dos resultados ou conhecimentos, obtidos pela mineração dos dados, através de formas visuais como árvores de decisão, regras de associação, agrupamentos (*clusters*), e outras.
- Visualização do processo de descoberta de conhecimento em bases de dados: esse tipo de visualização apresenta ao usuário as várias etapas do processo de descoberta de conhecimento em bases de dados através de formas visuais, de maneira que ele possa acompanhar todas as etapas desde a extração dos dados da base de dados passando pela limpeza, integração, pré-processamento e mineração, até o armazenamento e apresentação dos resultados.
- Mineração visual de dados interativa (*Visual Data Mining* – VDM): nesse tipo de visualização, ferramentas de visualização podem ser utilizadas no processo de *data mining* para ajudar o usuário a tomar decisões durante o processo.

Segundo Wong [Wong, 1999], a representação gráfica proporcionada por um sistema de VDM deve seguir os seguintes princípios: simplicidade, permitir autonomia ao usuário, confiabilidade, reusabilidade, disponibilidade e segurança.

A simplicidade de um sistema de VDM está relacionada com a facilidade de utilização do mesmo. O que se deseja, é que ele seja sintaticamente simples para ser utilizado facilmente. Simplicidade aqui, não significa que os recursos do sistema sejam triviais mas, segundo Traina [Traina, 2001], que ele seja de fácil aprendizado, intuitivo e use técnicas de iteração "amigáveis".

Um sistema de VDM genuíno não deve impor conhecimento aos usuários, mas guiá-los através do processo de mineração, auxiliando-os a tirarem suas próprias conclusões. Dessa maneira, os usuários, então, poderiam estudar as apresentações visuais e tomar as decisões apropriadas ao invés de simplesmente aceitar os resultados gerados automaticamente pelo sistema.

Para um sistema de VDM ser confiável, ele deve fornecer informações de estimativa de erro ou precisão sobre a geração dos resultados obtidos a cada passo do processo de mineração. Essas informações de erro, obtidas através de medidas de erro desse tipo, podem compensar a deficiência que a análise imprecisa da visualização de dados pode causar.

Em termos de reusabilidade o sistema de VDM deve ser capaz de adaptar-se a uma variedade de sistemas e ambientes para minimizar os esforços de personalização e aumentar a portabilidade.

Quanto à disponibilidade, um sistema de VDM deve ser amplamente disseminado. Dessa maneira, segundo Traina [Traina, 2001], aumenta-se a comunidade de usuários do sistema e amplia-se o retorno do esforço de seu desenvolvimento.

Finalmente, é necessário que um sistema de VDM completo dê particular atenção aos aspectos de segurança e privacidade tanto dos dados quanto dos novos conhecimentos gerados.

2.5 Conclusão

Com o intuito de permitir uma melhor compreensão deste trabalho, este capítulo apresentou uma visão geral dos assuntos relacionados a ele, em particular: KDD, DM, *data warehouse* e visualização de dados. Para KDD foram apresentadas e especificadas todas as etapas do processo, e com destaque para a etapa de DM, algumas das principais tarefas de mineração foram descritas.

Para *data warehouse*, foram descritos alguns esquemas para o modelo multidimensional de dados, os quais apresentam uma idéia geral de como são consideradas as relações envolvidas no processo de descoberta de conhecimento. Neste trabalho considera-se que essas relações podem ser entendidas como apresentando uma estrutura hierárquica orientada por assunto, ou em estrela.

E, finalmente, para visualização de dados, várias maneiras de integrá-la ao processo de DM foram apresentadas. O presente trabalho pode ser enquadrado em duas delas: visualização dos dados armazenados em banco de dados, uma vez que possibilita a visualização das operações de junção em sistemas de bases de dados; e mineração visual interativa, que é o foco principal deste trabalho, o qual permite não só a visualização das operações de junção, mas a exploração dessas operações com restrições para seleção baseadas na indicação do usuário em dados de interesse. Em particular, este trabalho permite executar a etapa de DM sobre múltiplas relações simultaneamente, fato inédito na área.

Capítulo 3

3. Estado da Arte

Este capítulo aborda o estado da arte sobre o processo de descoberta de conhecimento em bases de dados (*Knowledge Discovery in Databases* – KDD), mineração de dados (*Data Mining* – DM) e visualização de dados. Para KDD e DM são enumerados, além da motivação para o surgimento de tais disciplinas, alguns algoritmos propostos para a mineração de grandes bases de dados. Já para a visualização de dados, é dado um enfoque na utilização da visualização na mineração de dados e sendo assim, técnicas de visualização, que podem ser utilizadas em sistemas de *visual data mining*, são apresentadas e exemplificadas.

3.1 Descoberta de Conhecimento em Bases de Dados e Mineração de Dados

O progresso tecnológico ocorrido nas últimas décadas possibilitou que a quantidade de dados coletados e armazenados aumentasse rapidamente [Han & Kamber, 2000]. Isso foi devido a diversos fatores como: a disponibilização de discos baratos e com grande capacidade de armazenamento; a proliferação do uso e padronização dos sistemas gerenciadores de banco de dados (SGBD); a disseminação do uso de códigos de barra; e ao aumento da utilização de sistemas computacionais em diversas transações comerciais, científicas e governamentais [Eick & Fyock, 1996][Han & Kamber, 2000].

Esse crescimento explosivo da quantidade de dados armazenados fez com que pessoas e organizações se deparassem com o problema de fazer uso desses dados de maneira a auxiliar no processo de tomada de decisão [Fayyad, 1997]. Isso gerou a necessidade do desenvolvimento de novas técnicas e ferramentas que pudessem transformar, de maneira inteligente e automática, os dados processados em informações úteis e conhecimento [Chen et al., 1996].

Muito trabalho tem sido efetuado recentemente para o desenvolvimento de técnicas e algoritmos para serem utilizados no processo de descoberta de conhecimento em bases de dados. Neste trabalho o objetivo está voltado às atividades de pré-processamento, redução de dimensionalidade e tarefas de detecção de agrupamentos e classificação. Trabalhos relacionados a essas atividades são descritos nas subseções a seguir.

3.1.1 Pré-processamento

Atualmente, os algoritmos e ferramentas disponíveis para mineração de dados possuem requisitos específicos para a entrada de dados que fazem com que a etapa de pré-processamento do processo de KDD seja uma das etapas que mais consomem tempo no processo como um todo. Visando auxiliar a realização dessa etapa Vaduva em [Vaduva et al., 2001] apresenta o metamodelo M^4 utilizado pelo sistema *Mining Mart*. Esse sistema propõe um ambiente amigável para suportar o pré-processamento de dados para *data mining*, permitindo tanto a automatização da etapa de pré-processamento quanto a sua reutilização.

Outro problema importante enfrentado no processo de descoberta de informações em dados armazenados em bases de dados é que o volume de dados que podem estar armazenados pode ser muito grande. Com o aumento da quantidade de dados disponível para análise, a fase de pré-processamento tornou-se fundamental no processo de KDD. Em particular, as etapas de redução e seleção de dados tornaram-se cruciais para que uma massa de dados possa ser eficientemente minerada [Becher et al., 2000]. Vários trabalhos foram propostos visando solucionar esse problema. Becher em [Becher et al., 2000] focalizou seu trabalho no problema da seleção de atributos, propondo uma estratégia para automatizar a etapa de análise exploratória de dados no processo de KDD. Em [Smyth & Wolpert, 1997], é apresentada uma técnica que visa possibilitar a análise exploratória de grandes conjuntos de dados. E ainda considerando o processamento de grandes conjuntos de dados, Derthick apresenta em [Derthick et al., 1997] um ambiente de visualização interativo para a exploração de dados com o propósito de auxiliar em diversas etapas desse processo (desde a criação do conjunto de dados alvo até a projeção e visualização dos dados reduzidos), para encontrar subconjuntos interessantes de dados para minerar.

3.1.2 Redução de Dimensionalidade

Um ponto importante a ser considerado é que para muitas aplicações como, por exemplo, sistemas de armazenagem de imagens, os bancos de dados têm crescido não somente em número de registros, mas também em número de atributos. Técnicas e algoritmos de busca de dados que se mostram interessantes para conjuntos de dados pequenos podem não ser

adequados quando o volume de dados é escalado para volumes que são de ordens de grandeza maiores, tanto em número de atributos (dimensões) envolvidos, quanto em número de itens tratados. Vários pesquisadores têm atacado esse problema, focalizando seus trabalhos no desenvolvimento de algoritmos escaláveis[2] de *data mining* para grandes bases de dados.

No contexto de DM, escalabilidade refere-se tanto à complexidade computacional sobre o número de tuplas numa relação, quanto sobre o número de atributos envolvidos no processo. Sendo assim, técnicas de redução de dimensionalidade são extensivamente utilizadas em processos de DM [Kanth et al., 1998]. Dentre as técnicas de redução de dimensionalidade mais comuns estão: Análise de Componentes Principais [Anderson, 1984] [Johnson & Wichern, 1982], Análise de Fatores [Harman, 1976] e Escala Multidimensional [Torgenson, 1952] [Kruskal & Wish, 1978] [Young, 1987]. Um dos marcos dentre essas técnicas é a técnica *FastMap* proposta por Faloutsos e Lin em [Faloutsos & Lin, 1995], um dos principais algoritmos de mapeamento, para redução de dimensionalidade (número de atributos envolvidos no processo). Esse algoritmo efetua o mapeamento de dados de espaços originais de altas dimensões para pontos em um espaço alvo k-dimensional de dimensão menor, procurando preservar ao máximo a relação de distâncias originais entre os pontos no espaço de alta dimensão para gerar os pontos no espaço alvo. O ponto fundamental desse algoritmo baseia-se na projeção dos objetos em uma linha (eixo) cuidadosamente selecionada. A linha de projeção é definida através da escolha de dois objetos, chamados pivôs, e os pontos no espaço são projetados sobre essa linha (eixo) utilizando as distâncias entre ela e os objetos. A idéia chave é supor que os objetos são, realmente, pontos em algum espaço n-dimensional e tentar projetar esses pontos em k direções (eixos) mutuamente ortogonais. O algoritmo *FastMap* recebe como entrada três parâmetros: o conjunto de objetos a ser mapeado, uma função de distância métrica definida entre eles e o número de dimensões k, que indica a dimensão do espaço para o qual os objetos serão mapeados. Esse algoritmo foi utilizado na construção da ferramenta *FastMapDB* que será apresentada no capítulo 5.

3.1.3 Detecção de Agrupamento e Classificação

Os sistemas escaláveis mais representativos utilizados para o reconhecimento de agrupamentos (*clusters*), são o sistema *Clarans* (*Clustering Large Applications based upon RANdomized Search*) [Ng & Han, 1994], e o sistema *Birch* (*Balanced Iterative Reducing and*

[2] Segundo Ganti [Ganti et al., 1999a] um algoritmo é dito ser escalável se, dada uma quantidade fixa de memória principal, seu tempo de processamento aumenta linearmente com o número de registros na base de dados de entrada.

Clustering using Hierarchies) [Zhang et al. ,1996]. O sistema *Clarans* surgiu da integração de dois algoritmos usados na estatística *Pam* (*Partitioning Around Medoids*) e *Clara* (*Clustering LARge Applications*) [Kaufman & Rousseeuw, 1990] apud [Chen et al., 1996]. Ele utiliza algoritmos de particionamento e trata a tarefa de reconhecimento de agrupamentos como uma pesquisa aleatória em um grafo, onde cada nó representa uma partição do conjunto de dados. Já o sistema *Birch* trata essa tarefa concentrando-se em regiões dos dados densamente ocupadas, através da utilização de representações resumidas dessas regiões. Ele utiliza medidas que captam a proximidade natural dos dados. Essas medidas podem ser armazenadas e atualizadas de uma maneira incremental em uma *height-balanced tree*.

Para domínios de dados espaciais existem também vários estudos e técnicas desenvolvidas, como por exemplo em [Ester et al., 1998] e [Syed et al., 1999]. Para o domínio de dados métricos, onde apenas objetos e distâncias entre eles são disponíveis, um exemplo interessante é apresentado em [Ganti et al., 1999b] que utiliza a localização da vizinhança dos elementos do conjunto de dados.

Além do reconhecimento de agrupamentos (*clusters*), outro processo comumente suportado por técnicas de *data mining*, que auxilia no processo de tomada de decisão, é a classificação de objetos no conjunto de dados. Segundo Ganti [Ganti et al., 1999a], entre as técnicas de classificação mais estudadas encontram-se as redes neurais, algoritmos genéticos, métodos Bayesianos e árvores de decisões. Visões gerais sobre essas técnicas podem ser encontradas em [Adriaans & Zantingue, 1996], Han e Kamber [Han & Kamber, 2000] e Silva [Silva, 2001].

3.2 Visualização de Dados

Sabe-se que a tradução de dados e informações para um formato gráfico, denominada de visualização, possibilita que o ser humano consiga absorver e entender os dados muito mais rapidamente. Segundo Oliveira e Minghim [Oliveira & Minghim, 1997] "A conseqüência imediata da representação de informação de forma mais inteligível para a mente humana é o aumento de produtividade". Em conseqüência disso, sempre que é preciso sumarizar grandes quantidades de dados numéricos procura-se utilizar histogramas, gráficos ou algum mecanismo de apresentação visual. Entretanto, quando muitas das informações que devem ser apresentadas encontram-se em espaços de altas dimensões, ou mesmo em espaços adimensionais (por exemplo, conjuntos de palavras), as técnicas de visualização usuais não são mais adequadas.

Na visualização de dados o problema fundamental é encontrar uma representação gráfica que reflita o conteúdo e significado dos conjuntos de dados a serem visualizados. Essa tarefa possui aspectos que dependem da classe de aplicações para a qual uma técnica específica é utilizada.

Inúmeras áreas de atividade humana que envolvem o computador têm se beneficiado dos produtos da visualização, desde aplicações envolvendo imagens médicas, meteorologia e astronomia, até as de bio-informática e visualização na Web [Oliveira & Minghim, 1997] [Traina, 2001]. Além dessas áreas citadas, merece destaque, em computação, a utilização de técnicas de visualização integradas à mineração de dados (*Visual Data Mining* – VDM).

Segundo Wong [Wong, 1999], a visualização tem sido utilizada no processo de mineração de dados como uma ferramenta de apresentação para permitir navegar sobre estruturas de dados complexas, gerar vistas iniciais, bem como apresentar os resultados de análises solicitadas pelos usuários.

Um grande número de técnicas de visualização pode ser utilizado em sistemas de VDM e segundo Keim [Keim, 2002], essas técnicas podem ser classificadas de acordo com três critérios: quais dados serão visualizados (ex: unidimensionais, bi-dimensionais, multidimensionais, etc.), as técnicas de visualização empregadas (ex: representações padrão como gráficos de barra, transformações geométricas, etc.) e as técnicas de interação e distorção utilizadas.

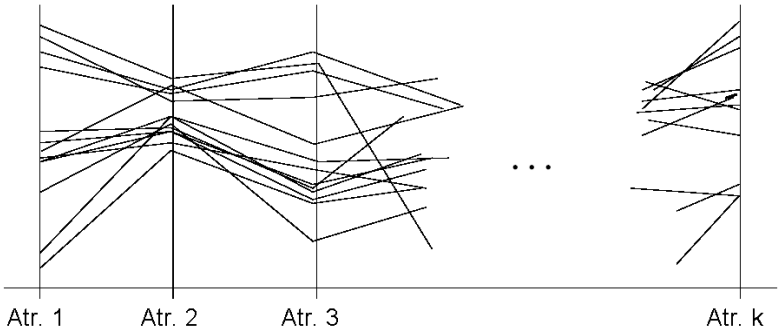


Figura 3.1 - Ilustração da técnica de coordenadas paralelas [Keim, 1997].

Dentre as técnicas de visualização está a **técnica de projeção geométrica** que lida com dados multidimensionais. Inselberg e Dimsdale, propõem em [Inselberg & Dimsdale, 1990] uma técnica de projeção geométrica, denominada técnica de coordenadas paralelas (Figura 3.1), onde a idéia básica consiste no mapeamento de um espaço *k*-dimensional sobre um dispositivo de apresentação bidimensional utilizando *k* eixos paralelos eqüidistantes a um dos eixos de apresentação dos dados. Os eixos do gráfico correspondem às dimensões e são

escaladas linearmente pelos valores de mínimo e máximo de cada dimensão correspondente. Cada objeto do conjunto de dados a ser visualizado é apresentado como um polígono que intercepta os eixos nas alturas correspondentes aos valores considerados em cada dimensão. Uma variação dessa técnica pode ser vista em [Fua et al., 1999].

Outras técnicas de visualização utilizadas em VDM são as baseadas em ícones, e são denominadas **técnicas de apresentação icônicas**. Nesse caso, a idéia central é mapear cada item de dado multidimensional em um ícone. Um exemplo de utilização dessa técnica (Figura 3.2) é apresentado em [Pickett & Grinstein, 1988], onde duas das dimensões do conjunto de dados são mapeadas às duas dimensões do dispositivo de exibição, e as demais dimensões são mapeadas criando/alterando os ícones [Hinneburg et al., 1999]. É importante notar que esta técnica limita o número de dimensões a serem apresentadas. Uma variação mais refinada de apresentações icônicas, que permite visualizar um número arbitrário de dimensões dos dados em análise, é chamada de técnica baseada em codificação de forma e pode ser vista em [Beddow, 1990].

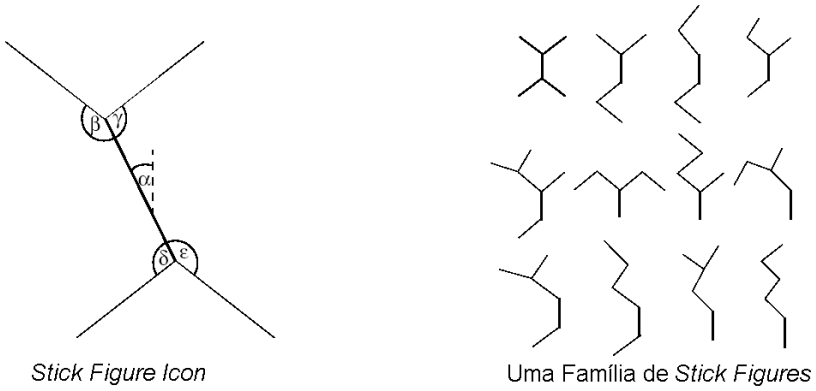


Figura 3.2 - Ilustração da técnica de apresentação icônica proposta por Pickett e Grinstein [Keim, 1997].

Uma outra abordagem utilizada para a visualização é a **técnica orientada a pixel**. Nessa técnica cada valor de dado é associado a uma cor e todos os valores de dados pertencentes a um atributo são apresentados em janelas de visualização específicas (Figura 3.3). Como essa técnica utiliza apenas um pixel da tela para cada valor de atributo da tupla, ela permite visualizar conjuntos de dados realmente grandes (testes foram efetuados para até 1 bilhão de valores de atributos). Sendo assim, a questão principal aqui se baseia em como organizar os pixels na tela de forma que a informação desejada seja captada mais facilmente. Uma visão geral sobre essa técnica bem como sobre as propostas de organização de tela para apresentação dos pixels encontra-se em [Keim & Kriegel, 1994].

Existem também as **técnicas hierárquicas**, que sub-dividem um espaço *k*-dimensional em sub-espaços que são apresentados hierarquicamente. Um exemplo dessa classe de técnicas é a chamada técnica de '*stacking*', que sub-divide o espaço de dados em sub-espaços bi-dimensionais e efetua a apresentação da informação graficamente [LeBlanc et al., 1990].

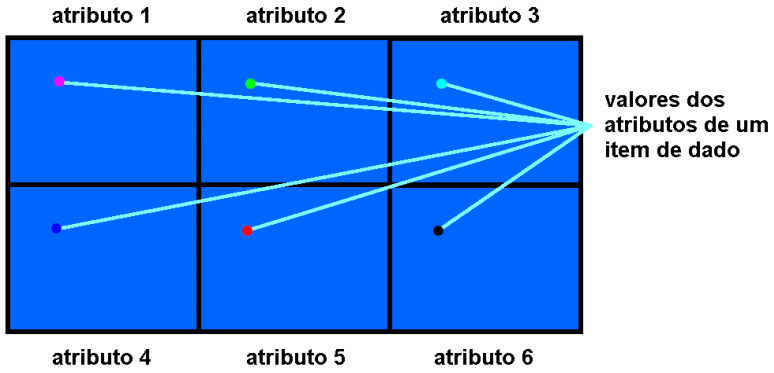


Figura 3.3 - Ilustração da técnica orientada a pixel [Keim, 1997] .

Além das técnicas de visualização, segundo Hinneburg [Hinneburg et al., 1999], é importante ressaltar que para obter-se uma exploração efetiva dos dados é necessária a utilização de algumas técnicas de interação e distorção. As técnicas de interação permitem ao usuário: interagir diretamente com a visualização obtida efetuando manipulações de mapeamento, projeção, filtragem, escalonamento, etc.; e alterar dinamicamente a visualização de acordo com os objetivos da exploração dos dados. Já as técnicas de distorção auxiliam no processo de exploração interativa da visualização, provendo meios para focalizar determinadas regiões, enquanto preserva uma visão geral do conjunto de dados. Isto é, apresentam-se algumas regiões dos dados com alto grau de detalhe, enquanto outras são mostradas com grau de detalhe muito menor. Uma visão geral sobre algumas técnicas de interação e distorção é apresentada em [Keim, 2002].

Uma das aplicações mais importantes para VDM é a certificação e delimitação de agrupamentos (*clusters*) no conjunto de dados em análise. Entre os trabalhos desenvolvidos nesta área está o sistema *HD-Eye* [Hinneburg et al., 1999], que combina a utilização de um algoritmo de detecção de agrupamentos, chamado de *OptiGrid* [Hinneburg & Keim, 1999] apud [Hinneburg et al., 1999], com técnicas de visualização que suportam o processo de agrupamento (*clustering*) através da apresentação das informações importantes visualmente. As técnicas de visualização utilizam uma combinação das técnicas orientadas a pixel e icônicas e permitem que os usuários especifiquem diretamente os separadores de *clusters* nas visualizações.

Outros exemplos de sistemas e estruturas que integram visualização à mineração de dados estão presentes em [Stolte et al., 2002] e [Kreuseler & Schumann, 2002]. Stolte em [Stolte et al., 2002], descreve o sistema *Polaris*. Esse sistema apresenta uma interface que permite a exploração e análise de grandes bancos de dados multidimensionais. Essa interface possibilita a construção de especificações visuais a partir de representações gráficas baseadas em tabelas e habilita a geração de um conjunto preciso de consultas relacionais a partir dessas especificações. Já em [Kreuseler & Schumann, 2002] é apresentada uma estrutura geral que utiliza uma combinação de métodos analíticos e visuais para a realização de VDM. Nessa proposta, a integração de pré-processamento e visualização possibilita a exploração de grandes espaços de informação em diferentes níveis de detalhes, permitindo que o usuário refine arbitrariamente uma visão geral inicial fornecida de todo o espaço de informação.

3.3 Conclusão

Nas últimas décadas, a capacidade de gerar e coletar informações aumentou rapidamente. Essa explosão no volume de dados gerou a necessidade do desenvolvimento de novas técnicas e ferramentas que pudessem processar essa enorme quantidade de dados em informações úteis, de maneira inteligente e automática. Isso fez surgir um proeminente campo de pesquisa para a extração de informação em bases de dados denominado *Knowledge Discovery in Databases*. Atualmente, muitas técnicas e algoritmos têm sido desenvolvidos e utilizados para a descoberta de conhecimento em bases de dados. Alguns trabalhos têm procurado integrar técnicas de visualização à mineração de dados, e é nesse contexto que está inserido o presente trabalho.

Capítulo 4

4. O Processo *Wagging*

Este capítulo apresenta a técnica *wagging* desenvolvida neste trabalho. Essa técnica utiliza a visualização de dados para preparar interativamente dados, oriundos de múltiplas tabelas relacionais, para serem submetidos a tarefas de mineração. Além da definição dos conceitos relativos a essa técnica, e da descrição dos problemas que incentivaram a sua criação, também é apresentado um exemplo comparando o processo *wagging* e o processo de DW.

4.1 Descrição do Problema

A obtenção de bons resultados em um processo de descoberta de conhecimento em dados armazenados em bases de dados depende de uma adequada preparação desses dados. Sendo assim, a etapa de extração de conhecimento (DM) no processo de KDD, como foi descrita na seção 2.2, é normalmente precedida de etapas de limpeza e pré-processamento, onde os dados disponíveis são preparados para a realização da mineração. Usualmente, nessa fase as diferentes tabelas, que porventura incluam dados que devem passar pelo processo de DM, são integradas através de operações de junção, e a seguir as tabelas resultantes são reduzidas através de operações de seleção. As tarefas de DM demandam grandes recursos computacionais, em termos de volume de processamento e memória. Assim, elas dependem fortemente de quão adequadamente o preparo dos dados é realizado, incluindo o quanto a massa de dados inclui/reduz dados a serem submetidos a elas. É importante ressaltar que as etapas de limpeza e pré-processamento são majoritariamente controladas de forma manual, o que leva a um dos pontos fracos do processo como um todo, pois implica em forçar o analista a decidir quais partes dos dados disponíveis devem ser submetidos à etapa de DM antes que esse processo possa retornar qualquer informação sobre o que se busca. É importante ressaltar também que muitas vezes, durante a especificação dos dados que irão ser minerados, o

usuário ainda não tem uma idéia muito precisa dos dados que devem ser extraídos. Dessa maneira, toda vez que o usuário sente a necessidade de algum dado que não foi extraído, o processo de exploração tem que ser interrompido e um novo pré-processamento tem que ser realizado para que os dados requeridos sejam extraídos e possam ser analisados.

4.2 Descrição da Solução Proposta

Neste trabalho é apresentada uma técnica que visa suportar a mineração visual de dados de múltiplas relações, que é denominada processo *wagging* [Barioni et al., 2002].

Essa técnica permite que a etapa de mineração de dados possa atuar sobre várias relações simultaneamente, trazendo as operações de junção para serem parte do processo. Isso é possível através da utilização de informações agregadas de cada relação envolvida no processo, evitando a necessidade de executar repetidamente a cara operação de junção. Neste trabalho, as relações envolvidas no processo de KDD são consideradas como apresentando uma estrutura hierárquica orientada por assunto ou um esquema estrela. Utilizando esse esquema, o usuário começa selecionando uma relação principal (a relação base) e prossegue incluindo outras relações (relações subordinadas). A inclusão de relações subordinadas é realizada pela iteração de dois passos principais: no primeiro os atributos da relação base são analisados, e no segundo os atributos das relações subordinadas são analisados. Esses dois passos são repetidos iterativamente em um processo de "vai e vem" (*wagging*) entre a relação base e as relações subordinadas. Os atributos de cada nova relação subordinada incluída utilizam operações de sumarização para compor uma relação resultante, chamada de relação operacional, que pode então ser utilizada nos passos seguintes de mineração e visualização. O esquema estrela utilizado é construído usando relações de cardinalidade opostas àquelas utilizadas em esquemas estrela de aplicações OLAP. Os agregados também são construídos baseados nos atributos, e não seguindo o modelo floco de neve estático, como nas técnicas OLAP correntes. O alto custo da junção e das operações de sumarização é reduzido através do pré-processamento dos atributos agregados que sumarizam cada relação adicional. Dessa maneira, o alto custo computacional é pago na etapa de pré-processamento do processo de KDD, mas o processo de DM pode acessar seletivamente as informações derivadas do processo *wagging*, emulando a geração de diversos conjuntos de dados em tempo de análise, a um custo muito baixo.

4.2.1 Comparação entre o Processo *Wagging* e *Data Warehouse*

Para evitar equívoco, é importante que se faça uma distinção entre a técnica proposta neste

trabalho (o processo *wagging*) e DW. Ambas as técnicas assumem esquemas como o hierárquico, estrela ou floco de neve ligando todas as relações. No processo de *data warehousing*, como descrito na seção 2.3, o foco de atenção é a relação principal, que no jargão de DW é conhecida como tabela fato. Essa tabela possui como atributos os valores dos fatos (medidas numéricas) e chaves para cada tabela dimensão relacionada a ela. A tabela fato armazena todos os detalhes dos dados de um DW, e cada tabela relacionada a ela representa um parâmetro – ou uma das dimensões – que compõem os fatos.Os relacionamentos entre a tabela fato e as tabelas dimensão têm cardinalidade VÁRIOS:1, ou seja, muitas tuplas da tabela fato para cada tupla da tabela dimensão. Nessa arquitetura, a tabela fato é vista como um cubo multidimensional, havendo uma dimensão para cada tabela dimensão diretamente relacionada. O processo de análise em um DW é realizado principalmente através da realização de projeções desse cubo ao longo de suas várias dimensões.

Assim como o processo de *data warehousing*, o processo *wagging* também focaliza a tabela central. Essa tabela central, no processo *wagging*, é chamada de tabela base, pois é a tabela que sumariza os dados de interesse para a etapa de mineração. Cada tabela relacionada à tabela base descreve a ocorrência de um atributo (ou conjunto de atributos) na tabela base, e são denominadas tabelas subordinadas. Os relacionamentos entre a tabela base e as tabelas subordinadas têm cardinalidade 1:VÁRIOS. Assim, de uma maneira oposta à considerada em DW, o processo *wagging* utiliza um esquema que lembra os esquemas estrela e floco de neve, porém a cardinalidade dos relacionamentos é invertida. É importante ressaltar também que no processo de *data warehousing* a tabela central (fato) é composta de uma enorme coleção de itens, cada um individualmente irrelevante para o processo de análise, enquanto que no processo *wagging* a tabela central (base) é formada por muitos atributos com um relativamente pequeno número de tuplas, cada um semanticamente significativo para o processo de análise.

A Figura 4.1 exemplifica uma comparação entre DW e o processo *wagging* considerando a análise de uma base de dados contendo prontuários médico-hospitalares para minerar informações sobre pacientes. A Figura 4.1(a) apresenta um diagrama entidade-relacionamento da base original que é composta por cinco tabelas: Paciente, Médico, Descrição do Exame, Consulta e Exame. A Figura 4.1(b) mostra como essa base é estruturada para ser analisada em um processo de *data warehousing* , enquanto a Figura 4.1(c) mostra como essa base é estruturada para ser minerada a respeito das informações de Pacientes utilizando o processo *wagging*. Como pode ser observado na Figura 4.1, a cardinalidade N:M presente na base de dados é explorada de maneiras diferentes em ambas as técnicas. Além disso, o alvo de cada

técnica também é diferente. O objetivo do DW é analisar o banco de dados a partir da perspectiva do conjunto de exames, tratando cada ocorrência como uma transação, desconsiderando os detalhes de cada exame, médico ou paciente – a técnica de DW leva o analista a tentar diferentes detalhes manualmente, alterando as projeções das respectivas dimensões. A técnica *wagging* é mais flexível, pois permite que alvos diferentes possam ser escolhidos. Note que na Figura 4.1(c) a tabela paciente foi escolhida como alvo. A Tabela 4.1 apresenta um resumo da comparação realizada entre o processo *wagging* e DW.

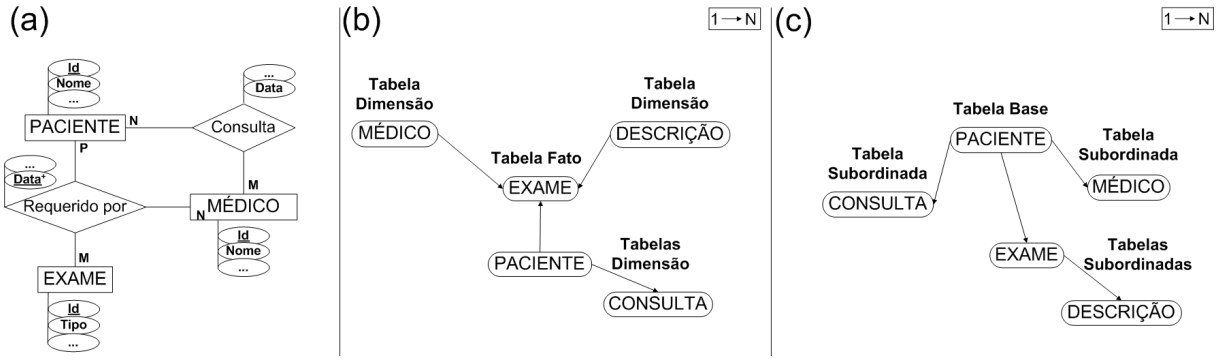


Figura 4.1 - Um exemplo comparando DW e o processo *wagging*. (a) Modelo Entidade Relacionamento de uma base de dados contendo registros de prontuários médico hospitalares; (b) modelo floco de neve da mesma base de dados no contexto de DW; (c) esquema hierárquico da mesma base de dados no contexto do processo *wagging*.

Após o processo *wagging*, qualquer tarefa de mineração de dados ou processo de visualização pode ser realizado automaticamente, com as tarefas de mineração de dados analisando várias combinações diferentes de detalhes para minerar as informações desejadas.

Tabela 4.1 – Resumo da comparação realizada entre o processo *wagging* e DW.

Processos / Características	*Wagging*	*Data Warehouse*
Esquemas	hierárquico, estrela ou floco de neve	hierárquico, estrela ou floco de neve
Tabela Central	sumariza os dados de interesse para a etapa de DM	armazena todos os detalhes dos dados de um DW
Demais Tabelas	descrevem a ocorrência de um atributo na tabela base	representam os parâmetros que compõem os fatos
Cardinalidade	1:VÁRIOS	VÁRIOS:1

4.2.2 Definição de Conceitos

As etapas de mineração e visualização de dados recebem como entrada uma tabela (ou arquivo de dados). Dessa forma, antes que os dados de múltiplas relações de uma base de dados relacional possam ser processados, uma relação operacional tem que ser criada. Sem perda de generalidade, pode-se assumir que os dados a serem processados são formados por um conjunto de objetos com uma representação homogênea, e que parte desses dados são armazenados em uma tabela relacional denominada relação base *B*. Por definição, essa relação base armazena detalhes a respeito de objetos de apenas um tipo. Dessa forma, outros objetos referenciados por essa relação têm seus detalhes armazenados em um conjunto de outras tabelas da mesma base de dados, denominadas relações subordinadas S_i. Se o processo de análise utiliza dados das relações base e subordinadas, estas devem ser "ligadas" (*joined*) para criar a relação operacional *R*, que é então submetida às etapas de mineração e visualização de dados. Isto é, as etapas de mineração e visualização de dados recebem uma relação operacional *R* que é criada utilizando a informação obtida a partir de um conjunto de relações $\{B, \{S_i\}\}$.

Considerando que uma relação de uma base de dados $\mathbb{B}$ pode ser definida como um subconjunto do Produto Cartesiano dos domínios de seus atributos, a relação operacional *R* pode ser definida como a junção de:

- uma relação base $B = \{b_i\}$, onde b_i são atributos e $B \in \mathbb{B}$ e
- uma ou mais relações subordinadas $S_i = \{s_{ij}\}$, onde s_{ij} são atributos e $S_i \in \mathbb{B}$.

Uma vez que a relação base *B* é escolhida, é necessário especificar as condições de junção J_k entre a relação base e uma relação subordinada S_i ainda não selecionada, ou entre uma relação subordinada já selecionada e uma relação subordinada ainda não selecionada. Cada condição de junção J_k pode envolver mais de um atributo de cada relação "ligada", ou seja, a condição de junção é o conjunto não vazio

$$J_k(E, F, c_k) = \{c_k = <e\ \theta\ f> \mid e \in E, f \in F\}^+, E, F \in \{B, \{S_i\}\},$$

onde θ é uma operação de comparação definida no domínio dos atributos *e* e *f*. É importante notar que, começando com a relação base *B*, as relações subordinadas são "ligadas" uma de cada vez, de maneira a existir apenas um caminho de *B* para cada S_i.

A relação operacional *R* é criada utilizando atributos selecionados de *B* e funções de sumarização aplicadas sobre atributos selecionados de cada S_i, ou seja,

$$R = J_l(J_k(\{b_a, b_b, ... | b_a, b_b \in B\}, \{Agr(s_{id}), Agr(s_{ie}), ... | s_{id}, s_{ie} \in S_i\}, (b_c \theta s_{if} | b_c \in B, s_{if} \in S_i)),$$
$$\{Agr(s_{jg}), Agr(s_{jh}), ... | s_{jg}, s_{jh} \in S_j\}, (b_p \theta s_{jq} | b_p \in (B \cup S_i), s_{jq} \in S_j)),...$$

onde b_a, b_b, ... são atributos da relação base, e s_{id}, s_{ie},... e s_{jg}, s_{jh},... são atributos das relações subordinadas S_i, S_j,... que são "ligadas" à relação base (ou às relações "ligadas" anteriormente) através das condições de junção J_k, J_l, ... O subconjunto de atributos s_{id} das relações subordinadas geram atributos $Agr(s_{id})$ agregados que são incluídos na relação operacional R. O subconjunto de atributos b_p pode incluir atributos tanto da tabela base B quanto de tabelas subordinadas S_i já ligadas por operações de junção realizadas anteriormente. As funções de sumarização são as usuais fornecidas pelos sistemas gerenciadores de bases de dados, como soma (*SUM*), média (*AVG*), mínimo (*MIN*), máximo (*MAX*) e contagem (*COUNT*). Mais de uma função pode ser aplicada a cada atributo das relações subordinadas. Por exemplo, parâmetros de medida de um exame médico da base de dados exemplo, podem ser descritos na relação R através dos valores mínimo, máximo e a média para cada paciente.

Após a seleção dos atributos agregados a partir das relações subordinadas, a relação operacional R é materializada através das condições de junção, a qual então estará pronta para ser submetida para as etapas de mineração e visualização. A relação R pode incluir atributos de várias relações, mas todos os atributos são diretamente relacionados aos objetos descritos pela relação base B utilizada no início do processo. Nem todos os atributos são utilizados nos passos seguintes de análise, assim os atributos requeridos têm agora que ser selecionados. A seleção de atributos e o passo seguinte de análise são executados iterativamente, entretanto uma vez que a relação operacional R é materializada, nenhum tempo é gasto para refazer a cara operação de junção, que de outra maneira seria necessária. Após os passos de análise, quando os atributos importantes já foram determinados, a junção completa (sem a utilização das funções de sumarização) pode ser refeita para completar a etapa de mineração de dados. Utilizando a técnica proposta, as operações de junção são executadas apenas uma vez para criar a relação R. De outra maneira, as junções teriam que ser repetidas em cada passo de iteração do processo de análise.

Essa técnica resulta em uma relação operacional contendo muitos atributos, originados a partir dos atributos agregados das relações subordinadas. Isso pode sobrecarregar o passo seguinte de análise, dessa forma é interessante reduzir a quantidade de atributos submetidos a ele. Por isso, após a materialização da relação operacional, é necessário que o usuário selecione alguns atributos, em um processo de pré-análise. Essa seleção é submetida a um processo de visualização, onde o usuário deve interpretar os dados visualizados confirmando se há informações suficientes para a análise dos dados desejada. Se a informação não é suficiente,

uma nova seleção se torna necessária, de maneira que o processo *wagging* entra no processo iterativo da atividade de descoberta de conhecimento, atuando tanto na etapa de pré-processamento e preparo dos dados quanto na etapa de DM propriamente dita. A Figura 4.2 ilustra todos os passos do processo *wagging* que foram descritos aqui.

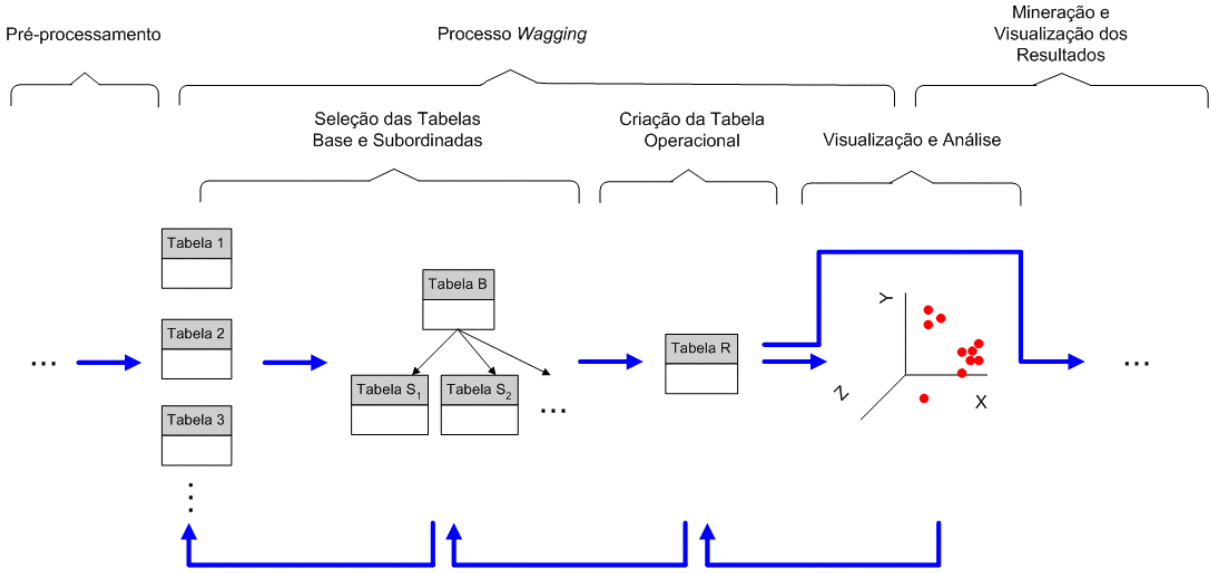


Figura 4.2 - Ilustração do processo *wagging*.

4.3 Conclusão

O pré-processamento de dados é uma etapa importante no processo de KDD, pois prepara um conjunto de dados disponível para ser minerado. Usualmente nessa etapa, a preparação de dados originados em múltiplas relações de uma base de dados é realizada através de operações de junção que consomem um certo tempo, e como o processo de KDD iterage através de todas as suas etapas, muito tempo é gasto com essas operações.

O processo *wagging*, descrito neste capítulo, cria uma relação operacional *R* que inclui atributos de uma relação base – a relação alvo do processo de KDD – e atributos calculados como funções de sumarização de atributos de uma ou mais relações subordinadas. Essa relação operacional permite que as etapas de mineração e visualização do processo de KDD realizem vários ciclos de iteração sem a necessidade de retornar à etapa de pré-processamento. Além disso, a utilização da técnica *wagging* apresenta as seguintes vantagens:

- Tempo de processamento reduzido para a criação da relação operacional *R*. Como muitos atributos são agregados em uma única operação de junção, a necessidade de outras operações de junção quando os atributos previamente selecionados não se mostram úteis é

eliminada. Sem a relação operacional *R*, uma nova relação operacional tem que ser criada após cada iteração da etapa de análise;

- Suporte para o usuário na etapa de preparação dos dados. Como a seleção de atributos submetidos para as etapas seguintes de análise é suportada por ferramentas visuais, o usuário agora tem um método rápido para verificar a utilidade de atributos escolhidos, e não precisa se basear exclusivamente na semântica dos dados e nas inferências já conhecidas sobre eles.

Capítulo 5

5. Descrição da Ferramenta *FastMapDB* e Avaliação de Desempenho

Este capítulo apresenta a ferramenta de visualização de dados armazenados em bases de dados denominada *FastMapDB*. Além da versão original são apresentadas as alterações realizadas na ferramenta, as etapas de trabalho que foram utilizadas na integração dos resultados do presente trabalho e os passos necessários para a utilização da *FastMapDB* após a integração do processo *wagging*. Este capítulo apresenta também uma avaliação de desempenho, descrevendo os experimentos que foram realizados para ilustrar o ganho de performance obtido com a utilização da técnica *wagging* e a habilidade visual da ferramenta no auxílio ao entendimento de conjunto de dados.

5.1 Versão Original

O objetivo da ferramenta *FastMapDB* [TrainaJr et al., 1999] [Traina et al, 2001] é gerar mapeamentos de dados armazenados em bases de dados relacionais permitindo a visualização desses dados em representações tri-dimensionais. Essa ferramenta foi desenvolvida com o intuito de possibilitar ao usuário "ver" a distribuição dos dados sem basear-se em qualquer propriedade espacial intrínseca possivelmente presente nos dados. Ela permite, por exemplo, verificar a existência de *outliers*, verificar a formação de agrupamentos (*clusters*) e auxiliar o usuário a escolher conjuntos reduzidos de atributos para minerar. Ela também possui recursos para que o usuário crie interativamente uma função de distância vetorial, a partir de qualquer quantidade dos atributos de uma única relação de uma base de dados conectada via ODBC. Os atributos utilizados pela função de distância podem ser quaisquer dados não categóricos, sejam eles numéricos, textuais (considerando a função Ledit) ou datas (considerando a contagem de dias a partir de uma data-referência), e podem ser ponderados, normalizados,

e/ou utilizados em escala linear ou logarítmica. A função de distância criada é utilizada para visualizar a distribuição das tuplas da relação, mapeadas para um gráfico em três dimensões. Esse sistema dispõe ainda do recurso de utilizar um atributo da relação (com domínio discreto) para classificar as tuplas da mesma (possibilitando a visualização de classes em diversas cores e formatos), além de permitir a utilização de diversos filtros para selecionar as tuplas de interesse para a visualização.

O núcleo da ferramenta foi construído utilizando o algoritmo de mapeamento de objetos em espaços de diferentes dimensões, denominado *FastMap* [Faloutsos & Lin, 1995] que foi descrito na seção 3.1.2. A ferramenta foi desenvolvida visando possibilitar ao usuário interagir com o sistema durante as várias etapas de apresentação da informação, permitindo guiar os passos das apresentações pelas preferências do usuário.

A ferramenta *FastMapDB* foi desenvolvida em linguagem *C++Builder*, para ambiente *NT*, utilizando o *BDE* e o protocolo *ODBC* para conexão com os sistemas gerenciadores de bases de dados. E atualmente pode conectar-se com gerenciadores *Oracle*, *Sybase*, *MS-SQLServer* e *Interbase* e com algumas restrições de operações também com *Paradox*.

5.2 Módulo Visualizador

Antes de adicionar o processo *wagging* à ferramenta *FastMapDB*, algumas extensões foram implementadas através do trabalho conjunto entre membros do Grupo de Bases de Dados e Imagens – GBDI – do Instituto de Ciências Matemáticas e de Computação - ICMC. Essas extensões da ferramenta consistiram na implementação e integração de um módulo de visualização que a tornou independente da ferramenta *WGnuPlot*[3], que antes era utilizada para a visualização dos objetos em gráficos bi – ou tri – dimensionais.

O módulo visualizador implementado é composto basicamente de duas janelas: uma janela para a visualização de todos os dados mapeados (janela a)), e uma janela para uma outra visualização dos dados transformados (janela b)). A janela a) é fixa, e apresenta um cubo 3D que delimita os pontos mapeados que são visualizados na janela b). Através de transformações geométricas realizadas nesse cubo (que envolvem rotação, translação e escala), o usuário pode observar a nova disposição dos pontos na janela b) e assim analisar os dados sob diferentes ângulos. Além disso, é importante notar que, os dados apresentados na janela b) podem ser manipulados e visualizados em diferentes "visões" do cubo: frontal (eixos

[3] Maiores informações em http://www.gnuplot.org

x e y), esquerda (eixos z e y) e topo (eixos x e z). A Figura 5.1 mostra um exemplo da tela do módulo visualizador.

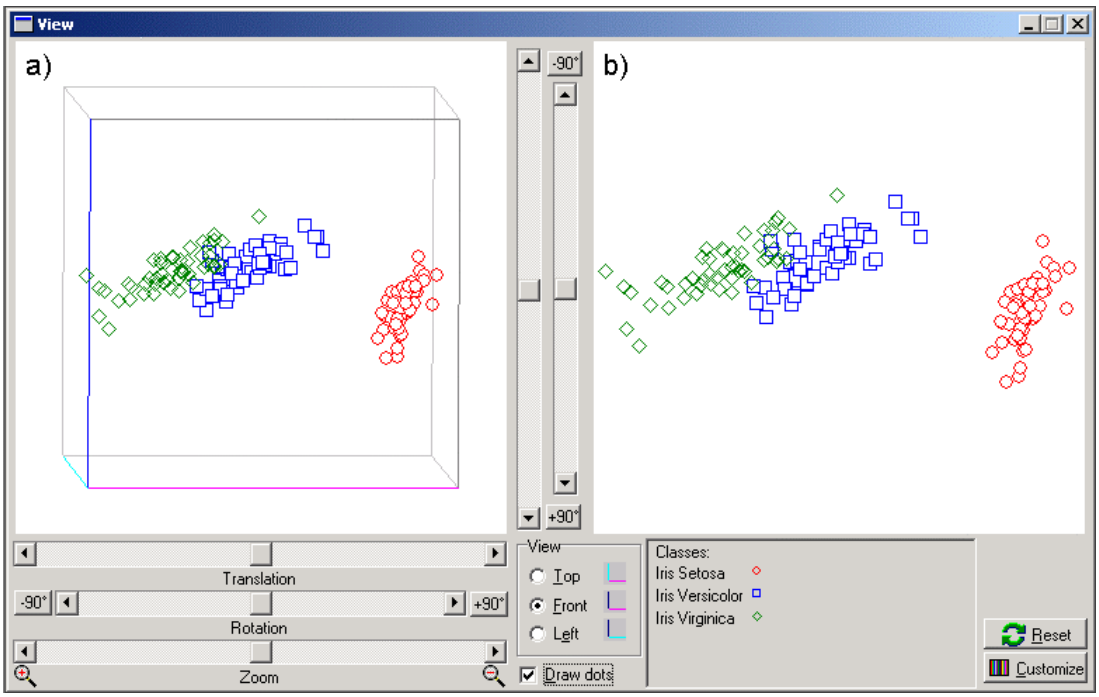


Figura 5.1 - Janela de Visualização da ferramenta *FastMapDB*. Visualização do conjunto *Iris Plant*[4].

5.3 Processo *Wagging*

Como já foi dito na seção 5.1, a versão original da ferramenta *FastMapDB,* permitia trabalhar com os atributos de apenas uma única relação. O objetivo deste trabalho baseou-se na extensão dessa ferramenta para permitir a seleção de atributos navegando-se na modelagem completa de um sistema, usando e respeitando os relacionamentos (ou dependências) entre objetos. Esse é um recurso inédito em ferramentas para DM, dado que as técnicas e ferramentas disponíveis permitem trabalhar em apenas uma tabela (concentrando o processo de extração de conhecimento em apenas uma classe de objetos de cada vez) [Fayyad, 1997].

A extensão do conceito de visualização de objetos em espaços de duas ou três dimensões para dados obtidos a partir de operações de junção (*'joins'*) [Mishra & Eich, 1992], torna essa ferramenta um recurso importante em processos de DM, pois habilita que processos de DM possam atuar sobre várias relações simultaneamente, trazendo as operações de junção para

[4] Conjunto de dados disponível no *web site* do *Machine Learning Repository* da Universidade da Califórnia em Irvine (URL: http://www.ics.uci.edu/~mlearn/MLSummary.html).

serem parte desse processo. Ao mesmo tempo, o processo (*wagging*) desenvolvido inclui recursos para minimizar a necessidade de repetir operações de junção, as quais estão entre as mais demoradas operações relacionais executadas pelo Banco de Dados.

5.3.1 Etapas de Implementação

As associações entre tabelas através de chaves e chaves estrangeiras representam os conceitos de relacionamentos de diferentes cardinalidades, e/ou abstrações de generalização. Dessa maneira, sempre que ocorre uma ligação chave/chave estrangeira, existe uma motivação semântica para a mesma, que determina o que se espera da ligação e, portanto como operações subseqüentes devem ser tratadas no que diz respeito à associação entre as duas relações. Um exemplo de como a motivação conceitual entre as ligações podem ser repassadas para a sintaxe das operações entre as relações são os comandos e restrições da linguagem SQL voltados para a construção de visões que podem ser atualizadas (restrições de chave estrangeira e *view update*). Este trabalho aplica os resultados dessas pesquisas para habilitar a visualização de múltiplas tabelas, utilizando a motivação semântica das ligações entre tabelas para especificar a construção da função distância do processo de mapeamento.

Com a adição do processo *wagging* na ferramenta *FastMapDB*, a representação dos dados mapeados no módulo visualizador da ferramenta teve de ser alterada. O módulo visualizador utilizava um ponto, ou um pequeno símbolo para indicar na tela a localização de cada tupla da relação. Essa representação é adequada quando todas as tuplas têm igual "valor". No entanto, suponha que essa relação tenha um relacionamento de cardinalidade 1:VÁRIOS com uma segunda relação. Nesse caso, cada tupla deverá "valer" tanto quanto for o número de tuplas na segunda relação que corresponde à tupla original. Isso faz aumentar a densidade de pontos ao redor do ponto original. Para se conseguir esse efeito nas operações tradicionais, onde uma única tupla é utilizada, é necessário realizar a junção das duas tabelas. No entanto, o mesmo efeito pode ser obtido, por exemplo, contando o número de tuplas da segunda tabela para cada tupla da primeira tabela, numa operação de cálculo de agregados.

A intensidade de expansão de um símbolo, ou ponto, depende da função distância original, a qual por sua vez depende dos atributos escolhidos nas duas tabelas originais, e de como eles podem ser usados. No entanto, essa intensidade também pode ser obtida ou estimada a partir de operações como média, mínimo e máximo dos atributos da segunda tabela. Note-se que média, mínimo e máximo são operações de sumarização que podem ser obtidas no mesmo passo que conta o número de tuplas da segunda tabela.

Neste trabalho a ferramenta *FastMapDB* foi estendida para suportar o processo *wagging*

através das seguintes etapas:

a) Etapa 1: foram identificadas as situações semânticas que originam as ligações de chave/chave estrangeira, e incluídas, na ferramenta, opções de "ligar" uma nova relação ao conjunto de relações já escolhidas até o momento. O processo de interação do usuário com a ferramenta foi mantido com a escolha pelo usuário de uma relação inicial (relação base), mas a partir dela outras podem agora ser escolhidas (relações subordinadas), bastando para tanto que o usuário selecione o(s) atributo(s) de "ligação" entre as relações que deseja "ligar". Quando já existe mais de uma relação escolhida, a lista de atributos que podem ser selecionados passa a incluir a união de todos os atributos da relação base, mais os atributos adicionais selecionados das relações "ligadas". A cada atributo selecionado é associada uma ou mais funções de sumarização.

b) Etapa 2: o módulo de processamento da função de distância foi estendido para tratar cada atributo selecionado proveniente das relações adicionadas através das novas opções de ligação, segundo a semântica da equivalente especificada para a ligação. A justificativa para essa extensão é que incluir atributos aumenta a dimensão do espaço original, mas como a dimensão final de visualização é sempre a mesma (espaço tri-dimensional), a inclusão de mais atributos causa um aumento na tensão ("*stress*") que o algoritmo de mapeamento impõe sobre as distâncias relativas entre os objetos mapeados. Atributos incluídos devido a ligações devem, de acordo com sua respectiva semântica, ter seus parâmetros de pesos e funções vinculados aos parâmetros dos atributos que efetuam a ligação, modulando o efeito de crescimento da representação da tupla da relação origem.

c) Etapa 3: o módulo de mapeamento foi modificado para utilizar a informação de atributos incluídos devido a ligações para restringir a operação de junção necessária ao tratamento da ligação, segundo a semântica especificada para a ligação. Isso foi feito da seguinte maneira: como os atributos incluídos devido a ligações de distância estão vinculados aos parâmetros do atributo que efetua a ligação, a visualização de cada ponto adicional, incluído na visualização, estará correlacionado com a visualização do ponto que efetua a ligação. Ou seja, os pontos oriundos da operação de junção que iriam, teoricamente, gerar uma nuvem de pontos ao redor do ponto que mapearia cada objeto sem a operação de junção, são substituídos por um símbolo cuja "área" é modulada pelo espalhamento que os atributos vindos das ligações causam na posição original de cada ponto. Assim, cada objeto original é representado não por um ponto, mas por uma região, cuja forma depende da parte da função distância que envolve os atributos incluídos devido a uma ligação.

e) Etapa 4: Uma vez completada a especificação do conjunto de atributos a serem visualizados e dos atributos agregados das demais relações, estes são calculados e armazenados na forma de uma tabela pertencente ao visualizador. É essa tabela, e não mais os dados originais, que são agora trabalhados pelo módulo visualizador.

d) Etapa 5: o módulo visualizador foi estendido para desenhar regiões além de pontos, atendendo aos dados gerados pelo módulo de mapeamento modificado.

Note-se que este trabalho, além de tirar proveito da semântica dos dados provida pelo esquema da base de dados para agilizar o processo de visualização e reduzir (potencialmente de maneira drástica) o volume de dados, abre a perspectiva de representar essa mesma semântica no resultado da visualização (pois possibilita a representação de regiões, além da mera representação de pontos), fortalecendo a capacidade de representação de informações da ferramenta e sua usabilidade para a identificação de informações nos dados visualizados.

5.3.2 Utilização da Ferramenta *FastMapDB*

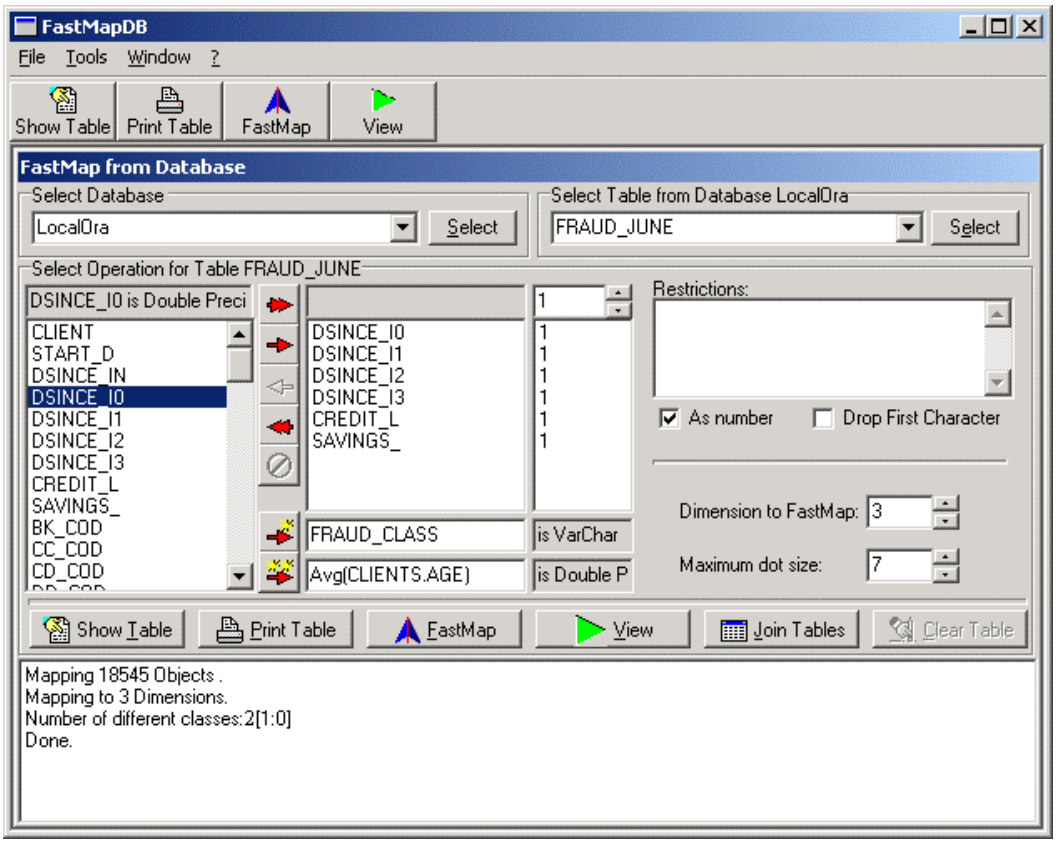


Figura 5.2 - Janela principal da ferramenta *FastMapDB*.

De uma maneira geral, após a extensão implementada, a utilização da ferramenta *FastMapDB* corresponde à execução dos seguintes passos:

1. Escolha de uma base de dados B;

2. Escolha da relação base B;
3. Escolha de uma ou mais relações subordinadas S_i e dos atributos agregados que deverão compor o conjunto de atributos disponíveis para a visualização;
4. Materialização da relação operacional R;
5. Seleção dos atributos de R para compor a visualização e definição da função distância $d()$;
6. Definição dos parâmetros da visualização;
7. Visualização interativa do resultado.

Esses passos são executados seqüencialmente pelo usuário. É possível retornar a qualquer passo anterior, entretanto, quando os passos 1 e 2 são executados, os dados dos passos subseqüentes são descartados. Os passos 3 e 4 são exclusivos do processo *wagging*, embora os outros tenham sofrido alterações para suportá-lo.

A ferramenta apresenta uma interface gráfica que guia o usuário na execução dos primeiros passos. Inicialmente, uma lista das bases de dados registradas é apresentada ao usuário. Depois que a base de dados é escolhida, uma lista das relações acessíveis nesta base de dados é apresentada, e a relação base B é então escolhida. Após a seleção dessa relação, o *FastMapDB* apresenta uma lista de atributos b_i inicial, que podem ser selecionados e que formam a relação escolhida. A partir daí, a ferramenta apresenta uma interface que permite ao usuário executar os passos de 3 a 6 (Figura 5.2) iterativa e interativamente.

As relações subordinadas são escolhidas no passo 3 (Figura 5.3(a)). Para cada relação subordinada S_i, o usuário deve indicar o(s) atributo(s) de "ligação" J_k (Figura 5.3(b)), entre as relações que deseja "ligar", ou seja, entre cada nova relação que está sendo adicionada e uma das já selecionadas (que inicialmente será apenas a relação base). Para cada relação adicional escolhida, é possível selecionar (Figura 5.3(a)) operações de sumarização sobre os seus atributos (ex: soma, média, mínimo, máximo e contagem) para compor, juntamente com os atributos da tabela base, a lista de atributos que podem ser selecionados para o mapeamento, classificação e/ou controle de tamanho. Depois que todas as relações subordinadas requeridas foram selecionadas, a relação operacional R é materializada como uma relação persistente no passo 4.

No passo 5 alguns atributos da relação R são selecionados para a visualização (mapeamento). Após a seleção desses atributos, o usuário deve proceder à definição da função distância alterando os pesos e filtros associados a cada um dos atributos selecionados.

No passo 6, para a definição dos parâmetros da visualização, um dos atributos da lista de atributos que podem ser selecionados pode ser escolhido como "separador" (classificador), fazendo com que tuplas pertencentes a diferentes classes sejam representadas em diferentes cores e formatos. Da mesma maneira, um item dessa lista de atributos também pode ser selecionado como referência para o tamanho dos pontos visualizados, ou seja, para controlar o tamanho dos pontos, sendo que esse "tamanho" vai variar de acordo com um valor máximo estabelecido pelo usuário.

E, finalmente, no passo 7 a visualização gerada pode ser explorada e interativamente manipulada através de operações de rotação, translação e escala, pelo módulo de visualização descrito na seção 5.2.

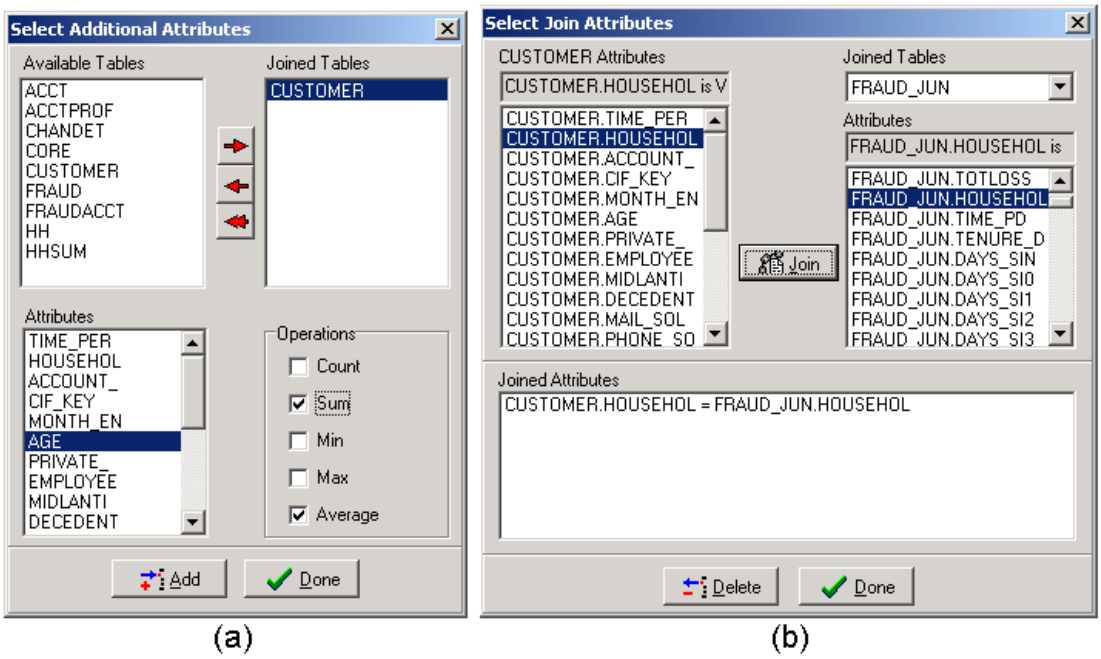


Figura 5.3 - Janelas de junção da ferramenta *FastMapDB*. (a) para seleção das relações subordinadas e dos atributos agregados; (b) para seleção das condições de junção.

5.4 Avaliação de Desempenho

Esta seção apresenta os resultados da aplicação da ferramenta *FastMapDB* em dois conjuntos de dados. São eles:

- *Fraud*: contendo dados de clientes de uma instituição financeira, com informações sobre clientes fraudulentos e uma amostra aleatória de clientes não fraudulentos[5]. Para cada

[5] Devido a restrições contratuais não é permitida a divulgação do nome da instituição e qualquer outra informação que identifique a mesma.

cliente, na tabela principal *Fraud_June*, existem 90 campos armazenando informações como a quantidade de contas de um certo tipo (por exemplo: poupança, linhas de crédito, etc. – num total de 13 tipos diferentes), o saldo em cada uma dessas contas e os dias desde que uma conta de um certo tipo foi aberta. Informações adicionais são armazenadas em outras 6 tabelas, como *Clients* (que armazena informações sobre os clientes), *Accounts* (que armazena informações sobre os diversos tipos de conta – por exemplo: data de abertura/encerramento de uma determinada conta), e etc.;

- *Congressional Voting Records*[6]: contendo registros dos votos do congresso americano de 1984. Cada tupla desse conjunto de dados corresponde ao voto de um congressista em 16 questões (por exemplo: gastos com educação, crime, etc.). Os atributos têm os valores 1 (aprovado), -1 (não aprovado) ou 0 (neutro ou abstenção). Cada tupla também apresenta um atributo categórico indicando a qual partido o congressista pertence, Republicano (168 tuplas) ou Democrata (267 tuplas).

Os experimentos realizados procuraram ilustrar dois aspectos: o ganho de performance obtido com a utilização da técnica *wagging* e a habilidade visual da ferramenta no auxílio ao entendimento de conjunto de dados. Eles foram realizados em um micro-computador Pentium III 866MHz rodando o sistema operacional Windows 2000, e utilizando um servidor de banco de dados Oracle 8i.

5.4.1 Avaliação de Performance

Para avaliar a performance da ferramenta *FastMapDB* alguns experimentos foram realizados utilizando o conjunto de dados "*Fraud*" descrito anteriormente. O primeiro experimento realizado avaliou o tempo gasto pela ferramenta variando-se o número de tabelas "ligadas". Nesse experimento foram utilizadas uma relação base mais zero, uma, duas e três relações subordinadas ligadas através de junções internas, o que resultou em relações operacionais com 19287, 18545, 17816 e 16854 tuplas respectivamente. Considerando que a relação operacional criada é utilizada várias vezes pelos passos seguintes do processo de análise, esse experimento mediu o tempo total gasto na preparação da relação operacional, utilizando e não utilizando o processo *wagging*. Os experimentos foram executados calculando-se a média de 11 conjuntos de operações de seleção, mapeamento e visualização de 10 atributos de uma relação operacional com 11 atributos. Os resultados desse experimento são apresentados na

[6] Conjunto de dados disponível no *web site* do *Machine Learning Repository* da Universidade da Califórnia em Irvine (URL: http://www.ics.uci.edu/~mlearn/MLSummary.html).

Figura 5.4(a). Esta figura também apresenta o tempo necessário para a execução de cada conjunto de operações de seleção, mapeamento e visualização. A obtenção de tempos menores utilizando números crescentes de relações subordinadas em cada conjunto de operações é resultado da aplicação das junções internas, que podem reduzir o número de tuplas na tabela resultante (devido a valores nulos nos atributos de junção) e dessa maneira os tempos de leitura e processamento para as operações de mapeamento e visualização.

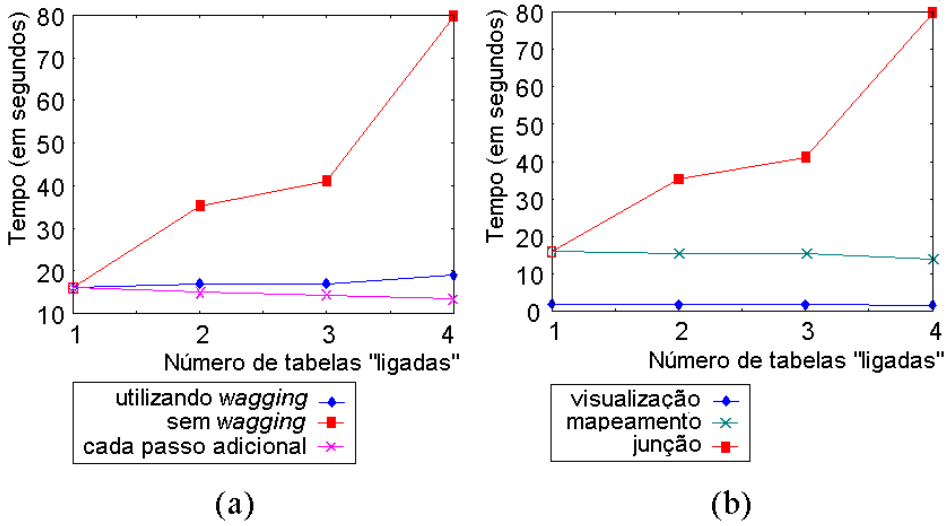


Figura 5.4 - Avaliando a performance da ferramenta *FastMapDB*. (a) tempo total gasto para realizar a junção de uma relação base com 0, 1, 2, 3 relações subordinadas e a visualização de 10 atributos a partir dos atributos resultantes das relações operacionais; (b) tempo gasto nas operações de junção, mapeamento e visualização sem a utilização da técnica *wagging*.

A Figura 5.4(b) mostra o tempo gasto para cada operação de junção, mapeamento e visualização que resultou no tempo total necessário para a preparação da relação operacional sem a utilização do processo *wagging* mostrado na Figura 5.4(a).

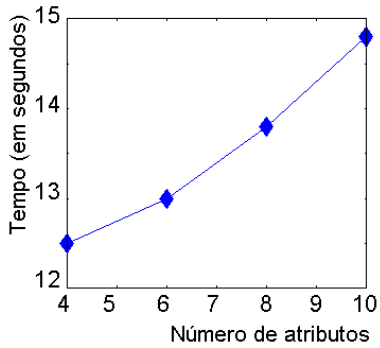


Figura 5.5 - Avaliando a performance da ferramenta *FastMapDB*. Tempo gasto pela ferramenta variando o número de atributos mapeados.

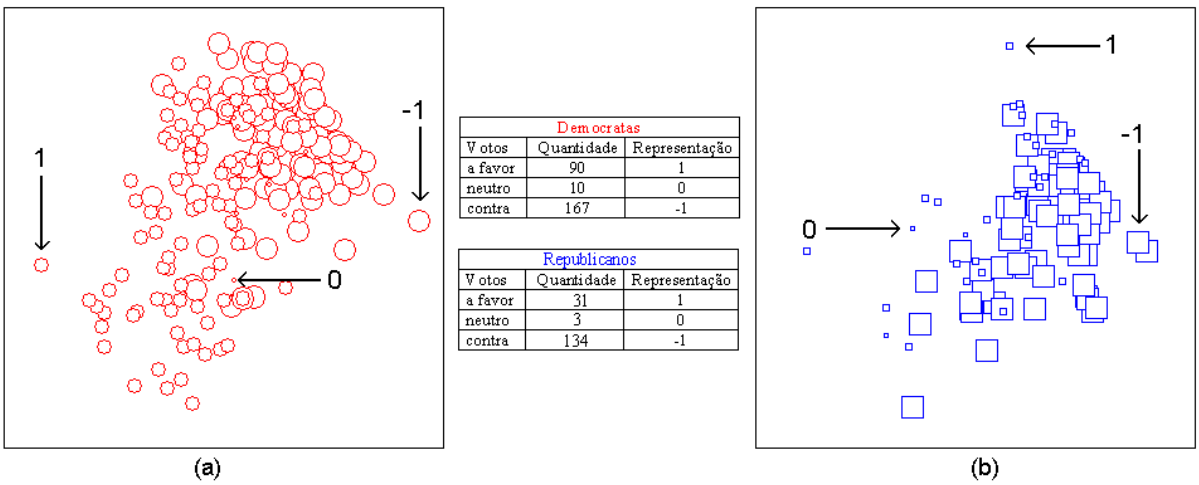
A Figura 5.5 apresenta os resultados de outro experimento realizado onde a ferramenta *FastMapDB* foi aplicada apenas na relação operacional resultante da junção de 2 relações – uma relação base e uma relação subordinada, com 18545 tuplas. Nesse experimento, a performance da ferramenta foi medida variando-se o número de atributos mapeados (de 4 até 10) de uma relação operacional com 11 atributos. Assim como no experimento anterior, os valores mostrados na Figura 5.5 correspondem à média de 11 execuções.

5.4.2 Visualização de Conjuntos de Dados

O experimento seguinte foi executado com o objetivo de ilustrar os recursos de visualização disponíveis na ferramenta *FastMapDB*. Para esse experimento foram usados os dois conjuntos de dados descritos no início da seção 5.4.

Para analisar o conjunto de dados "*Congressional Voting Records*", além da tabela já existente que armazena as informações dos votos para cada partido (tabela *Votes*), foram criadas mais 19 tabelas adicionais:

- 2 tabelas (*Democrat* e *Republican*), contendo as informações da tabela *Votes* separadas por partido;
- 16 tabelas, uma para cada atributo da tabela *Votes*, contendo o total de votos a favor, neutro e contra para cada partido;
- 1 tabela (*Votes_Summary*) totalizando, para cada congressista, o número de votos realizados a favor e contra.



Democratas		
Votos	Quantidade	Representação
a favor	90	1
neutro	10	0
contra	167	-1

Republicanos		
Votos	Quantidade	Representação
a favor	31	1
neutro	3	0
contra	134	-1

Figura 5.6 - Visualizações das junções das tabelas *Democrat* e *Republican*, com duas das tabelas auxiliares que foram criadas para cada um dos 16 atributos da tabela *Votes*. (a) visualização da junção da tabela *Democrat* com a tabela auxiliar criada para o atributo "*crime*"; (b) visualização da junção da tabela *Republican* com a tabela auxiliar criada para o atributo "*handicapped_infants*".

As visualizações desse conjunto de dados constituem um bom exemplo de como a representação de dados em regiões pode auxiliar no entendimento dos mesmos. Na ferramenta *FastMapDB* quando seleciona-se um atributo de um conjunto de dados para controlar o tamanho das regiões visualizadas desse conjunto, os valores desse atributo fazem com que o tamanho da região que representa cada tupla varie entre um mínimo (padrão na ferramenta) e um máximo (definido pelo usuário) conhecidos pela ferramenta. No caso do conjunto "*Congressional Voting Records*" as junções das tabelas *Democrat* ou *Republican* com as tabelas adicionais criadas para cada um dos 16 atributos da tabela *Votes* fazem com que seja possível visualizar como os membros de um partido votaram em cada questão, com o tamanho das representações dos dados variando de acordo com os diferentes tipos de votos. As Figura 5.6(a) e (b) apresentam o resultado do mapeamento de todos os 16 atributos, que representam as questões votadas, das tabelas resultantes das junções das tabelas *Democrat* e *Republican*, com duas das tabelas auxiliares criadas para cada atributo.

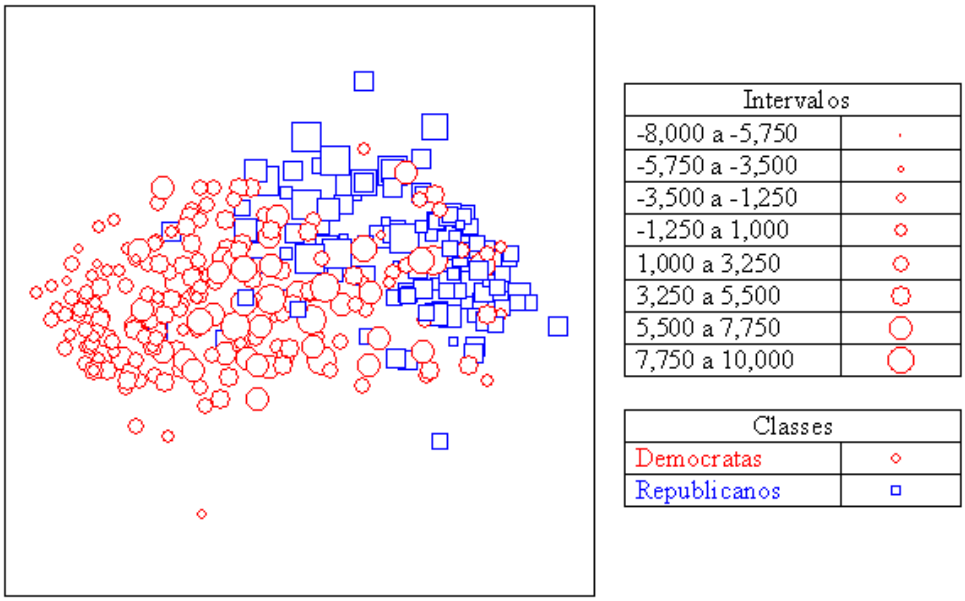


Figura 5.7 – Visualização da junção das tabelas *Votes* e *Votes_Summary*.

A junção da tabela *Votes* com a tabela *Votes_Summary* permite uma análise geral de como cada congressista votou em relação a todos os 16 atributos da tabela *Votes*. É importante destacar que, a tabela *Votes_Summary* foi criada desconsiderando os votos neutros e armazenando a soma dos votos contra como sendo valores negativos, de maneira que o tamanho da representação dos dados na visualização da sua junção com a tabela *Votes* é controlado pela aplicação da função de sumarização de soma sobre o atributo que contém todos os tipos de votos desta tabela (nesse caso apenas os votos a favor e contra). A Figura 5.7 apresenta a visualização obtida com a junção das tabelas *Votes* e *Votes_Summary*. As

variações do tamanho da representação dos dados, considerada pela ferramenta, são mostradas na legenda (Intervalos) ao lado da figura.

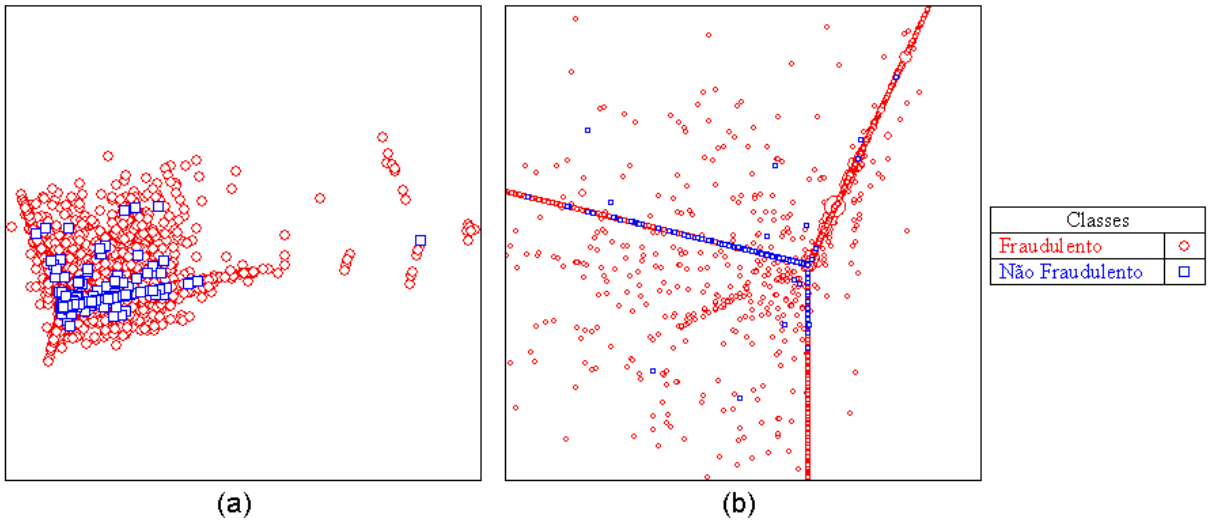


Figura 5.8 - Visualizações do conjunto de dados "*Fraud*". (a) visualizando atributos das tabelas "*Fraud_June*" e "*Clients*"; (b) visualizando atributos das tabelas "*Fraud_June*" e "*Accounts*", utilizando a média dos saldos para controlar o tamanho da representação dos dados.

Outro conjunto de dados também explorado nesse experimento foi o conjunto "*Fraud*". Utilizando os dados disponíveis nesse conjunto de dados é possível visualizar, de acordo com alguns campos selecionados, como os clientes fraudulentos estão distribuídos em relação aos não fraudulentos e também detectar quais são os campos mais prováveis de identificarem clientes fraudulentos. As Figura 5.8(a) e (b) apresentam visualizações obtidas do conjunto "*Fraud*" de acordo com diferentes conjuntos de relações e atributos selecionados.

5.5 Conclusão

Este capítulo apresentou o histórico de desenvolvimento da ferramenta *FastMapDB* descrevendo a versão original, o módulo visualizador e as etapas de implementação que foram realizadas para a integração dos resultados deste trabalho. Os passos necessários para a utilização da ferramenta, após a integração do processo *wagging*, também foram apresentados.

Além disso, resultados da aplicação da ferramenta *FastMapDB* em alguns conjuntos de dados ("*Fraud*" e "*Congressional Voting Records*") também foram mostrados. Os experimentos descritos focalizaram dois aspectos: o ganho de performance obtido com a utilização da técnica *wagging* e os recursos de visualização disponíveis na ferramenta *FastMapDB*.

Capítulo 6

6. Conclusões

A maioria das técnicas e algoritmos existentes para mineração e visualização de dados recebe como entrada uma única relação criada durante a fase inicial da etapa de pré-processamento (principalmente nas etapas de seleção e transformação dos dados) do processo de KDD. Nessa etapa, as diversas fontes de dados disponíveis são integradas e preparadas para as etapas seguintes de mineração e visualização, através de operações de junção. O volume de dados gerado por essas operações de junção tem que ser limitado por condições de seleção de objetos definidas como parâmetros das operações de junção, pois caso contrário o volume de dados gerado pode ser imenso. Como na maioria das vezes esse processo é executado de maneira manual, isso faz com que o analista tenha que decidir quais são os critérios de seleção antes mesmo de receber qualquer informação sobre o que se busca.

Outro problema importante enfrentado na etapa de pré-processamento se deve ao fato de que, na maioria das vezes, durante essa etapa o usuário ainda não tem uma idéia muito clara sobre qual porção dos dados deve ser extraída. Dessa maneira, toda vez que o usuário sente a necessidade de algum dado que não foi extraído, a etapa de exploração dos dados tem que ser interrompida e uma nova etapa de pré-processamento tem que ser executada. Como o processo todo é iterativo, as operações de junção têm que ser refeitas a cada passo, fazendo com que muito tempo seja gasto com essas operações.

6.1 Contribuições do Trabalho

O processo *wagging*, apresentado neste trabalho, contribuiu para solucionar os problemas citados acima da seguinte maneira:

- Propondo a utilização de uma relação operacional *R* criada a partir da seleção de atributos de uma relação base - que representa o alvo do processo de KDD - e de atributos

calculados como funções de sumarização aplicadas sobre atributos de uma ou mais relações subordinadas, que permite que as etapas de mineração e visualização do processo de KDD possam realizar vários ciclos de iteração sem a necessidade de retornar à etapa de pré-processamento, reduzindo efetivamente o tempo gasto no processo como um todo;

- Auxiliando na etapa de pré-processamento dos dados. Como essa etapa pode ser apoiada por ferramentas visuais, a utilização da relação operacional *R* da maneira como ela é construída, permite que o usuário seja guiado durante a inclusão de novas relações para a criação da relação operacional.

Os conceitos apresentados neste trabalho permitiram a extensão da ferramenta *FastMapDB*, incluindo recursos para visualizar interativamente seleções de atributos a partir de uma relação operacional, e utilizando esses recursos para auxiliar o processo *wagging* na criação das relações operacionais. É importante ressaltar que, a ferramenta *FastMapDB* é voltada para explorar a capacidade humana de interpretar dados visualmente, e utilizar o resultado dessa interpretação em um processo interativo para controlar a seleção das porções de dados utilizadas para a continuação das operações no processo de KDD. Isso propiciou a exploração de operações de junção com restrições para seleção baseadas na indicação do usuário em dados de interesse, o que diminuiu a pressão de aumento explosivo de dados, e trouxe o processo para um patamar aceitável de consumo de recursos computacionais.

Os experimentos descritos neste trabalho apresentaram os tempos necessários para visualizar as relações preparadas para serem submetidas à etapa de mineração de dados, comparando quando é usado e quando não é usado o processo *wagging*. Os resultados obtidos mostraram ganhos expressivos de performance, atingindo uma redução de até 85% do tempo quando é utilizado o processo *wagging* (no caso, um exemplo típico realizando a junção de 4 relações de aproximadamente 20.000 tuplas em relações consistindo de 10 a 20 atributos).

Assim, resumidamente pode-se destacar as seguintes contribuições principais deste trabalho:

- Criação do processo *wagging*;
- Desenvolvimento de uma ferramenta de suporte ao processo *wagging*, através de seu acoplamento à ferramenta *FastMapDB*.

Além dessas contribuições principais algumas outras foram feitas como, por exemplo, a extensão da ferramenta *FastMapDB* para suportar a representação de modulação do tamanho de áreas visualizadas e o módulo interativo de visualização.

6.2 Sugestões de Trabalhos Futuros

Este trabalho pode ter prosseguimento nas duas frentes de trabalho que correspondem às suas duas principais contribuições. Assim, tanto o processo *wagging* quanto seu suporte na ferramenta *FastMapDB* propiciam trabalhos que dão continuação a este.

6.2.1 Continuação do Desenvolvimento do Processo *Wagging*

6.2.1.1 Utilizando Operadores de Junção que Consideram Valores Nulos (OuterJoins)

Para isso deve-se analisar que implicações teriam, para os processos tanto de visualização quanto de *data mining* subseqüentes, a existência de valores nulos nos atributos agregados ou mesmo em atributos da tabela base. A maior dificuldade aqui é a interpretação dos resultados que seriam obtidos do processo em cada aplicação específica.

6.2.1.2 Utilizando Operadores de Junção que Consideram Operações de Comparação Através de Continência de Conjuntos.

Esse tipo de operador tem sido considerado bastante atraente em aplicações para *Web*, considerando por exemplo que uma tabela base possa representar assuntos de interesse, e as páginas da *Web* informações correspondentes em tabelas subordinadas. Nesse caso, a busca deve integrar a busca por continência de textos nas páginas de termos associados aos assuntos da tabela base.

6.2.1.3 Considerando a Análise de Dados em Evolução

Um aspecto de análise importante corresponde ao tratamento de dados que evoluem no tempo. A armazenagem de dados com referência temporal é realizada em bases de dados que suportam explicitamente o conceito de tempo como, por exemplo, pelas chamadas bases de dados temporais ou bases de dados históricas. Na maioria dos casos, o suporte a dados temporais é feito através da representação dos dados históricos em múltiplas tabelas como, por exemplo, através do uso de uma tabela separada para cada atributo do qual se pretende preservar o histórico, a qual tem como chave a mesma chave da tabela original, mais o tempo de validade dos valores do atributo, o qual é mantido em atributos não-primos da tabela. A análise desses esquemas através de processos de *data mining* tradicionais requer a integração das diversas tabelas históricas em uma única tabela através de um processo de junção.

No entanto, o tratamento do tempo acrescenta diversos aspectos semânticos aos processos de análise, que podem ser tratados de maneiras mais específicas. Por exemplo, ao invés de se

representar cada objeto como um ponto (ou uma região), cada objeto pode ser representado por uma curva que acompanha sua evolução. Assim, conjuntos de objetos que apresentam comportamentos dinâmicos semelhantes poderiam ser visualmente destacados por uma coleção de curvas com comportamentos semelhantes.

Dessa maneira, seria muito interessante efetuar uma análise de como as operações de junção poderiam ser modificadas para suportar mais adequadamente o caso específico da integração de dados históricos, e sua posterior submissão para processos de visualização e análise, também por processos específicos para suporte ao tratamento do tempo. Por exemplo, se uma tabela base tiver diversos atributos que apresentam variação temporal independente, a integração dos dados envolvendo dois ou mais atributos deve levar em conta não apenas a chave dessa relação, mas deve ser feita também a correspondência com os períodos de validade dos valores de cada atributo. Isso envolve o sincronismo entre os períodos de tempo, com a possível geração de vários períodos sincronizados mais curtos a partir de períodos individuais não sincronizados. Note-se que embora essa seja uma operação de integração de múltiplas tabelas cujo resultado é uma tabela estruturalmente semelhante à tabela resultado do processo *wagging*, a operação a ser utilizada não é a junção, o que requer o desenvolvimento de uma variação temporal para o processo *wagging*.

6.2.2 Desenvolvimento da Ferramenta *FastMapDB*

6.2.2.1 Considerando a Integração de uma Ferramenta de Manipulação de Esquemas

A escolha das tabelas e dos respectivos atributos de junção são feitos baseados tanto na semântica da aplicação quanto na estrutura sintática das tabelas. Esses dois elementos estão representados no esquema de dados da base de dados. Dessa maneira, ferramentas CASE utilizadas para a criação do esquema relacional contêm dados que permitem ao usuário indicar as junções através da manipulação gráfica dos esquemas de dados, e não pela seleção das tabelas numa lista. Assim, é interessante considerar a integração de uma ferramenta de manipulação de esquemas à etapa de escolha das tabelas e atributos do processo *wagging*. Além dessa integração propiciar uma maior facilidade de reconhecimento do significado das tabelas que são ligadas, dados adicionais fornecidos pela ferramenta CASE podem auxiliar no processo, como por exemplo no reconhecimento de chaves estrangeiras, restrições de integridade do tipo "*check*" e definição dos valores admissíveis para atributos categóricos.

6.2.2.2 Considerando a Representação da Área de Abrangência de uma Tabela Ligada

Isso é possível através da realização de um sub-mapeamento dos subconjuntos de tuplas, de

uma determinada relação subordinada, que correspondem a cada tupla da relação base, para determinar a elongação máxima que ocorre por causa desse subconjunto.

6.2.2.3 Considerando a Visualização de Dados em Evolução

Dando suporte ao preparo de dados temporais para análise, é possível conceber diversos recursos de visualização especiais para esses dados. Além da integração à ferramenta de recursos de visualização já apresentados na literatura em geral, é possível o desenvolvimento de recursos vinculados à representação espacial da evolução de dados que, a princípio, não apresentam uma distribuição espacial nativa, tal como é realizado pela ferramenta *FastMapDB*. Uma possibilidade que pode ser imediatamente reconhecida é a de visualizar como o mesmo objeto se "deslocaria" no espaço, originando uma curva. Embora conceitualmente simples de ser entendida (e portanto potencialmente útil num processo de análise de dados temporais), a implementação desse recurso pode ser problemática, o que indica a necessidade de estudos futuros mais aprofundados. Por exemplo, considerando que em dois instantes de tempo todo o conjunto original de dados evoluiu, então os objetos que são considerados pivôs para o processo de mapeamento evoluíram também. Como o processo de mapeamento depende da alocação dos pivôs, o cálculo das posições dos objetos do conjunto em dois instantes de tempo não é apenas a realização de duas operações de mapeamento. É necessário que se considere também, e separadamente, como os objetos utilizados como pivôs evoluíram.

Referências Bibliográficas

[Adriaans & Zantingue, 1996] ADRIAANS, P.; ZANTINGUE, D. *Data Mining*. 1 ed. USA: Addison-Wesley, 1996, 158 p.

[Anderson, 1984] ANDERSON, T. W. *An Introduction to Multivariate Statistical Analysis*. 2 ed. USA: John Wiley & Sons, 1984, 704 p.

[Barioni et al., 2002] BARIONI, M. C.; RAZENTE, H.; TRAINAJR, C.; TRAINA, A. Visually Mining on Multiple Relational Tables at Once. In: *East-European Conference on Advances in Databases and Information Systems*, 6., Bratislava, Eslováquia, 2002. **Proceedings**. (to appear).

[Becher et al., 2000] BECHER, J. D.; BERKHIN, P.; FREEMAN, E. Automating Exploratory Data Analysis for Efficient Data Mining. In: *International Conference on Knowledge Discovery and Data Mining*, 6., Boston, USA, 2000. **Proceedings**. AAAI Press, 2000, p. 424-429.

[Beddow, 1990] BEDDOW, J. Shape Coding of Multidimensional Data on a Microcomputer Display. In: *IEEE Conference on Visualization*, 1., San Francisco, USA, 1990. **Proceedings**. IEEE Computer Society Press, 1990, p. 238-246.

[Campos & Vieira, 1998] CAMPOS, M. L.; VIEIRA, A. Data Warehouse. In: *Escola Regional de Informática – Região Minas Gerais/Centro-Oeste*, 2., Goiânia, Brasil, 1998. **Anais**. DCC – UFMG, 1998, p. 27-70.

[Chen et al., 1996] CHEN, M.; HAN, J.; YU, P. S. Data Mining: an Overview from Database Perspective. *IEEE Transactions on Knowledge and Data Engineering*, v.8, n.6, p. 866-863, 1996.

[Derthick et al., 1997] DERTHICK, M.; KOLOJEJCHICK, J.; ROTH, S. F. An Interactive Visualization Environment for Data Exploration. In: *International Conference on Knowledge Discovery and Data Mining*, 3., Newport Beach, USA, 1997. **Proceedings**. AAAI Press, 1997, p. 2-9.

[Eick & Wills, 1993] EICK, S.; WILLS, G. J. Navigating Large Networks with Hierarchies. In: *IEEE Conference on Visualization*, 4., San Jose, USA, 1993. **Proceedings**. IEEE Computer Society Press, 1993, p. 204-210.

[Eick & Fyock, 1996] EICK, S. G.; FYOCK, D. E. Visualizing Corporate Data. *AT&T Technical Journal*, v. 75, n. 1, p. 74-85 ,1996.

[Elmasri & Navathe, 2000] ELMASRI, R.; NAVATHE, S. B. *Fundamentals of Database Systems*. 3 ed. USA: Addison-Wesley, 2000, cap. 26, p. 841-872.

[Ester et al., 1998] ESTER, M.; FROMMELT, A.; KRIEGEL, H.-P.; SANDER, J. Algorithms for Characterization and Trend Detection in Spatial Databases. In: *International Conference on Knowledge Discovery and Data Mining*, 4., New York, USA, 1998. **Proceedings**. AAAI Press, 1998**,** p. 44-50.

[Faloutsos & Lin, 1995] FALOUTSOS, C. and LIN, K.-I. FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. *ACM SIGMOD*, v.24, n.2, p.163-174, 1995.

[Fayyad, 1997] FAYYAD, U. Mining Databases: Towards Algorithms for Knowledge Discovery. *Bulletin of the IEEE Technical committee on Data Engineering*, v. 21, p. 39-48, 1997.

[Fua et al., 1999] FUA, Y. -H.; WARD, M. O.; RUNDENSTEINER, E. A. Hierarchical Parallel Coordinates for Exploration of Large Datasets. In: *IEEE Conference on Visualization*, 10., San Francisco, USA, 1999. **Proceedings**. IEEE Computer Society Press, 1999, p. 43-50.

[Ganti et al., 1999a] GANTI, V.; GEHRKE, J.; RAMAKRISHNAN, R. Mining Very Large Databases. *IEEE Computer*, v. 32, p. 38-45, 1999.

[Ganti et al., 1999b] GANTI V.; RAMAKRISHNAN R.; GEHRKE J.; POWELL A.; FRENCH, J. Clustering Large Datasets in Arbitraty Metric Spaces. In: *International Conference on Data Engineering*, 15., Sydney, Austrália, 1999. **Proceedings**. IEEE Press, 1999, p. 502-511.

[Han & Kamber, 2000] HAN, J.; KAMBER, M. *Data Mining – Concepts and Techniques*. 1. ed. New York: Morgan Kaufmann, 2000, 550 p.

[Harman, 1976] HARMAN, H.H., *Modern Factor Analysis*. 3 ed. USA: University of Chicago Press, 1976, 487 p.

[Hinneburg et al., 1999] HINNEBURG, A.; KEIM, D. A; WAWRYNIUK, M. HD-Eye: Visual Mining of High-Dimensional Data. *IEEE Computer Graphics and Applications*, v. 19, p. 22-30, 1999.

[Hinneburg & Keim, 1999] HINNEBURG, A.; KEIM, D. A. Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering. In: *International Conference on Very Large Databases*, 25., Edinburgh, Escócia, 1999. **Proceedings**. Morgan Kaufmann, p.506-517.

[Inselberg & Dimsdale, 1990] INSELBERG A.; DIMSDALE B. Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry. In: *IEEE Conference on Visualization*, 1., San Francisco, USA, 1990. **Proceedings**. IEEE Computer Society Press, 1990, p. 361-378.

[Inmon, 1996] INMON, W. H. *Building the Data Warehouse*. 2 ed. New York: John Wiley & Sons, 1996, 401 p.

[Johnson & Wichern, 1982] JOHNSON, R. A.; WICHERN, D. W. *Applied Multivariate Statistical Analysis*. 1 ed. London: Prentice-Hall, 1982, 594 p.

[Kanth et al., 1998] KANTH, K. V. R.; AGRAWAL, D.; SINGH, A. K. Dimensionality Reduction for Similarity Searching in Dynamic Databases. In: *International Conference on Management of Data*, Seattle, USA, 1998. **Proceedings**. ACM Press, 1998, p. 166-176.

[Kaufman & Rousseeuw, 1990] KAUFMAN, L.; ROUSSEEUW, P. J. *Finding Groups in Data: an Introduction to Cluster Analysis*, New York: John Wiley & Sons, 1990.

[Keim, 1997] KEIM, D. A. *Tutorial VLDB'97 on Visual Data Mining*. Disponível na URL: http://www.informatik.uni-halle.de/~keim/tutorials.html. Em 17 de abril de 2002.

[Keim, 2002] KEIM, D. A. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, v. 8, n. 1, p. 01-08, 2002.

[Keim & Kriegel, 1994] KEIM, D. A.; KRIEGEL, H. –P. VisDB: Database Exploration Using Multidimensional Visualization. *IEEE Computer Graphics and Applications*, v. 14, p. 40-49, 1994.

[Keim & Kriegel, 1996] KEIM, D. A., KRIEGEL, H. -P. Visualization Techniques for Mining Large Databases: A Comparison. *IEEE Transactions on Knowledge and Data Engineering*, v.8, n. 6, p. 923-938, 1996.

[Kreuseler & Schumann, 2002] KREUSELER, M; SCHUMANN, H. A Flexible Approach for Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, v. 8, n. 1, p. 39-51, 2002.

[Kruskal & Wish, 1978] KRUSKAL, J. B.; Wish, M. *Multidimensional Scaling*. Beverly Hills and London: SAGE Publications, 1978, 92 p.

[LeBlanc et al., 1990] LEBLANC, J.; WARD, M. O.; WITTELS, N. Exploring N-Dimensional Databases. In: *IEEE Conference on Visualization*, 1., San Francisco, USA, 1990. **Proceedings**. IEEE Computer Society Press, 1990, p 230-237.

[Mishra & Eich, 1992] MISHRA, P.; EICH, M. H. Join Processing in Relational Databases. *ACM Computing Surveys*, v. 24, p. 63-113, 1992.

[Ng & Han, 1994] NG, R. T.; HAN, J. Efficient and Effective Clustering Methods for Spatial Data Mining. In: *International Conference on Very Large Data Bases*, 20., Santiago, Chile, 1994. **Proceedings**. Morgan Kaufmann, 1994, p. 144-155.

[Oliveira & Minghim, 1997] OLIVEIRA, M. C. F.; MINGHIM, R. *Uma Introdução à Visualização Computacional*. In: Jornada de Atualização em Informática, 16., Brasília, Brasil, 1997. **Anais**. SBC, 1997, p. 85-131.

[Pickett & Grinstein, 1988] PICKETT, R. M.; GRINSTEIN, G. G. Iconographic Displays for Visualizing Multidimensional Data. In: *IEEE Conference on Systems, Man and Cybernetics*, Beijing and Shenyang, China, 1988. **Proceedings**. IEEE Press, 1998, p. 514-519.

[Silva, 2001] SILVA, D. R. *Avaliação e Acompanhamento em Educação a Distância com Uso de Data Mining*. São Carlos, 2001. Exame de Qualificação de Mestrado, Centro de Ciências Exatas e de Tecnologia, Universidade Federal de São Carlos, 63 p.

[Smyth & Wolpert, 1997] SMYTH, P.; WOLPERT, D. Anytime Exploratory Data Analysis for Massive Data Sets. In: *International Conference on Knowledge Discovery and Data Mining*, 3., Newport Beach, USA, 1997. **Proceedings**. AAAI Press, 1997, p. 54-60.

[Stolte et al., 2002] STOLTE, C.;TANG, D.; HANRAHAN, P. Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. *IEEE Transactions on Visualization and Computer Graphics*, v. 8, n. 1, p. 52-65, 2002.

[Syed et al., 1999] SYED, N. A.; LIU, H.; SUNG, K. K. A Study of Support Vectors on Model Independent Example Selection. In: *International Conference on Knowledge Discovery and Data Mining*, 5., San Diego, USA, 1999. **Proceedings**. AAAI Press, 1999, p. 272-276.

[Torgenson, 1952] TORGENSON, W. S. *Multidimensional Scaling I. Theory and Methods*. Psychometrika, 1952, cap. 17, p. 401-419.

[Traina, 2001] TRAINA, A. J. M. *Suporte à Visualização de Consultas por Similaridade em Imagens Médicas através de Estrutura de Indexação Métrica*. Tese Livre-Docente em Computação, ICMC – USP, São Carlos, 2001, 104 p.

[Traina et al., 2001] TRAINA, A. J. M.; TRAINAJR, C.; BOTELHO, E.;BARIONI M. C. N.; BUENO R. Visualização de Dados em Sistemas de Bases de Dados Relacionais. In: *Simpósio Brasileiro de Banco de Dados*, 16., Rio de Janeiro, Brasil, 2001. **Anais**. s. ed., 2001, p. 95-109.

[TrainaJr et al., 1999] TRAINAJR, C.; TRAINA, A. J. M.; FALOUTSOS, C. *FastMapDB User's Manual*. Carnegie Mellon University - School of Computer Science, 1999, 8 p.

[Vaduva et al., 2001] VADUVA, A.; KIETZ, J. U.; ZUCKER, R. M^4 – A Metamodel for Data Preprocessing. In: *International Workshop on Data Warehousing and OLAP*, 4. Atlanta, EUA, 2001. **Proceedings**. Disponível na URL: http://www.cis.drexel.edu/faculty/song/DOLAP2001/2001_dolap__Final_pgm.htm. Em 17 de abril de 2002.

[Wong, 1999] WONG, P. C. Visual Data Mining. *IEEE Computer Graphics and Applications*, v. 19, p. 20-21, 1999.

[Young, 1987] YOUNG, F. *Multidimensional Scaling: History, Theory, and Applications*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1987, 317 p.

[Zhang et al., 1996] ZHANG T.; RAMAKRISHNAN, R.; LIVNY, M. BIRCH: An Efficient Data Clustering Method for Very Large Databases. In: *International Conference on Management of Data*, Montreal, Canada, 1996. **Proceedings**. ACM Press, 1996, p. 103-114.