Renata Meireles de Paula Lavôr

Implementação de serviços relacionados à mineração de regras de associação

UFRJ/IM/NCE

2003

Implementação de serviços relacionados à mineração de regras de associação

Renata Meireles de Paula Lavôr

Universidade Federal do Rio de Janeiro

Instituto de Matemática

Núcleo de Computação Eletrônica

Mestrado

Pedro Manoel, PhD

Rio de Janeiro

2003

Implementação de serviços relacionados à mineração de regras de associação

Renata Meireles de Paula Lavôr

Dissertação submetida ao corpo docente do Instituto de Matemática – Universidade Federal do Rio de Janeiro – UFRJ, como parte dos requisitos necessários à obtenção do grau de Mestre.

Aprovada por:

_____ Orientador

Prof. Pedro Manoel da Silveira – PhD.

_____

Prof. Asterio Kiyoshi Tanaka – PhD.

_____

Prof. Maria Luiza Machado Campos – PhD.

Rio de Janeiro

2003

FICHA CATALOGRÁFICA

Lavôr, Renata Meireles de Paula.

Implementação de serviços relacionados à mineração de regras de associação / Renata Meireles de Paula Lavôr – Rio de Janeiro, 2003.

XVIII, 152 p. 29 cm (NCE/ IM/ UFRJ, M.Sc., Informática, 2003)

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro – UFRJ, Instituto de Matemática – IM, Núcleo de Computação Eletrônica – NCE, 2003.

Orientador: Pedro Manoel da Silveira

1. Descoberta de Conhecimento em Banco de Dados. 2. Mineração de Dados. 3. Regras de Associação. 4. KDD. 5. Data Mining.

Ao meu querido Carlile.

AGRADECIMENTOS

Ao amado Carlile que em todos os momentos, principalmente nos mais difíceis, me ajudou a seguir em frente e jamais desistir dos meus objetivos. Pela sua infinita compreensão, serenidade, dedicação, companheirismo e amor agradecerei sempre.

Aos meus pais e à minha irmã Paula, pelo amor incondicional a mim dedicado, pela formação do meu caráter, por todo apoio dado em todas as minhas escolhas, por terem me dado a estrutura necessária para trilhar os meus caminhos, por tudo.

À família do Carlile, que agora também é minha, pelo acolhimento caloroso e carinhoso e por torcerem por mim durante toda esta jornada.

A todos os amigos pelo carinho e estímulo, mostrando-me que em momento algum estive só.

Aos familiares que puderam estar presentes em mais esta etapa e que sempre torceram ao meu favor.

Ao meu orientador Pedro Manoel, fundamental pela conquista deste trabalho, pelo incentivo constante que me deu e por todo o apoio técnico.

RESUMO

LAVÔR, Renata Meireles de Paula. **Implementação de serviços relacionados à mineração de regras de associação**. Orientador: Pedro Manoel. Rio de Janeiro: UFRJ/IM, 2003. Dissertação (Mestrado em Informática).

O processo de Descoberta de Conhecimento em Bases de Dados (DCBD) tem por objetivo extrair informações relevantes e desconhecidas, a partir de uma extensa quantidade de dados. Este processo, por sua vez, constitui-se de várias etapas, dentre elas, a atividade de mineração de dados. Esta atividade representa um conjunto de técnicas para obtenção de informações que não podem ser obtidas através de consultas convencionais ou que exigiriam um esforço computacional muito grande. Uma das técnicas é a mineração de regras de associação: padrões descritivos que representam a probabilidade de um conjunto de itens aparecer em uma transação, visto que outro conjunto está presente. Muitos trabalhos relativos a este tema já foram propostos, e, a partir de um levantamento bibliográfico, fez-se uma aplicação de mineração de regras de associação, utilizando como base o algoritmo *Apriori*. O modelo de implementação proposto foi baseado no conceito de disponibilização de serviços. O enfoque principal da implementação é a diversificação da utilidade das regras de associação e caracteriza-se por conter diversos parâmetros que possibilitam selecionar as regras potencialmente mais interessantes. Os serviços implementados foram aplicados na base de dados da Vara de Execuções Penais do Estado do Rio de Janeiro, de forma que o modelo pôde ser validado e os resultados obtidos apresentados.

ABSTRACT

LAVÔR, Renata Meireles de Paula. **Implementação de serviços relacionados à mineração de regras de associação**. Orientador: Pedro Manoel. Rio de Janeiro: UFRJ/IM, 2003. Dissertação (Mestrado em Informática).

The process of Knowledge Discovery in Databases (KDD) targets at extracting relevant and unknown information from an extensive amount of data. This process, however, is made up by several steps, including the activity of data mining. This activity is represented by an array of techniques focused on the gathering of information that would be either impossible to be obtained through conventional queries or require a very large computational effort. One of these techniques is the mining of association rules: descriptive patterns that represent the probability of a collection of items showing up in a transaction once another collection is already present. Plenty of work related to this theme has been done so far. Their bibliographic review led to the application of the mining of association rules based on the use of the *Apriori* algorithm. The proposed implementation model was grounded on the service availability concept. The main focus of the implementation is the diversification of the usefulness of association rules. Its outstanding feature is the fact that it contains several parameters allowing to select those rules which may be potentially more interesting. The implemented services were applied in the database running at the Vara de Execuções Penais in Rio de Janeiro. The model could be therefore validated and the results presented.

LISTA DE SIGLAS

DCBD – Descoberta de Conhecimento em Banco de Dados;

KDD – Knowledge Discovery in Databases;

MD – Mineração de Dados;

TJRJ – Tribunal de Justiça do Estado do Rio de Janeiro;

VEP – Vara de Execuções Penais.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

CAPÍTULO 1. INTRODUÇÃO

A tecnologia atual disponibiliza formas eficientes e econômicas de armazenar grandes volumes de dados, em virtude da queda nos custos de armazenamento, da automatização dos processos de coleta de dados, da disponibilidade de computadores com desempenho cada vez maior, entre outros fatores. Entretanto, a capacidade de armazenar informações supera a capacidade de recuperá-la, pois os métodos tradicionais de transformar os dados em conhecimento baseiam-se em análises manuais, lentas, caras e, em parte, subjetivas. Como o custo da análise cresce com volume de dados e com a complexidade de sua estrutura, torna-se impraticável a manipulação dessas grandes coleções de dados sem a ajuda de técnicas automáticas que permitam a seleção dos principais padrões implícitos em uma base de dados.

A pesquisa em Descoberta de Conhecimento em Bases de Dados (DCBD) é uma área que objetiva encontrar recursos mais eficientes para a recuperação das informações úteis que estão contidas nos bancos de dados. O produto final mais importante do armazenamento dos dados é o conhecimento que pode ser inferido a partir dos mesmos, e, por este motivo, esta área tem atraído esforços em virtude das grandes coleções de dados disponíveis e da dificuldade de extração de informações através dos recursos atualmente disponíveis.

O conhecimento descoberto pode ajudar a camada gerencial e estratégica das instituições nas tomadas de decisões, já que apenas coletar dados não significa possuir informações.

A DCBD é composta por diversas etapas seqüenciais, desde a etapa de seleção dos dados a serem manipulados até a etapa de interpretação das descobertas. A etapa de mineração de dados (MD) é uma das fases da DCBD e é considerada o seu núcleo, sendo composta pela execução de algoritmos de extração de informações implícitas, previamente desconhecidas e potencialmente úteis.

O presente trabalho enfoca o tema de extração de regras de associação, que é uma tarefa específica da área de mineração de dados. As regras de associação representam a probabilidade de que um item apareça em uma transação, visto que outro item está presente. Essas regras foram inicialmente propostas para utilização no comércio varejista

para análise dos itens adquiridos pelos consumidores em uma mesma transação, conhecido como análise de cesta de supermercado (*market basket analysis*). Atualmente, existem diversas propostas para a utilização das regras de associação e por se tratar do foco principal deste trabalho, as regras de associação e suas aplicações serão detalhadas ao longo do mesmo.

1.1 MOTIVAÇÃO

Esta dissertação é relacionada com a extração de regras de associação, que é uma tarefa específica de mineração de dados e que tem sido alvo de muitos estudos recentes. Diversos algoritmos têm sido propostos para esta finalidade, bem como alternativas para lidar com as dificuldades inerentes à tarefa. Em virtude da grande quantidade de informação dispersa, a primeira motivação deste trabalho relaciona-se com o anseio de reunir uma parte das pesquisas e elaborar um levantamento bibliográfico sobre o assunto.

As informações relacionadas com o assunto se encontram dispersas em artigos, teses, dissertações, em ferramentas existentes no mercado, etc. O volume de propostas encontradas motivou a implementação de serviços para apoio à geração das regras de associação reunindo algumas propostas já existentes e novas idéias sugeridas neste trabalho.

A utilização de uma base de dados contendo informações ainda não exploradas na pesquisa sobre regras de associação também motivou a realização do presente trabalho. Esta base de dados contém informações sobre os indivíduos condenados penalmente, cuja execução da pena esteja sob responsabilidade do Poder Judiciário do Estado do Rio de Janeiro.

1.2 OBJETIVOS

Um dos objetivos desta dissertação é realizar um levantamento bibliográfico sobre a fase de mineração de dados do processo de descoberta de conhecimento em base de dados, enfocando, principalmente, a tarefa de extração de regras de associação.

Outro foco deste trabalho é implementar um modelo para extração de regras de associação com as seguintes características:

- será gerado um serviço e não uma ferramenta completa, para que tanto um usuário final quanto outras aplicações, com distintas abordagens, possam utilizar o serviço implementado;
- irá considerar diversas propostas encontradas na literatura e, ainda, os novos recursos apresentados que auxiliarão o especialista;
- permitirá diferentes níveis de participação do operador no processo automatizado de extração de regras;
- poderá lidar com vários parâmetros para manipulação das regras de associação, possibilitando o direcionamento das regras para o objetivo do operador da aplicação;
- poderá gerar outros serviços relacionados com a extração das regras de associação.

Este estudo visa, ainda, apresentar uma utilização prática dos serviços implementados, utilizando para este fim, a base de dados de indivíduos condenados cuja execução da pena esteja vinculada ao Poder Judiciário do Estado do Rio de Janeiro, propondo, desta forma, uma nova oportunidade de aplicação das regras de associação.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está organizada em cinco outros capítulos, dois apêndices e as referências bibliográficas. A estrutura do trabalho apresenta-se da seguinte maneira:

- O Capítulo 2 apresenta uma introdução ao tema mineração de dados, onde são descritas as principais tarefas relacionadas ao tema e as principais características de cada uma delas.
- O Capítulo 3 aborda a tarefa de extração de regras de associação com maiores detalhes.
- O Capítulo 4 explica e detalha os serviços relacionados à extração de regras implementados.
- O Capítulo 5 apresenta estudos de casos e os resultados obtidos na execução dos serviços implementados.

- O Capítulo 6 contém a conclusão deste trabalho, apresentando suas contribuições e sugestões para desenvolvimentos futuros.
- O Apêndice I apresenta as funcionalidades básicas de alguns produtos encontrados no mercado destinados à extração de regras de associação.
- O Apêndice II descreve as bases de dados utilizadas nos estudos de caso do Capítulo 5 desta dissertação.

CAPÍTULO 2. MINERAÇÃO DE DADOS

Este capítulo apresenta os aspectos fundamentais da Descoberta de Conhecimento em Banco de Dados (*Knowledge Discovery in Databases*), descrevendo este processo e detalhando suas etapas, enfocando principalmente a fase de mineração de dados e as principais tarefas a ela associadas.

2.1 DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS

A Descoberta de Conhecimento em Banco de Dados é resultado de um longo processo de pesquisa que se iniciou devido ao grande volume de dados armazenado nos bancos de dados e à complexidade destes dados. Seja pela existência de equipamentos que transformaram de forma expressiva a capacidade de coletar e armazenar dados ou seja pelo acesso e disponibilidade desses equipamentos em larga escala a preços compatíveis com a valia dos processos, a habilidade de armazenar os dados superou em muito a habilidade de analisar e entender os dados armazenados. Desta forma, exigem-se novas técnicas computacionais que suportem a extração automática de conhecimento dos grandes bancos. Essas técnicas, que auxiliam, inteligente e automaticamente, na tarefa de analisar dados na busca de conhecimentos úteis, são o objeto de estudo da área de pesquisa de DCBD.

A DCBD é um conjunto de procedimentos pelo qual se analisa e transforma um conjunto de dados em conhecimento, em padrões interessantes, fazendo utilização de técnicas automáticas para a extração destes padrões.

Diversas etapas compõem a DCBC, onde os dados são manipulados até que a informação útil seja revelada. Para que o processo seja iniciado é necessária a compreensão do domínio da aplicação e dos objetivos a serem explorados. Fazem parte deste processo as seguintes etapas [FAY96]:

- Seleção

  Nesta etapa ocorre a seleção da base de dados relevantes que servirá para todo o processo.

- Pré-processamento

Fase utilizada para a limpeza dos dados, retirada de ruídos ou aberrações dos dados, para integração de dados heterogêneos e para tratar dados incompletos.

- Transformação

  Esta etapa serve para que os dados sejam convertidos para o formato adequado à sua utilização pela fase de mineração de dados. Pode-se realizar, também, a redução de dados, permitindo um número menor de variáveis sob consideração na mineração.

- Mineração dos dados

  Pode ser considerada o núcleo da DCBD, consistindo na aplicação de algoritmos para extração de padrões dos dados. Em virtude de sua importância, o termo mineração de dados é, por vezes, utilizado para identificar todo o processo de DCBD.

- Interpretação

  Fase onde ocorre a interpretação correta dos resultados obtidos pela mineração para posterior consolidação do conhecimento e a viabilização de uma utilização prática do mesmo.

Esta apresentação das atividades pode sugerir que exista uma trajetória linear do processo de DCBD. No entanto, isso geralmente não se verifica, uma vez que em cada etapa pode ser identificada a necessidade de retorno para cada uma das etapas anteriores, sendo portanto um processo iterativo e interativo, também, pois necessita da intervenção de um especialista em todas as fases da descoberta. As etapas descritas anteriormente estão representadas na ilustração a seguir [Figura 2.1]:

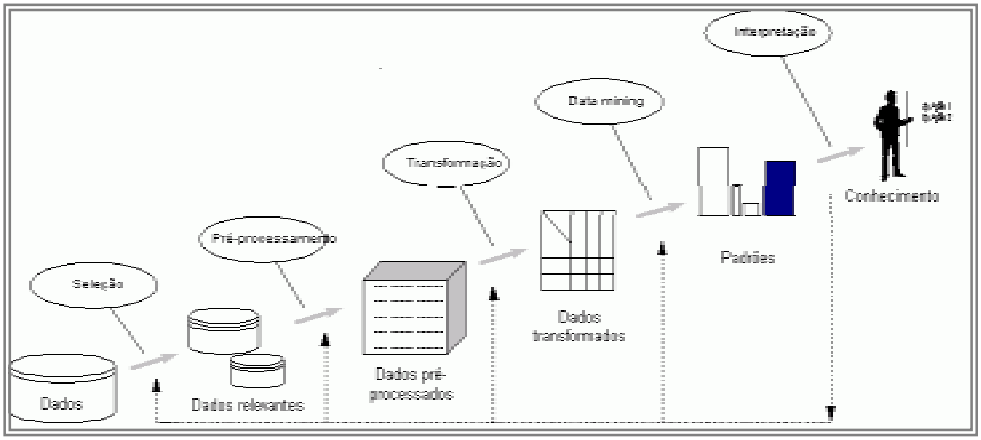


Figura 2.1 – O processo de Descoberta de Conhecimento em Banco de Dados

2.2 A MINERAÇÃO DE DADOS E SUAS PRINCIPAIS TAREFAS

A mineração de dados (*data mining*) se refere a um passo particular no processo de DCBD e este passo é relativo à aplicação de algoritmos específicos para a extração de padrões dos dados, ou seja, a explicitação de relações entre elementos de um banco de dados (registros, atributos e seus domínios). A importância das técnicas de MD está relacionada com sua capacidade de extrair conhecimento útil e previamente desconhecido e que está implícito nas relações entre os dados do banco [FAY96].

A MD tem sido utilizada em diversos tipos de aplicações, incluindo o *marketing*, a segmentação de mercado, a detecção e predição de fraudes, análises financeiras, entre outras [GON01].

Através da MD, um usuário pode descobrir padrões e construir modelos, automaticamente, sem saber exatamente o que se está procurando. Os modelos podem ser tanto de descrição ou de predição, ou seja, são descobertas como as coisas acontecem e o que pode estar para acontecer. Além disso, as suas ferramentas podem responder questões que tradicionalmente consumiriam muito tempo, a um custo muito alto. Entretanto, para que o conhecimento extraído possa vir a ser utilizado pelos usuários finais, a forma de representar este conhecimento deve ser acessível para os mesmos. Exemplos comuns de questionamentos que podem ser respondidos pelas ferramentas de mineração de dados: “Qual é a expectativa de vida da conta corrente de um cliente?”, “Quais são os prováveis consumidores de um determinado produto?”, “Este consumidor possui características que o levem a ser um mau pagador?”, etc.

Muitas ferramentas tradicionais de análise suportam uma abordagem baseada na verificação, na qual o usuário cria hipóteses em relação aos dados e utiliza-se da ferramenta para validar ou contestar as mesmas. Esta abordagem difere da utilização da mineração de dados, já que ao invés do usuário propor hipóteses, a própria mineração gera as proposições garantindo mais eficácia ao processo [GON01].

Os padrões que podem ser obtidos na fase de mineração de dados podem estar relacionados com a predição de valores desconhecidos ou com a descrição de subgrupos do banco de dados, usando a base de dados selecionada como origem destas descobertas. A cada tipo de padrão está relacionada uma tarefa específica de mineração de dados e existem diversos algoritmos que podem ser executados para atender a cada uma destas tarefas.

Existem diversas tarefas de extração de padrões. Pode-se considerar, entretanto, quatro tarefas genéricas: análise de dependências, identificação de classes, descrição de classes e detecção de desvios [MAT93]:

- Análise de Dependências

  Uma dependência existe entre dois itens se o valor de um deles pode ser utilizado para prever o valor do outro. Podemos dividir a tarefa de análise de dependências em duas: a primeira, a tarefa de gerar regras de associação e, a segunda, a tarefa de detecção de seqüências temporais.

- Identificação de classes

  A tarefa de identificação de classes consiste em selecionar os registros do banco de dados e agrupá-los de acordo com algum critério. Podemos dividir a tarefa em classificação, regressão e clusterização.

- Descrição de classes

  Esta tarefa se apóia no interesse em determinar características de registros individuais de uma classe, identificar uma descrição abstrata que sumariza qualidades interessantes sobre a classe, discriminar como as classes diferem entre si. Esta tarefa é geralmente chamada de sumarização.

- Detecção de desvios

  A detecção de desvios é a extração de regras do banco de dados que se apresentam como extremos. Por exemplo, instâncias que não se encaixam em classes padrões, *outliers* que podem vir a formar novas classes, etc.

2.2.1 REGRAS DE ASSOCIAÇÃO

A descoberta de regras de associação consiste em encontrar padrões que descrevam dependências significativas entre eventos que ocorrem juntos. Pode se apresentar em dois aspectos: o nível estrutural, onde são especificadas variáveis localmente dependentes de outras e o nível quantitativo, que especifica a força das dependências utilizando algum critério numérico [FAY96].

Geralmente, os problemas envolvem centenas ou milhares de itens no banco de dados, que resultam em um número imenso de combinação entre eles. O grande desafio das associações consiste em encontrar combinações significativas entre os dados sem ser preciso examinar um número excessivo de combinações que são pouco úteis [INF96].

O uso mais comum da tecnologia de associação pode se dar num negócio de venda de itens ao público. Um exemplo de aplicação bem sucedida de mineração de dados é baseado na experiência da rede americana *Wal-Mart*, que descobriu ao explorar seus números, que 60% das mães que compram a boneca *Barbie* nas lojas da rede também levam uma barra de chocolate. Baseado nesta informação, as gôndolas foram posicionadas de modo que as prateleiras de doces e chocolates ficassem próximas das de brinquedos. As vendas de guloseimas e brinquedos deram um salto [FEL99].

Outro exemplo de aplicação é a verificação de redundância de pedidos de exames médicos que são solicitados em conjunto. Os pedidos que são redundantes poderão ser eliminados após análise de especialistas, trazendo economia e maior agilidade no atendimento [INF96].

Os resultados das associações são geralmente apresentados em forma de regras *SE-ENTÃO*. Podem ser representados, ainda, em forma de gráficos de dependência ou de forma tabular. A seguir, representamos nessas três formas o exemplo anterior da rede *Wal-Mart*.

- Regras

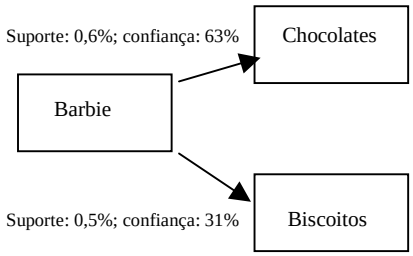
As regras do tipo *SE X ENTÃO Y*, ou simplesmente, $X \rightarrow Y$, possuem dois elementos: o antecedente, *X*, e o conseqüente, *Y* [AGR93.1]. Geralmente essas regras vêm associadas a valores de suporte e de confiança dos elementos da regra. O suporte indica a porcentagem de ocorrências deste tipo de associação dentre o montante total de registros e a confiança indica a porcentagem de todas as ocorrências do antecedente onde o item conseqüente está associado [LEV99].

Exemplo:

$\text{Barbie} \rightarrow \text{Chocolate}$

Suporte (s): 0,6%; confiança (c): 63%

- Gráficos de Dependência



- Tabelas

Antecedente	Conseqüente	Suporte %	Confiança %
Barbie	Chocolates	0,6	63
Barbie	Biscoitos	0,5	31
Barbie Chocolates	Biscoitos	0,3	22

Em princípio, os algoritmos para extração das regras funcionam a partir da análise das combinações dos dados do conjunto a ser pesquisado. Esta operação cresce de forma exponencial em função do número de itens a ser comparado. Logo, se a confiança de dois itens pode ser feita manualmente, a confiança de cinco itens já se torna uma tarefa muito complexa para ser apurada manualmente, devido ao grande número de consultas que deverão ser realizadas e ao enorme tempo gasto em executar tais consultas.

Como o estudo das regras de associação é o objeto deste trabalho, o Capítulo 3 é relacionado a este assunto.

Técnicas originadas nas áreas de Estatística, Teoria de Conjuntos, Modelos Probabilísticos de gráficos de dependência são as técnicas mais utilizadas para a extração de regras de dependência [FAY96, BIG96].

2.2.2 DETECÇÃO DE SEQÜÊNCIAS

A detecção de seqüências consiste na descoberta de padrões seqüenciais dos itens do banco de dados [FAY96.2], descrevendo a tendência de certos eventos ocorrerem obedecendo a uma determinada seqüência temporal. Enquanto a associação encontra eventos que ocorrem concomitantemente, a detecção de seqüências encontra eventos relacionados que ocorrem ao longo de um período de tempo [INF96] [LEV99].

A tarefa de detecção de seqüências também pode se dar em um negócio de venda de itens. Por exemplo, pode ser verificado após a análise do banco de dados que se um cliente compra um celular do modelo *XYZ* então em 78% das vezes abrirá um chamado reclamando do carregador de baterias. Ainda como exemplo, pode-se verificar que se um cliente compra um produto *X* então em 40% das vezes compra o produto *Y* dentro de 1 semana.

A detecção de seqüências também pode ser útil na área médica, onde o histórico do paciente pode ser avaliado para a descoberta de prováveis doenças futuras. A seguir é apresentado um exemplo da extração de possíveis regras de padrões seqüenciais.

Considere um banco com dados históricos do paciente *X* representado conforme a estrutura da Tabela 2-1 seguinte, onde as colunas representam um tempo específico ($t_n < t_{n+1}$) e as linhas exibem os valores dos atributos considerados pela técnica de descoberta em cada um dos tempos.

t_1	t_2	t_3
Idade=50	Idade=50	Idade=50
Sexo=M	Sexo=M	Sexo=M
Fumante=S	Fumante=S	Fumante=S
Diabetes= N	Diabetes= N	Diabetes= S
Eletrocardiograma= Anormal	Eletrocardiograma= Anormal	Eletrocardiograma=
Parada Cardíaca=N	Parada Cardíaca=N	Parada Cardíaca=S

Tabela 2-1 - Banco com dados históricos de paciente [LEV99]

Uma possível regra a ser extraída do exemplo acima poderia ser: *"SEXO=M e IDADE>=50 e FUMA=S e ELETROCARDIOGRAMA=Anormal (2 X) → PARADA CARDÍACA = S"*.

Na literatura, encontramos referências relacionadas à Estatística e à Teoria de Conjuntos como algumas das técnicas utilizadas para a detecção de padrões seqüenciais [FAY96, BIG96].

2.2.3 REGRESSÃO

O objetivo da técnica de regressão é identificar um padrão preditivo de comportamento dos itens de um banco de dados a partir dos dados já existentes. Para alcançar este objetivo, a técnica de regressão visa descobrir uma função que possa mapear os itens de dados em um valor retornado pela função descoberta. Esta função é utilizada como uma forma de prever o valor a ser assumido por um novo item do banco de dados [FAY96].

Prever a demanda de consumo de um produto, sendo o valor desta demanda calculado através de uma função que leve em consideração os gastos em publicidade, pode ser um exemplo da tarefa de regressão [FAY96]. Uma companhia de cartão de crédito pode se utilizar da técnica para, através de algumas características do indivíduo contidas no banco de dados, prever o nível de risco de inadimplência do mesmo. As características a serem consideradas na função poderiam ser a renda, o tipo de emprego, o histórico de crédito, a idade, etc. [INF96].

Pode-se estimar o valor do crédito máximo a ser financiado a um indivíduo usando uma função linear em relação à renda do cliente. Na Figura 2.2, os pontos *0* indicam que o cliente possui bom crédito e os pontos *1* indicam um crédito ruim. A representação da tarefa de regressão pode se dar através da especificação de funções lineares ou não-lineares.

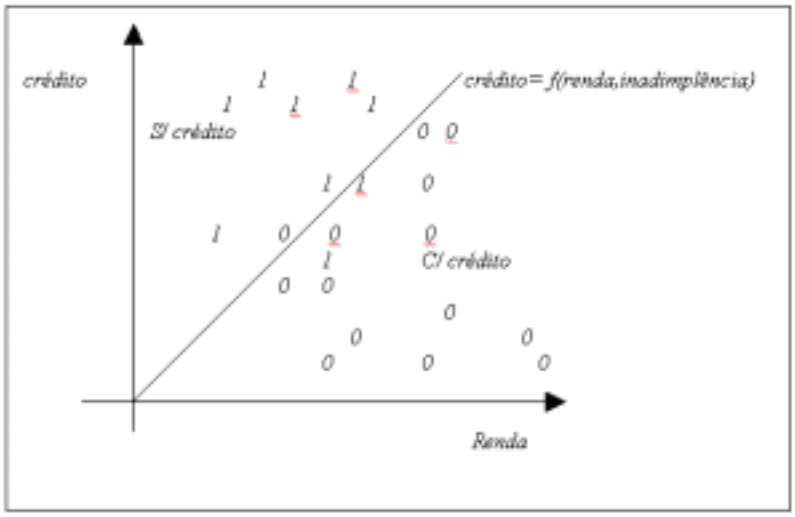


Figura 2.2 – Representação de uma função linear para a tarefa de regressão

Dentre as técnicas utilizadas para a tarefa de regressão pode-se citar o uso de redes neurais, o ajuste de curvas e as árvores de decisão [FAY96].

2.2.4 CLASSIFICAÇÃO

A finalidade da classificação também é, tal como a regressão, identificar padrões que possam prever o comportamento de novos itens de um banco de dados a partir dos dados já existentes. Entretanto, esta técnica visa mapear um novo item de dados em uma ou mais classes pré-definidas, e não em um determinado valor. O item será mapeado em uma determinada classe de acordo com sua maior semelhança com a característica da classe, sendo possível que qualquer atributo ou conjunto de atributos do banco de dados possa ser utilizado para definir as classes [MAT93, AGR93.2, FAY96].

Algumas classes já são de domínio geral, tal como o campo *REGIÃO* de um banco de dados: *Norte*, *Nordeste*, *Centro-Oeste*, *Sudeste* e *Sul*. Para essas classes citadas, a tarefa de

classificação poderia identificar um padrão que classificasse uma cidade em uma determinada região baseada em informações sobre a sua posição geográfica.

Existem, entretanto, outros exemplos onde as classes não são triviais. Por exemplo, a classe de indivíduos que poderão conseguir um empréstimo automático em uma financeira, a classe de indivíduos que deverão ir à gerência para melhor avaliação de suas características e a classe dos indivíduos com a negação de crédito, etc. A partir de um banco de dados com as características de indivíduos que já foram avaliados para a obtenção de empréstimo, o objetivo da tarefa de classificação seria encontrar o perfil dos indivíduos em cada uma das classes. Encontrado esse perfil, um novo indivíduo que tentasse obter um empréstimo poderia ser classificado em uma das classes mencionadas anteriormente baseando-se em suas características pessoais.

Uma variação no problema de classificação é o problema chamado de *BestN*. Uma companhia pode estar interessada em encontrar os melhores *N* clientes para enviar proposta de aquisição de um novo produto. Primeiramente, um pequeno número de propostas é enviado para que seja selecionado o perfil das pessoas com resposta positiva obtida. Após este passo, cria-se um grupo de confiança com este perfil e outro com perfil diverso. A partir de então são identificados dentre os indivíduos mapeados na classe de resposta positiva os melhores *N* clientes para o envio da proposta, de acordo com um grau de confiança adotado [AGR93.2].

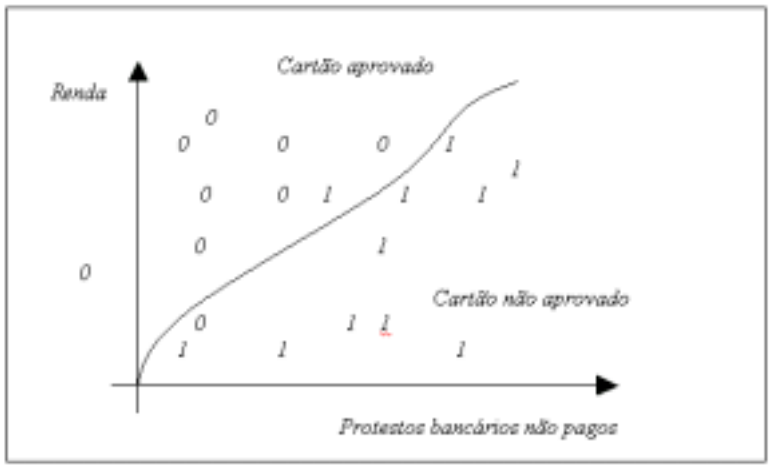


Figura 2.3 – Representação de uma função não linear para a tarefa de classificação

No exemplo anterior [Figura 2.3], foi realizada uma classificação não-linear de um banco de dados. Se um novo item do banco de dados for incluído, poderá ser classificado na região acima da curva, que é a classe de indivíduos que possuem o cartão de crédito aprovado, ou abaixo da curva, que é a classe dos que não possuem cartão de crédito aprovado.

Árvores de decisão, redes neurais, métodos baseados em exemplos (por exemplo, classificação baseada no vizinho mais próximo) são exemplos de técnicas utilizadas na tarefa de classificação [FAY96, BIG96].

2.2.5 CLUSTERIZAÇÃO

A técnica de clusterização classifica a informação em conjuntos homogêneos baseando-se em atributos específicos, cabendo ao algoritmo explorar diferentes alternativas e detectar os padrões [MOR98]. É uma tarefa descritiva utilizada para identificar um conjunto finito de categorias (*clusters*), onde essas categorias são determinadas a partir dos dados. Ao contrário da classificação, onde as categorias são pré-definidas, os *clusters* são definidos pelo agrupamento natural de itens de dados baseados em similaridades entre tais itens [FAY96.2]. Os membros das categorias são inicialmente desconhecidos e as categorias podem ser mutuamente exclusivas ou não. O mais comum é agrupar *clusters* que possuem, ao mesmo tempo, a máxima homogeneidade interna e a máxima diferenciação em relação a todos os outros clusters.

A clusterização pode, por exemplo, agrupar clientes com as mesmas características de consumo. Os hábitos destes consumidores podem, então, ser comparados para determinar quais são os segmentos que serão selecionados para a realização de uma nova campanha de venda.

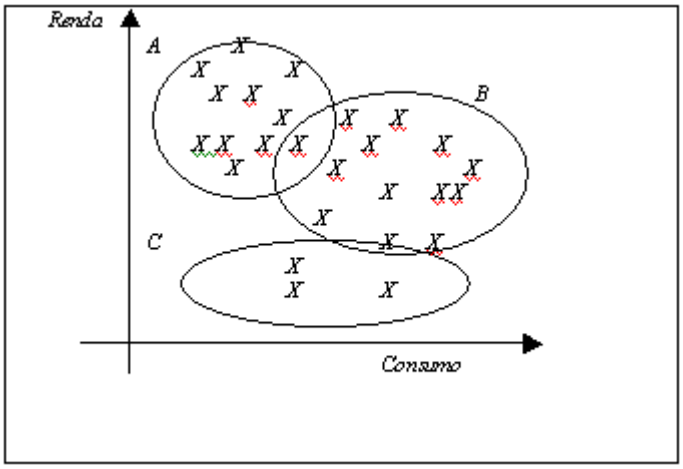


Figura 2.4 – Representação de 3 clusters encontrados em função das variáveis renda e consumo

Na Figura 2.4 exibimos uma clusterização de um banco de dados. Foram representados 3 *clusters*: *A*, *B* e *C*. Os *clusters* são definidos pela relação dos campos consumo e renda dos clientes do banco de dados. Cada um dos elementos possui maior semelhança com os membros do *cluster* onde estão inseridos e maior distinção com os outros, embora alguns destes elementos (clientes) estejam inseridos em mais de um *cluster*.

Citamos as Árvores de Decisão, as Redes Neurais e os Métodos baseados em exemplos (classificação em relação ao vizinho mais próximo) como técnicas para a clusterização de base de dados [FAY96, BIG96, MED99].

2.2.6 DESCRIÇÃO DE CLASSES (SUMARIZAÇÃO)

A sumarização é utilizada para encontrar uma descrição compacta de uma classe, encontrando as qualidades interessantes da mesma. Existem duas formas de obtenção destas descrições das classes: a caracterização e a discriminação. A caracterização descreve o que os registros de uma classe possuem em comum entre eles, não levando em consideração as demais classes existentes. Já a discriminação descreve como duas ou mais classes diferem entre si [MAT93, AGR93.2].

A tarefa de sumarização pode ser utilizada para a descrição de qualquer classe relativa ao banco de dados: descrição da classe de vendedores mais produtivos de uma firma, descrição dos consumidores de bebidas alcoólicas, etc.

Considere um banco de dados onde existam informações sobre o pagamento de faturas de cartões de crédito e dados pessoais dos titulares dos cartões com as seguintes classes:

- Classe *A* - Pagamentos de 60% das faturas.
- Classe *B* - Pagamentos em dia.
- Classe *C* - Pagamentos das faturas em atraso.

A representação da caracterização da classe *C* poderia ser a seguinte: BAIRRO = "TIJUCA, MÉIER, LARANJEIRAS"; GRAU DE INSTRUÇÃO = "2º grau completo, 3º grau incompleto".

A representação da discriminação das Classes *A*, *B* e *C* poderia ser a seguinte: Se BAIRRO = "LARANJEIRAS" então Classe B; Senão se RENDA >= 10.000 então Classe A; Senão Classe C.

Neste exemplo, o resultado do algoritmo de sumarização é proposto na forma de regras [MAT93]. Outras formas de representação envolvem técnicas de visualização ou de relações funcionais entre os campos [FAY96.2, FAY96].

Árvores de decisão, redes neurais e algoritmos genéticos são exemplos de técnicas associadas à tarefa de sumarização [FAY96.2, FAY96].

2.2.7 DETECÇÃO DE DESVIOS

A detecção de desvios é utilizada para detectar anomalias em bases de dados, podendo evidenciar problemas de qualidade de dados ou descobrir eventos raros. Dado um conjunto de dependências, seqüências ou descrições de conceitos, que podem ser obtidos automaticamente ou através do usuário, o algoritmo procura os elementos contidos no banco de dados que estão fora destes padrões [FEL99]. Por exemplo, instâncias com anomalias que por esta razão não se enquadram em nenhuma classe, *outliers* que podem dar início à criação de uma nova classe, classes que possuem um valor médio de seus elementos significantemente distinto das classes vizinhas, mudanças bruscas em um determinado valor ou conjunto de valores de um momento para o outro, discrepâncias entre o valor assumido de um atributo e o valor previsto para o mesmo, etc.

O denominador comum entre os métodos de descoberta destes tipos de padrões é a procura de diferenças significativas entre o valor assumido pelo atributo e alguma referência que pode ser, por exemplo, um valor calculado por um modelo aplicado aos dados, o valor mensurado anteriormente ou, ainda, algum valor normativo [MAT93, FAY96].

A descoberta de desvios no banco de dados pode ser utilizada em diversas situações: descoberta do caso de um homem submetido a uma cesariana [FEL99], descoberta da utilização exagerada de combustível em uma determinada rota, descoberta de ligações clandestinas de energia elétrica, etc.

Considere o modelo de referência para uma tarefa de detecção de desvios dado pela figura abaixo:

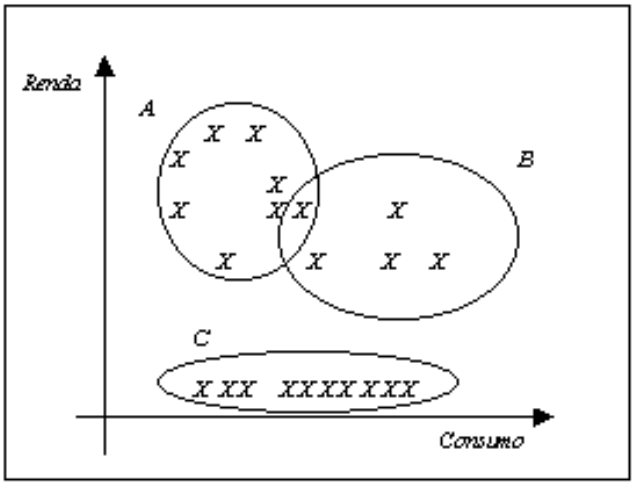


Figura 2.5 – Representação de modelo de referência com as classes A, B e C para a tarefa de detecção de desvios [MAT93]

A Figura 2.5 representa um banco de dados de consumidores e seus atributos renda e consumo. Neste banco de dados estão representadas três classes fictícias nomeadas de *A*, *B* e *C*. Na tarefa de detecção de desvios este banco de dados servirá de modelo de referência para que sejam encontradas diferenças significativas entre este modelo e um novo banco de dados a ser testado.

Para um banco de dados a ser testado com referência ao modelo anteriormente exibido, os algoritmos de detecção de desvios poderiam encontrar diversas divergências entre o banco de dados avaliado e a referência utilizada:

Considere um banco de dados como o da Figura 2.6 e que este banco de dados seja submetido à tarefa de detecção de desvios utilizando como modelo o banco da Figura 2.5. A detecção de desvios irá indicar algumas divergências entre os modelos. Como forma de exemplificar, algumas divergências poderiam ser identificadas: a classe *A* sofreu uma mudança em sua definição, a classe *C* teve sua densidade decrescida, ocorreu o aparecimento de um *outlier* que poderá dar origem a uma nova classe. Essas divergências são consideradas desvios neste novo banco de dados.

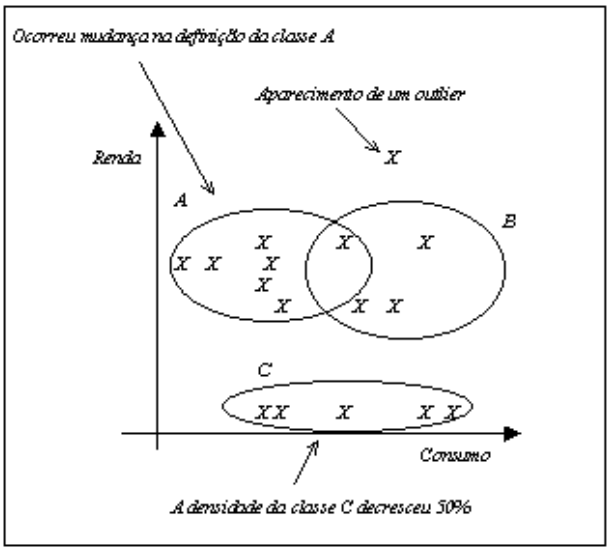


Figura 2.6 – Resultados da aplicação dos algoritmos de detecção de desvios [MAT93]

A Teoria de Conjuntos e a Estatística são as principais técnicas para a implementação da descoberta de desvios.

2.3 RESUMO DAS TAREFAS DE MINERAÇÃO DE DADOS

Resumindo, o processo de DCBD consiste em vários passos de descoberta que permitem que padrões sejam extraídos do banco de dados de forma automatizada. Entre esses passos, podemos citar o da mineração de dados, que é justamente aquele onde são aplicados os algoritmos de descoberta de conhecimento. Os algoritmos da mineração de dados são específicos para cada tipo de tarefa que se deseja aplicar. A seguir mostramos um quadro com as principais tarefas que podem ser aplicadas na fase de mineração de dados.

GRUPO DE TAREFAS	TAREFA	DESCRIÇÃO	TIPO DA TAREFA	MÉTODOS
Análise de Dependências	Associação	Consiste na descoberta de regras que descrevem dependências significativas entre os itens de um banco de dados que ocorrem na mesma transação.	Tarefa descritiva	Estatística, teoria de conjuntos, modelos probabilísticos de gráficos de dependência.
	Detecção de Seqüências Temporais	Consiste na descoberta de regras que descrevem dependências entre itens que ocorrem ao longo do tempo.	Tarefa descritiva	Estatística, teoria de conjuntos.
Identificação de classes	Regressão	Consiste em identificar um padrão de comportamento dos itens do banco de dados, descobrindo uma função que possa mapear os novos itens em um valor retornado pela função descoberta.	Tarefa preditiva	Regressão linear, redes neurais, ajuste de curvas, árvores de decisão.
	Classificação	Consiste em identificar um padrão de comportamento dos itens do banco de dados, visando mapear os novos itens em classes pré-existentes.	Tarefa preditiva	Árvores de decisão, redes neurais, métodos baseados em exemplo.
	Clusterização	Consiste em gerar e identificar grupos, clusters, a partir do grupamento natural de itens do banco de dados de acordo com a similaridade entre eles.	Tarefa descritiva	Árvores de decisão, redes neurais, métodos baseados em exemplo.
Descrição de classes	Sumarização	Consiste em descobrir regras que descrevam sucintamente uma classe.	Tarefa descritiva	Árvores de decisão, redes neurais, algoritmos genéticos.
Detecção de desvios	Detecção de desvios	Consiste em detectar desvios e anomalias dos itens em um banco de dados.	Tarefa descritiva	Teoria de conjuntos, estatística.

Tabela 2-2 - Principais tarefas de mineração de dados

Essas são as tarefas habitualmente exploradas na literatura que cobrem de forma genérica todo o escopo de descoberta em banco de dados na fase de mineração de dados. Os

métodos aplicados às mesmas também são os mais utilizados, embora outros métodos possam ser testados e empregados para implementar cada tarefa.

CAPÍTULO 3. REGRAS DE ASSOCIAÇÃO

Este capítulo apresenta uma revisão bibliográfica de assuntos relacionados com a tarefa de extração de regras de associação de base de dados.

3.1 INTRODUÇÃO

Uma importante tarefa de mineração de dados é a descoberta de regras de associação. Essa tarefa é capaz de identificar grupos de atributos de um banco de dados tipicamente relacionados, representando a probabilidade de que um grupo apareça em uma transação visto que outro está presente na mesma.

Embora a descoberta de regras de associação possa ser aplicada a qualquer base de dados, foi inicialmente proposta para a análise de cesta de produtos (*market basket analysis*). Este tipo de base de dados é composta pelos itens adquiridos pelo cliente em uma determinada cesta de compras, ou seja, em uma mesma transação.

Muitas decisões de *marketing* podem ser tomadas após o estudo dessas transações, auxiliando na reorganização das lojas, na produção de catálogos customizados, na seleção de produtos para promoção, na determinação de itens para uma propaganda mais inovadora, na seleção de artigos concorrentes, no fomento de *cross-selling*, na identificação de oportunidades de vendas de pacotes de produtos ou serviços, etc. [BRU00, JOR03, GON01].

Além da análise utilizada para o *marketing*, diversos estudos já foram realizados propondo a utilização da tarefa de regras de associação em diversas áreas: serviços bancários, telecomunicações, saúde, análise de páginas da *WEB*, entre outros [BRU00, JOR03, SRI97].

Informalmente, as regras de associação podem ser vistas como um tipo de regra SE-ENTÃO [SIE03]. O item 2.2.1 do Capítulo 2 tem por objeto a exposição de alguns exemplos de regras de associação.

3.2 DESCRIÇÃO FORMAL DO PROBLEMA

Em sua forma original, a tarefa de descobrir regras de associação foi definida para um tipo especial de dados, freqüentemente chamado de "*market basket data*". Este tipo de banco de dados é composto por um conjunto de transações onde cada transação é composta por um conjunto de itens. Traduzindo para o ambiente relacional, este banco é representado por uma tabela *T* composta de atributos binários. Os atributos correspondem aos itens e os registros na tabela correspondem às transações. Os atributos binários assumem valor *1*, caso o item esteja presente na transação, ou valor *0*, caso o item não esteja presente na transação [SIE03].

A Figura 3.1 representa a tabela *T* composta do atributo $t_{ID,}$ identificador da transação e de atributos binários $I_1,...,I_{12}$ que representam os itens.

t_{ID}	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}
1	1	1	0	0	1	0	0	0	0	0	0	0
2	0	1	0	0	1	1	0	0	0	1	1	1
3	0	0	0	0	0	1	0	0	1	0	0	0
4	0	1	1	0	0	1	0	0	1	0	0	0
5	0	0	0	0	0	1	0	0	0	1	0	0
6	0	0	0	0	0	1	0	0	0	0	1	0
7	0	1	0	0	0	1	0	0	0	0	0	0
8	1	0	0	0	0	1	0	0	0	1	0	0
9	0	0	0	0	0	1	0	0	1	0	0	0

Figura 3.1 – Tabela *T*

Uma regra de associação consiste em uma expressão da forma $X \rightarrow Y$, onde *X* e *Y* são conjuntos de itens [AGR93.1]. Ou seja, $X_1 \wedge ... \wedge X_m \rightarrow Y_1 \wedge ... \wedge Y_n$, onde $X_i (i \in \{1, m\})$ e $Y_j (j \in \{1, n\})$ representam os itens, ou seja, são os atributos do banco de dados [CHEN96]. Intuitivamente, as regras significam que as transações do banco que contêm *X* tendem a conter *Y*.

Um modelo formal para representar o problema de descoberta de regras de associação foi proposto por Agrawal, Imielinski e Swami em 1993 [AGR93.1]. Considere um conjunto de atributos binários $I = \{I_1, I_2, \ldots, I_m\}$, chamado de itens. Seja

D um conjunto de transações, onde cada transação $t \subset I$ é um conjunto de itens, chamado *itemset*. Cada transação é associada a um identificador único, chamado t_{ID}. Seja $X \subset I$ um conjunto de itens. Uma transação *t* contém *X* se $X \subset t$. Uma regra de associação é um relacionamento na forma $X \rightarrow Y$, onde $X \subset I,\ Y \subset I\ e\ X \cap Y = \phi$.

A regra $X \rightarrow Y$ possui suporte *s* no conjunto de transações *D* se *s*% das transações em *D* contêm *X* e *Y* [AGR93.1, AGR96]. O suporte é a probabilidade de uma transação em *D* conter $X \cup Y$, ou seja, ele indica a freqüência da regra. O suporte $s(X \cup Y)$ é dado, então, por

$$quantidade(X \cup Y) / tamanho(D)$$

$$= | \{t \in D \mid X \subset t, Y \subset t\} | / | D | .$$

A regra $X \rightarrow Y$ é válida no conjunto de transações *D* com confiança *c* se *c%* das transações do banco de dados que contêm *X* também contêm *Y*. A confiança é a probabilidade de *Y* ocorrer em uma transação de *D* visto que *X* ocorre, ou seja, indica a "força" da regra. A confiança é dada por

$$suporte(X \cup Y) / suporte(X) .$$

3.3 DECOMPOSIÇÃO DA GERAÇÃO DE REGRAS

A tarefa de descobrir regras de associação consiste em gerar todas as regras que possuam suporte maior ou igual a um suporte mínimo e uma confiança maior ou igual a uma confiança mínima, sendo o suporte mínimo e a confiança mínima especificados pelo usuário. Esses parâmetros servem como medida capaz de filtrar os padrões inúteis, sendo trabalhados apenas padrões significativos [LOP99].

O problema de geração de regras de associação é geralmente decomposto em dois subproblemas [AGR93.1, AGR96]:

1) Descobrir todas as combinações de itens que tenham suporte maior ou igual ao mínimo previamente especificado. Essas combinações de itens são chamadas de conjunto de itens freqüentes (*large itemsets*), e as outras combinações de conjunto de itens raros (*small itemsets*). Se a combinação de itens freqüentes possui *k* itens, então, é designada por *k-itemset*.

2) Gerar as regras de associação do banco de dados utilizando os conjuntos de itens freqüentes. A regra só será válida se a confiança da mesma for maior ou igual à mínima predeterminada.

A performance da tarefa de mineração de regras de associação é de fato determinada pelo subproblema 1. As regras de associação correspondentes podem ser derivadas de uma maneira direta. Logo, a descoberta dos conjuntos de itens freqüentes é o foco maior dos trabalhos encontrados [PAR97].

Na literatura, são encontrados diversos algoritmos de descoberta dos conjuntos de itens freqüentes: *AIS* [AGR93.1], *SETM* [HOU93], *Basic* [MAN94], *Apriori* [AGR94], *AprioriTid* [AGR94], *AprioriHybrid* [AGR94], *DHP* [PAR97], *ECLAT* [ZAK97], *FPGrowth* [HANPEI00], *DIC* [BRIN97], entre diversos outros.

3.4 ALGORITMOS

Os algoritmos para descoberta dos conjuntos de itens freqüentes percorrem a base de dados diversas vezes. No primeiro passo, contam o suporte de cada item individualmente e determinam dentre eles quais são freqüentes, isto é, possuem o suporte mínimo especificado. Utilizando os itens freqüentes do passo anterior e algum critério específico do algoritmo, em cada passo subseqüente são gerados os novos potenciais itens freqüentes, chamados de conjunto de itens candidatos. Para cada um dos conjuntos de itens candidatos, o suporte é calculado ao percorrer a base de dados novamente. No final do passo, são determinados quais itens candidatos são realmente freqüentes. Estes, por sua vez, servirão para a geração dos novos candidatos. Este processo continua até que não exista mais nenhum conjunto de itens freqüentes [AGR94].

Para a exposição de alguns algoritmos de descoberta de itens freqüentes, considere o banco de dados *D* a seguir [Figura 3.2], onde t_{ID} é o identificador da transação e a coluna *Itens* é o conjunto de itens encontrados na transação. Os itens existentes em *D* são: A, B, C, D, E e F. O suporte mínimo requerido é de 50%. Como *D* possui 6 transações, exigem-se três transações para que o conjunto de itens seja considerado freqüente.

t_{ID}	*Itens*
t1	A C D E
t2	A B C D F
t3	A D E
t4	D
t5	A B
t6	A B D E

Figura 3.2 – Banco de dados *D*

Para os exemplos a seguir, considere F_k a lista de conjuntos de itens freqüentes com *k* itens (*k-itemset*), e *F* a lista de todos os itens freqüentes. Como existem seis atributos em *D*, *k* irá variar de 1 até 6.

3.4.1 ALGORITMO DE SIMPLES COMBINAÇÃO DE ITENS

A Figura 3.3 exibe o algoritmo de simples combinação de itens. Em cada passo *k* do algoritmo, C_k é gerado a partir de todas as combinações de *k* atributos do banco de dados. Em seguida, a base de dados é percorrida e o suporte de cada candidato em C_k é computado. Por fim, todos os candidatos com suporte maior que o mínimo são registrados como freqüentes.

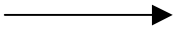
1) Para cada $k = 2$ $(passo = 1)$ faça
2) $C_k = \begin{pmatrix} atributos \\ k \end{pmatrix}$
3) Para todas as transações $t \in D$ faça
4) C_t = subconjunto (C_k , *t*); // Candidatos contidos em *t*
5) Para todos os candidatos $c \in C_t$
6) *c.suporte*= *c.suporte* +1
7) $L_K = \{c \in C_k \mid c.suporte \geq suporte\_mínimo\}$
8) Se $L_k \neq \phi$ então $F = F \cup L_K$
Senão Fim

Figura 3.3 – Algoritmo de simples combinação de itens

A seguir, cada passo do algoritmo é mostrado.

Passo $k = 2$:

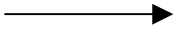
C_2	Suporte
{A B}	3
{A C}	2
{A D}	4
{A E}	3
{A F}	1
{B C}	1
{B D}	2
{B E}	1
{B F}	1
{C D}	2
{C E}	1
{C F}	1
{D E}	3
{D F}	1
{E F}	0



F_2	Suporte
{A B}	3
{A D}	4
{A E}	3
{D E}	3

Passo $k = 3$:

C_3	Suporte
{A B C}	1
{A B D}	2
{A B E}	1
{A B F}	1
{A C D}	2
{A C E}	1
{A C F}	1
{A D E}	3
{A D F}	1
{A E F}	0
{B C D}	1
{B C E}	0
{B C F}	1
{B D E}	1
{B D F}	1
{B E F}	0
{C D E}	1
{C D F}	1
{C E F}	0
{D E F}	0



F_3	Suporte
{A D E}	3

Passo $k = 4$:

C_4	Suporte
{A B C D}	2
{A B C E}	1
{A B C F}	1
{A B D E}	1
{A B D F}	1
{A B E F}	0
{A B E F}	0
{A C D E}	1
{A C D F}	1
{A C E F}	0
{A D E F}	0
{B C D E}	1
{B C D F}	1
{B C E F}	0
{B D E F}	0
{C D E F}	0

$\longrightarrow \quad F_4 = \phi$

Como $F_4 = \phi$, o critério de parada do algoritmo é satisfeito, logo, $F = F_2 \cup F_3$.

F	Suporte
{A B}	3
{A D}	4
{A E}	3
{D E}	4
{A D E}	3

3.4.2 *APRIORI*

A Figura 3.4 descreve o algoritmo *Apriori*, proposto por Rakesh Agrawal e Ramakrishnan Srikant [AGR94]. Inicialmente, o conjunto F_1 é identificado. F_1 é a lista de conjuntos de itens freqüentes com apenas um elemento. Em seguida, em cada passo k do algoritmo, C_k é gerado utilizando a lista F_{k-1}, onde F_{k-1} é a lista de conjuntos de itens freqüentes com $k-1$ elementos. Nesta etapa, a base de dados é percorrida para calcular o suporte de cada candidato. Todos os candidatos com suporte maior que o mínimo são incluídos na lista de itens freqüentes.

1) F_1 = {1-itemset}
2) Para cada $k = 2; L_{K-1} \neq 0; (passo = 1)$ faça
3) C_k = *Novos_Candidatos* (F_{k-1})
4) Para todas as transações $t \in D$ faça
5) C_t = subconjunto (C_k , t); // Candidatos contidos em t
6) Para todos os candidatos $c \in C_t$
7) *c.suporte*= *c.suporte* +1
8) $F_K = \{c \in C_k \mid c.suporte \geq suporte\_mínimo\}$
9) $F = F \cup F_k$

Figura 3.4 – Algoritmo *Apriori*

A rotina *Novos_Candidatos* do algoritmo *Apriori* gera os candidatos utilizando somente os conjuntos de itens freqüentes encontrados no passo anterior, sem considerar as transações no banco de dados, como faziam o AIS [AGR93.1] e o SETM [HOU93], pois estes algoritmos geram os candidatos durante a leitura dos dados. A percepção básica do *Apriori* é que qualquer subconjunto de um conjunto de itens freqüentes deve ser freqüente. Portanto, o conjunto de candidatos contento *k* itens pode ser gerado fazendo uma combinação dos conjuntos de itens freqüentes de tamanho *k-1*, e anulando aqueles que contenham algum subconjunto que não seja freqüente [AGR94].

Na Figura 3.5, pode-se observar o algoritmo utilizado pelo *Apriori* para a geração do conjunto de itens candidatos. No passo 1, é realizada a combinação dos conjuntos em F_{k-1} para a geração de C_k.. No passo 2, ocorre a retirada de C_k dos candidatos que possuem algum subconjunto de tamanho *k-1* que não seja freqüente, ou seja, não pertençam à lista F_{k-1}. No algoritmo *Apriori*, assume-se que os itens de cada transação do banco de dados estão em ordem lexicográfica, logo *p*.item1, *p*.item2, ..., *p*.item$_k$ e *q*.item1, *q*.item2, ..., *q*.item$_k$ estão assim ordenados.

1) $C_k \leftarrow$ select p.item1,p.item2, …, p.item$_{k-1}$, q.item$_{k-1}$
from F_{k-1} p, F_{k-1} q
where p.item$_1$=q.item$_1$, ..., p.item$_{k-2}$=q.item$_{k-2}$, p.item$_{k-1}$<q.item$_{k-1}$

2) Para todos os conjuntos de itens $c \in C_k$ faça
Para todos os subconjuntos s de tamanho *k-1* de *c* faça
Se $(s \notin L_{k-1})$ então
Retire *c* de C_k

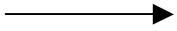
Figura 3.5 – Algoritmo para a geração de candidatos do *Apriori*

Considerando o banco de dados *D*, os passos do *Apriori* serão:

Passo k = 1:

Neste passo, todos os atributos são considerados candidatos e têm os seus suportes apurados para a geração da lista F_1 de itens freqüentes.

C_1	Suporte
{A}	5
{B}	3
{C}	2
{D}	5
{E}	3
{F}	1

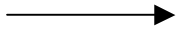


F_1	Suporte
{A}	5
{B}	3
{D}	5
{E}	3

Passo k = 2:

Neste passo, utiliza-se a lista de itens freqüentes F_1 para geração dos itens candidatos, através das combinações dos itens em F_1.

C_2	Suporte
{A B}	3
{A D}	4
{A E}	3
{B D}	2
{B E}	1
{D E}	3



F_2	Suporte
{A B}	3
{A D}	4
{A E}	3
{D E}	3

É observado que o item {C}, assim como o item {F}, não pertence à F_1. Portanto, não pertencerão à C_2, pois apenas as combinações entre os itens de F_1 são registradas.

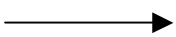
Passo $k = 3$:

Da mesma forma que no passo anterior, combinam-se os itens pertencentes à F_2 para a geração de C_3.

O único item a ser inserido em C_3 é o {A D E}, originário da combinação de {A D} e {A E}.

Da combinação de {A B} com {A D}, por exemplo, resultaria o item {A B D}. Embora {A B} e {A D} pertençam à F_1, a combinação {B D} não pertence, não sendo assim, um item freqüente. Logo, {A B D} também não pode ser um item freqüente por conter o subconjunto {B D}. Da mesma forma, da combinação de {A B} com {A E}, por exemplo, resultaria o item {A B E}, e como o subconjunto {B E} não é freqüente, {A B E} pode ser descartado.

C_3	Suporte
{A D E}	3



F3	Suporte
{A D E}	3

Passo $k = 4$:

O algoritmo *Apriori* irá parar neste passo já que a lista F_3 só possui um único elemento, não podendo mais haver combinação de itens. Logo $F = F_2 \; U \; F_3$ [Tabela 3-1].

F	Suporte
{A B}	3
{A D}	4
{A E}	3
{D E}	4
{A D E}	3

Tabela 3-1 - $F = F_2 \; U \; F_3$.

A mineração de regras de associação é uma tarefa freqüentemente difícil no mínimo por duas razões: primeiro, existem muitos dados e segundo, os dados possuem muitos atributos.

O tempo de execução do *Apriori* é linear em relação ao tamanho da amostra, logo, o número de passagens pela base de dados é um fator que reflete neste tempo. No *Apriori*, são realizadas *n* passagens pela base de dados, onde *n* é a quantidade de itens do maior conjunto de candidatos. Cada passagem pelo banco de dados significa uma rotina de E/S.

O número de atributos do banco de dados é um fator de aumento da complexidade do *Apriori*. O pior caso ocorre quando todos os subconjuntos do conjunto de atributos são freqüentes. Seja $|T|$ a quantidade de atributos da tabela *T*, que representa o banco de dados *D*. Neste caso, a complexidade é $O(2^{|T|})$. Entretanto, assumindo que *T* seja uma tabela esparsa (a maioria dos valores dos atributos é 0, vide Figura 3.1), então é esperado que o conjunto de itens freqüentes tenha tamanho máximo de *k*, onde $k << |T|$. Se esta expectativa for satisfeita, no pior caso a complexidade será [SIE03]:

$$O(\sum_{j=1}^{k} \binom{|T|}{j}) = O(|T|^k) << O(2^{|T|}).$$

Se considerarmos uma tabela densa, onde existam muitos conjuntos freqüentes longos (com muitos itens), a performance do *Apriori* é sensivelmente degradada. Isto

ocorre porque são realizadas tantas passadas pela base de dados quanto é o tamanho do mais longo conjunto de itens freqüentes. Além disso, verificar um conjunto grande de candidatos é computacionalmente caro, principalmente para conjuntos longos. Quando a quantidade de itens dos conjuntos freqüentes é grande, o *Apriori* é degradado por causa do custo computacional e não pelo excesso de E/S.

Quanto menor for o suporte mínimo especificado, maiores são as chances de serem considerados freqüentes os conjuntos candidatos. Esse parâmetro é portanto também um fator para a complexidade do *Apriori* que pode se refletir tanto no custo de E/S quanto no custo de CPU. Pode aumentar a E/S, já que com um suporte pequeno conjuntos com quantidades maiores de itens poderão vir a se tornar freqüentes e exigir mais passadas pelo banco. Quanto ao custo de CPU, maior número de candidatos deverão ser testados.

Diversos algoritmos foram propostos para amenizar as situações propostas anteriormente, dentre eles, pode-se citar:

- redução do número de passagens pelo banco de dados: *AprioriTid* [AGR94], *DHP* [PAR97], *DIC* [BRIN97], *Partition* [SAV95], *FPgrowth* [HANPEI00];
- alternativas ao suporte muito baixo: *MsApriori* [LIU99];
- utilização de amostragens [TOI96, DOM98];
- redução de candidatos *DHP* [PAR97], [CAM00];
- alternativas para base de dados densas: *MaxMiner* [BAY98], *A-clo*se [PAS99], *Closet* [HAN00], *Charm* [ZAK02].

3.4.3 GERAÇÃO DE REGRAS

A geração de regras efetua-se a partir do conjunto de itens freqüentes. Nesta etapa, é utilizado o parâmetro de confiança mínima, que mede a força com que um grupo de itens depende de outro grupo. Na geração das regras, a base de dados não necessita ser percorrida, pois o cálculo da confiança da regra pode ser feito somente com o suporte do antecedente e do conseqüente da regra sendo analisada. Na fase de geração dos itens freqüentes, o suporte de todos os itens já é calculado.

Para a geração de regras a partir dos conjuntos de itens freqüentes selecionados pode ser utilizado o procedimento da Figura 3.6. Para cada um dos conjuntos de itens freqüentes, são extraídos os seus subconjuntos não nulos e a confiança de cada regra é calculada. A regra será selecionada se a confiança da mesma for maior ou igual à confiança mínima previamente estabelecida.

Para cada conjunto de itens freqüentes *f* de *F* faça

Para cada subconjunto não nulo *s* de *f* faça

$confiança(f) = (\sup orte(f) / \sup orte(f - s))$

Se $confiança(f) \geq confiança\_mínima)$

A regra $(f - s) \rightarrow s$ é válida com confiança *confiança(f)* e *suporte(f)*

Figura 3.6 – Algoritmo para a geração de regras

A complexidade da geração de regras é também exponencial, pois para cada *f* são considerados $2^{|f|} - 1$ subconjuntos não nulos de *f*. Entretanto, como é considerado que $|f| \leq k << |T|$, este não se torna um problema. Em alguns casos, somente as regras com apenas um conseqüente são geradas, o que torna a complexidade do algoritmo de geração de regras linear [SIE03].

Considere a lista *F* de itens freqüentes gerada pelo *Apriori*. A partir de *F*, o algoritmo da Figura 3.6 gera os subconjuntos para cada item *f* de *F* [Tabela 3-2]:

F	Subconjuntos
{A B}	{A}, {B}
{A D}	{A}, {D}
{A E}	{A}, {E}
{D E}	{D}, {E}
{A D E}	{A}, {D}, {E}, {A D}, {A E}, {D E}

Tabela 3-2 - Subconjuntos para cada item *f* de *F*

Para cada subconjunto de *f*, a confiança das regras geradas por este subconjunto será calculada. Considerando uma confiança mínima de 80%, teremos os seguintes resultados:

Para o item {A B}:

Subconjunto B: Suporte ({A B}) / Suporte ({A}) = 3/5 = 60%

Subconjunto A: Suporte ({A B}) / Suporte ({B}) = 3/3 = 100%

Para {A D}:

Subconjunto D: Suporte ({A D}) / Suporte ({A}) = 4/5 = 80%

Subconjunto A: Suporte ({A D}) / Suporte ({D}) = 4/5 = 80%

Para {A E}:

Subconjunto E: Suporte ({A E}) / Suporte ({A}) = 3/5 = 60%

Subconjunto A: Suporte ({A E}) / Suporte ({E}) = 3/3 = 100%

Para {D E}:

Subconjunto E: Suporte ({D E}) / Suporte ({D}) = 3/5 = 60%

Subconjunto D: Suporte ({D E}) / Suporte ({E}) = 3/3 = 100%

Para {A D E}:

Subconjunto D, E: Suporte ({A D E}) / Suporte ({A}) = 3/5 = 60%

Subconjunto A, E: Suporte ({A D E}) / Suporte ({D}) = 3/5 = 60%

Subconjunto A, D: Suporte ({A D E}) / Suporte ({E}) = 3/3 = 100%

Subconjunto E: Suporte ({A D E}) / Suporte ({A D}) = 3/4 = 75%

Subconjunto D: Suporte ({A D E}) / Suporte ({A E}) = 3/3 = 100%

Subconjunto A: Suporte ({A D E}) / Suporte ({D E}) = 3/3 = 100%

As regras selecionadas serão:

B $\rightarrow$ A, com confiança =100% e suporte = 3

A $\rightarrow$ D, com confiança =80% e suporte = 4

D $\rightarrow$ A, com confiança =80% e suporte = 4

E $\rightarrow$ A, com confiança = 100% e suporte = 3

E $\rightarrow$ D, com confiança =100% e suporte = 3

E $\rightarrow$ A D, com confiança = 100% e suporte = 3

A E $\rightarrow$ D, com confiança =100% e suporte = 3

D E $\rightarrow$ A, com confiança =100% e suporte = 3

O algoritmo da Figura 3.6 pode ser aperfeiçoado utilizando-se a propriedade apresentada em [AGR94]: se a regra $a \rightarrow (f - a)$ não é válida, não é necessário verificar a regra $a' \rightarrow (f - a')$, onde a' é um subconjunto de *a*, pois esta também não será válida. Isto ocorre porque qualquer subconjunto a' de *a* possui suporte maior ou igual ao suporte de *a*. Logo, a confiança da regra $a' \rightarrow (f - a')$ não poderá ser maior que a confiança de $a \rightarrow (f - a)$. Por isto, se *a* não gera uma regra envolvendo todos os itens de *f* com *a* como antecedente, a' também não irá gerar.

No exemplo, se (A D $\rightarrow$ E) não é uma regra válida, (A $\rightarrow$ D E) e (D $\rightarrow$ A E) não precisam ser verificadas, pois também não terão validade. Isto é confirmado, pois a regra (A D $\rightarrow$ E) possui confiança de 75% e as regras (A $\rightarrow$ D E) e (D $\rightarrow$ A E) possuem confiança de 60%.

A propriedade acima descrita pode ser reescrita. Se a regra $(f - b) \rightarrow b$ é válida, todas as regras da forma $(f - b') \rightarrow b'$ também serão, onde b' é um subconjunto não vazio de *b*. Do exemplo, tiramos que se a regra (E $\rightarrow$ A D) tem confiança igual a 100%, então as regras (D E $\rightarrow$ A) e (A E $\rightarrow$ D) também deverão ser válidas, o que realmente é verificado, pois ambas também possuem confiança 100%.

3.5 REDUÇÃO DE E/S

Como visto na seção 3.4.2, o tempo de execução de um algoritmo de extração de regras de associação é influenciado pelo número de passagens realizadas na base de

dados. Se a base de dados não está na memória, serão realizadas *n* operações de E/S, onde *n* é o tamanho do maior conjunto de itens freqüentes. Alguns métodos foram propostos para minimizar a E/S dos algoritmos e priorizar as operações realizadas em memória.

O algoritmo *ApririTID* [AGR94] utiliza o mesmo algoritmo básico do *Apriori*. Entretanto, depois da primeira passagem pelo banco de dados, ele guarda para cada transação os conjuntos de itens freqüentes encontrados na mesma. Contudo, a base de dados reescrita pode não caber na memória. Uma solução seria usar o *AprioriHybrib* [AGR94] que utiliza o *Apriori* inicialmente e, quando possível, utiliza o *AprioriTID*.

Outra abordagem para a redução do número de passagens pelo banco de dados é proposta pelo algoritmo *DIC* (*Dynamic Itemset Couting*) [BRIN97]. O *DIC* é uma generalização do *Apriori*, onde o algoritmo faz paradas a cada *m* transações. Em cada parada, conjuntos freqüentes maiores são introduzidos. Há redução do número de passagens, pois o algoritmo passa a depender da variável *m*, onde $m \leq tamanho(D)$. Entretanto, o algoritmo depende da homogeneidade da base de dados. Uma solução possível para este problema seria aleatorizar a ordem das transações. A Figura 3.7 exibe uma amostra do que seria o número de passagens realizadas pelo *Apriori*, e a Figura 3.8 descreve as passagens do *DIC* [JOR03].

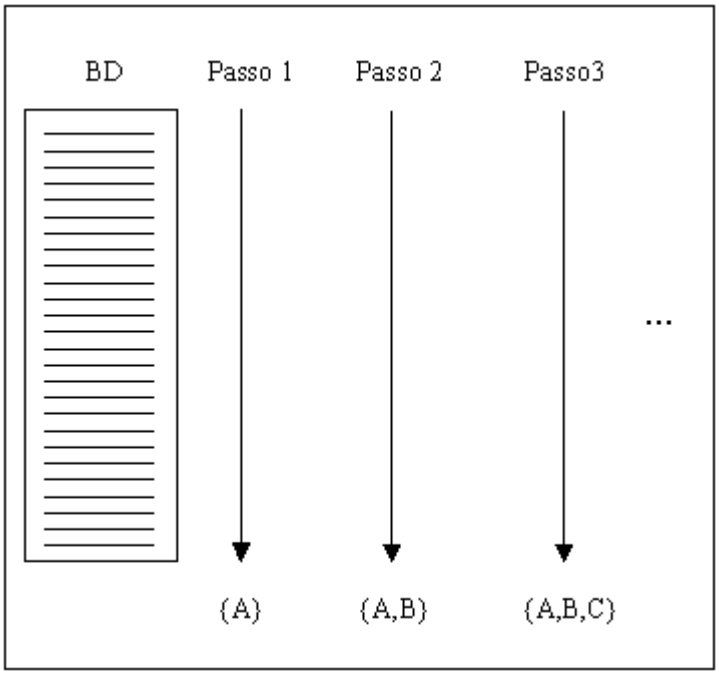


Figura 3.7 – Passagens realizadas pelo *Apriori* através da base de dados

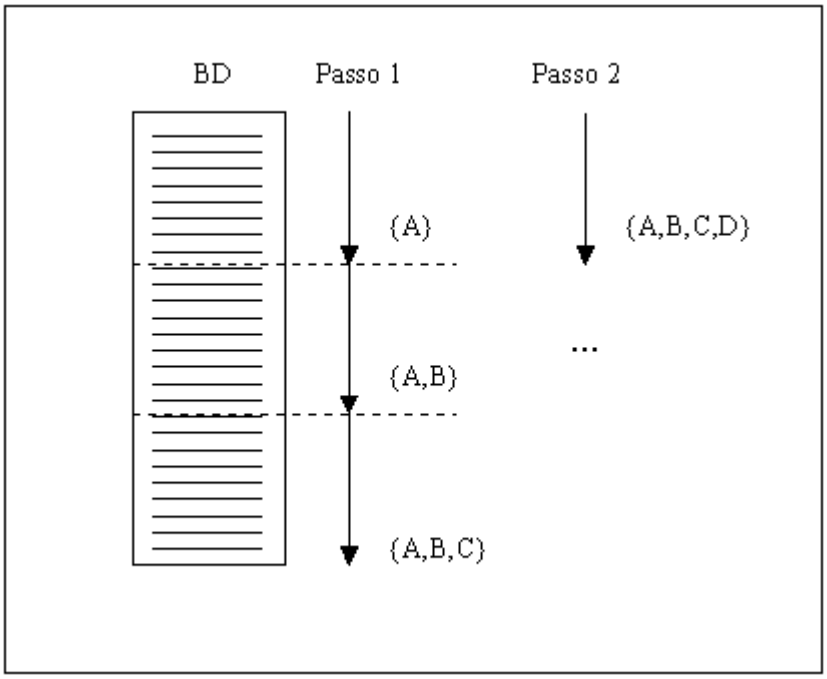


Figura 3.8 –Passagens realizadas pelo *DIC* através da base de dados

O algoritmo *Partition* [SAV95] também propõe um método para reduzir o número de passagens pelo banco de dados. O conceito do algoritmo baseia-se na divisão da base de dados em partições que caibam na memória. Para cada partição, é identificado o conjunto de itens candidatos, que são aqueles freqüentes na partição. Ao fim da primeira passagem, um superconjunto dos candidatos é obtido. Na segunda passagem, o suporte dos conjuntos de itens candidatos é calculado. O algoritmo *Partition* garante, no máximo, duas passagens pela base. Podem ser gerados, nesta abordagem, candidatos que não seriam gerados utilizando toda a base de dados. Isto pode ocorrer quando uma partição é uma amostra tendenciosa ou quando a partição é muito pequena. Da mesma forma que no *DIC*, a homogeneidade dos dados é importante [SIE03].

A amostragem também pode ser utilizada para a extração de conjunto de itens freqüentes. O tempo de execução do algoritmo *Apriori* é linear em relação ao tamanho da base de dados, logo, trabalhar sobre uma amostra da base de dados diminuirá o tempo de execução, o que reduz, também, o número de acessos de E/S [JOR03, TOI96]. O problema da utilização da amostragem é que podem ser gerados alguns resultados

incorretos: podem ser encontradas regras que não sejam válidas na base de dados inteira e podem não ser encontradas regras que sejam válidas para a base completa. A probabilidade de tais erros ocorrerem depende do tamanho da amostra utilizada, do suporte, etc. Em [TOI96, SIE03], estudos sobre erros são exibidos. Uma possibilidade para a utilização da amostragem é primeiro usar a amostra para extração dos resultados e, depois, verificá-los diante de toda a base de dados. Desta forma, a base de dados só será percorrida para a geração da amostra e uma única outra vez para teste dos conjuntos de itens freqüentes.

3.6 ATUALIZAÇÃO INCREMENTAL

Quando a base de dados é dinâmica, em termos de um constante fluxo de novos dados, uma associação de itens pode vir a se tornar fraca ou forte na medida em que a base de dados evolui. Uma possível solução para manter as regras de associação atualizadas é aplicar repetidamente um algoritmo tradicional de regras de associação assim que novas transações são inseridas, excluídas ou alteradas no banco de dados. O algoritmo é reexecutado para a geração de regras de associação mais novas. Entretanto, este processo não será eficiente se a taxa de atualização da base de dados for muito alta, pois o custo computacional da reaplicação muito freqüente dos algoritmos pode ser excessivamente alto.

Como o custo de extração freqüente de regras de associação em bases de dados dinâmicas muito grandes é alto, técnicas de atualização incremental das regras podem ser desenvolvidas para prevenir que se refaça a mineração de dados em toda nova base de dados.

Técnicas de atualização incremental de regras de associação utilizam o conhecimento anteriormente descoberto para manutenção dos novos conjuntos de itens freqüentes com objetivo de aumentar a performance dos algoritmos.

Quando uma base de dados sofre atualizações, as regras descobertas anteriormente podem não ser válidas na nova base de dados ou pode ocorrer o surgimento de outras regras que não eram válidas anteriormente. Aqueles conjuntos de itens freqüentes que continuam sendo válidos na base atualizada são chamados de *retained itemsets*. Os conjuntos de itens que se tornaram freqüentes após a atualização

da base de dados são chamados de *emerged itemsets* [VEL01]. Os algoritmos para manutenção incremental de regras de associações provêm mecanismos para geração dos *retained itemsets* e dos *emerged itemsets*, além de efetuar o descarte dos conjuntos freqüentes que se tornaram associações fracas.

Diversos algoritmos são encontrados para a atualização incremental de regras de associação, alguns deles são encontrados em [AYA99, CHE97, LEE97, VEL01, VEL02, CHE96 e THO00].

3.7 PARALELISMO

Apesar do aumento do poder computacional disponível, os algoritmos para determinação de regras de associação ainda são capazes de esgotar os recursos de computadores atuais. Uma forma de minimizar o tempo de resposta destes algoritmos é a sua paralelização.

O grande tamanho das bases de dados disponíveis e as suas características de multi-dimensionalidade (grande número de atributos) fazem com que a tarefa de encontrar todas as regras de associação necessite de muito poder computacional e de vastos recursos de memória, podendo resultar em tempos de execução muito altos.

O esforço computacional na descoberta de regras de associação consiste na geração de candidatos e na contagem de suas ocorrências na base de dados. As necessidades de memória vêm da armazenagem dos candidatos gerados. No *Apriori*, por exemplo, se o número de candidatos for muito grande, eles podem não caber na memória principal. A cada iteração, seriam necessárias várias passagens pela base de dados, causando um custo de E/S ainda maior que o usual.

Estas características vêm motivando o desenvolvimento de algoritmos paralelos e faz da descoberta de regras de associação um problema ideal para ser solucionado com múltiplos processadores em paralelo.

Outra razão para utilização de algoritmos paralelos é a existência de bases de dados distribuídas em diversos servidores de dados. O custo de trazê-los para um mesmo computador para a descoberta através de algoritmos seriais pode ser muito caro.

Para fazer uso da concorrência, o trabalho computacional e as requisições de memória precisam ser distribuídos entre todos os processadores disponíveis. Maiores detalhes podem ser encontrados em [MES02, MAH00, ZAK99]. Os algoritmos paralelos propostos foram desenvolvidos para efetivamente paralelizar um ou ambos problemas: geração de candidatos e contagem de candidatos. A fase de contagem dos candidatos pode ser paralelizada de maneira relativamente simples através da distribuição da base de dados e da coleta de contagens locais para o conjunto total de candidatos armazenado em todos os processadores. O algoritmo *CD* [AGR96.2] é um exemplo. Entretanto, a geração e o armazenamento de grande número de candidatos continua sendo um gargalo para esta proposta. Alguns algoritmos paralelos foram propostos para solucionar estes problemas (*IDD* em [HAN97]), onde o aspecto central é a distribuição dos conjuntos de candidatos entre os processadores para extrair cooperação na geração dos candidatos, assim como da fase de contagem. Outros algoritmos, como o *HD* e *PDM*, e outros estudos podem ser encontrados em [HAN97, PAR95, AGR96.2, ZAK97.2, TAL02, ZAK00 e CHE96.2].

3.8 MANIPULAÇÃO DO EXCESSO DE REGRAS

Na prática, as regras de associação estão sujeitas a um efeito desagradável que é a geração de regras em abundância. O número de regras descobertas é inversamente proporcional ao suporte mínimo e à confiança mínima. Se estes parâmetros são fixados com valores altos podem ser descobertas apenas regras já conhecidas. Se os parâmetros são fixados com valores baixos, o número de regras geradas cresce enormemente.

Existem duas direções para gerenciar o excesso de resultados: o pré-processamento e o pós-processamento das regras. O pré-processamento significa que devem ser geradas menos regras. O pós-processamento destina-se a filtrar ou ordenar as regras de forma a serem selecionadas apenas aquelas mais interessantes [SIE03].

3.8.1 PÓS-PROCESSAMENTO DOS RESULTADOS

O objetivo do pós-processamento é prover um mecanismo para que as regras geradas pelos algoritmos possam ser manipuláveis pelos usuários finais. Se o número de regras é muito grande, pode ocorrer que o usuário final não tenha como lidar com todo o

volume de informações. Basicamente, existem duas maneiras de efetuar o pós-processamento. A primeira é o mecanismo de ordenação das regras segundo algum critério, o que vai gerar uma organização na exibição das mesmas. A segunda é incorporar algumas medidas para a seleção de apenas determinadas regras mais importantes, uma vez que é interessante exibir para o usuário não uma quantidade grande de padrões, mas aqueles que são, de fato, interessantes [BRU00].

3.8.1.1 ORDENAÇÃO DOS RESULTADOS

Se muitas regras são geradas, uma alternativa é ordenar os resultados a serem exibidos. As regras podem ser ordenadas pelos conseqüentes, por cada conseqüente podem ser ordenados pelo tamanho do antecedente (quantidade de itens no antecedente). O suporte e a confiança também podem ser utilizados para definir uma ordem das regras. Ordenam-se as regras do suporte mais elevado para o mais baixo, para cada suporte ordenam-se as regras através da confiança, da maior para a menor [SIE03].

Embora estas formas de exibição ofereçam alguma ordem para o grau elevado de regras, elas não solucionam o problema dos resultados não interessantes.

3.8.1.2 REGRAS INTERESSANTES

Uma regra é dita interessante se ela proporciona alguma informação útil. Além do suporte e da confiança [Seção 3.2], existem diversas medidas para avaliar o interesse de uma regra: sustentação (*lift*), interesse (*interest*), convicção (*conviction*), força coletiva (*collective strength*), ganho (*gain*), entropia (*entropy*), X^2, *coverage*, *leverage*, etc. [SIE03, JOR03, BAY99, WIZ03, MAG03]. Tais medidas podem ser utilizadas para a decisão de quais regras devem ser mantidas e quais devem ser descartadas. Ao usuário, são exibidas apenas aquelas mais interessantes. Desta forma, reduz-se o tempo de análise das regras pelo usuário final.

O *lift* de uma regra de associação revela de que maneira uma regra prediz o conseqüente melhor que uma predição aleatória:

$$lift(X \rightarrow Y) = \frac{suporte(XY)/suporte(Y)}{suporte(Y)/|D|}$$

$$= confiança(X \rightarrow Y) \times \frac{|D|}{suporte(Y)},$$

onde $|D|$ é o tamanho da base de dados, como $|D|$ refere-se à 100% da base de dados, consideramos que $|D|=1$.

A medida interesse (*interest*) de uma regra mede a dependência entre *X* e *Y*. É o quociente entre a probabilidade conjunta observada e a probabilidade conjunta sob independência. Na verdade, a medida interesse é igual ao *lift*:

$$\frac{P(X,Y)}{P(X) \times P(Y)} = \frac{suporte(XY)}{suporte(X) \times suporte(Y)}$$

O interesse ou *lift* mínimo geralmente é igual a 1. Se a regra possui *lift* menor do que 1, deve ser descartada. Tomemos como exemplo o seguinte caso: 80% das pessoas compram leite, 2% das pessoas compram salmão e 1,5% das pessoas compram leite e salmão. O *lift* da regra $salmão \rightarrow leite$ será $\frac{1{,}5\%}{80\% \times 2\%} = 0{,}9375$. Esta regra pode ser descartada pois o *lift* é menor do que 1. Na verdade, leite não é dependente de salmão. O que ocorre é que geralmente as pessoas que levam salmão também levam leite, mas somente porque elas levam o leite na maioria das vezes, comprando ou não o salmão.

O interesse não mede se X causa Y. Para uma percepção deste tipo, utiliza-se a convicção (*conviction*) das regras que é dada por:

$$\frac{P(X) \times P(\neg Y)}{P(X,\neg Y)} = \frac{suporte(X) \times (|D| - suporte(Y))}{suprte(X) - suporte(XY)} = \frac{|D| - suporte(Y)}{|D|(1 - confiança(X \rightarrow Y)},$$

ou

$$\frac{P(X) \times P(\neg Y)}{P(X,\neg Y)} = \frac{1}{interesse(X \rightarrow \neg Y)}.$$

Quanto mais alta a convicção, mais freqüentemente *Y* ocorre junto de *X*. Quando for igual a 1, indica independência. Se possuir um valor muito alto, pode identificar regras pouco interessantes.

Vejamos outro exemplo. Se 1% da população é militar, se 50% das pessoas são adultas e se todos os militares devem ser adultos obrigatoriamente, então a regra $militar \rightarrow adulto$ possui

$$interesse = \frac{1\%}{1\% \times 50\%} = 2$$

e

$$convicção = \frac{1}{interesse(militar \rightarrow \neg adulto)} = \frac{1}{0} = \infty .$$

A convicção é sensível à direção, já o interesse não é. A convicção de $X \rightarrow Y$ não é igual à convicção de $Y \rightarrow X$, mas o interesse de $X \rightarrow Y$ é igual ao interesse de $Y \rightarrow X$.

Após a geração de todas as regras de associação, utilizando as medidas de avaliação de interesse das regras, aquelas regras que não atingirem determinados limiares de interesses poderão ser eliminadas.

Além das medidas acima, consideradas medidas objetivas de interesse porque dependem apenas dos dados e parâmetros utilizados, existem as medidas subjetivas de interesse. As medidas subjetivas não podem ser utilizadas para filtrar automaticamente as regras interessantes após o processamento das mesmas, pois essas medidas dependem diretamente da avaliação do usuário final da aplicação de mineração de dados. Como são medidas subjetivas, uma regra pode ser interessante para um usuário e não interessante para outro. Do ponto de vista do usuário, uma regra é interessante se ela possui um grau elevado de utilidade ou de inesperabilidade. Em relação à utilidade, um padrão é interessante se o usuário pode construir algo, tomar decisões a partir dele. A inesperabilidade, por sua vez, deve auxiliar na descoberta de regras surpreendentes, deve ser capaz de contradizer as expectativas do usuário [BRU00].

Vale a pena ressaltar que existem inúmeros outros estudos sobre o assunto de triagem de regras de associação, como exemplo pode-se citar a dissertação de mestrado a seguir referenciada: [SIL00].

3.8.2 PRÉ-PROCESSAMENTO DAS REGRAS

O gerenciamento do excesso de regras pode ser abordado pela perspectiva do pré-processamento. O objetivo é a geração de uma quantidade menor de regras, exatamente as mais interessantes.

Existem duas propostas para este problema [SIE03]:

- Dentro dos limites da estrutura de suporte e de confiança, são utilizadas representações condensadas (*condensed representations*). Exemplos destas representações são os *Maximal Frequent Itemsets* e *Closed Frequent Itemsets*.
- Fora dos limites da estrutura de suporte e confiança é utilizada a independência condicional (*condicional independence*). Como exemplo, citamos o MAMBO [VAL01]. A idéia central destas abordagens é que mesmo regras com suportes muito baixos podem ser interessantes, mas o custo de mineração com suporte baixo é alto. Logo, não se utiliza o suporte, mas a independência condicional para selecionar os conjuntos de itens [SIE03].

3.8.2.1 *MAXIMAL FREQUENT ITEMSETS*

Maximal frequent itemsets são conjuntos de itens freqüentes tal que nenhum de seus superconjuntos é freqüente. Desta forma, é fácil observar que cada conjunto de itens freqüentes é um subconjunto de um *Maximal frequent itemset*. Por isso, o conjunto de todos os *Maximal frequent itemsets* é uma representação condensada de todos os conjuntos de itens freqüentes.

Métodos para geração dos *Maximal frequent itemsets* são utilizados em bases de dados muito densas, onde encontrar todos os conjuntos possíveis de itens freqüentes se torna uma tarefa muito difícil. São especialmente interessantes quando existem muitos conjuntos de itens freqüentes extensos.

É importante ressaltar que encontrar somente os *Maximal frequent itemsets* significa que não poderão ser gerados a partir destes todos os conjuntos de itens freqüentes.

São exemplos destes métodos os algoritmos *MAXMINER* [BAY98], *All-MFS* [GUN97], *Pincer-search* [LIN98], *MAFIA* [BUR01], *DepthProject* [AGR00] e *GenMax* [GOU01].

3.8.2.2 *CLOSED FREQUENT ITEMSETS*

Closed frequent itemsets são conjuntos de itens freqüentes que caracterizam completamente seu conjunto de transações associadas. Isto é, um *itemset X* é *closed* se *X* contém todos os itens que ocorrem em todas as transações do suporte de *X*.

Considere um conjunto de itens freqüentes I e $\sigma(I)$ o conjunto de todas as transações em que todos os itens de *I* estão presentes ($\sigma(I)$ é o conjunto das transações que suportam *I*). Um conjunto de itens *I* é *closed* se, para todos os seus superconjuntos J, $\sigma(I)$ é um superconjunto de $\sigma(J)$ [SIE03].

Closed frequent itemsets possuem algumas propriedades importantes [SIE03]:

- O suporte de um conjunto de itens é igual ao suporte do menor c*losed itemset* que o contém.
- Um *maximal frequent itemset* é *closed*.
- O conjunto de todos os *closed frequent itemsets*, com o seus suportes, é um conjunto gerador para todos os conjuntos de itens freqüentes e seus suportes. Também podem ser derivados os suportes e as confianças de todas as regras de associação válidas na base de dados. Como é esperado que existam *closed frequent itemsets* em um número bem menor que o de conjuntos de itens freqüentes, espera-se, também, que até mesmo bases de dados muito densas possam ser mineradas.

Citamos alguns exemplos de algoritmos para geração de *closed frequent itemsets*: *CHARM* [ZAK02], *A–Close* [PAS99] e *Closet* [HAN00].

3.8.2.3 ITENS FREQÜENTES, *MAXIMAL* E *CLOSED*

Como exemplo (extraído do artigo [ZAK01]), considere o banco de dados da Figura 3.9. Existem cinco diferentes itens, *I*={A,B,C,D,E} e seis transações

T={1,2,3,4,5,6}. A tabela da direita exibe todos os 19 conjuntos de itens freqüentes que contêm pelo menos três transações, ou seja, o suporte mínimo é de 50%. Um conjunto de itens freqüentes é considerado *maximal* se ele não é subconjunto de nenhum outro conjunto de itens freqüentes. Um conjunto de itens freqüentes *X* é chamado de *closed* se não existe nenhum superconjunto $Y \supset X$ tal que $\sigma(X) = \sigma(Y)$.

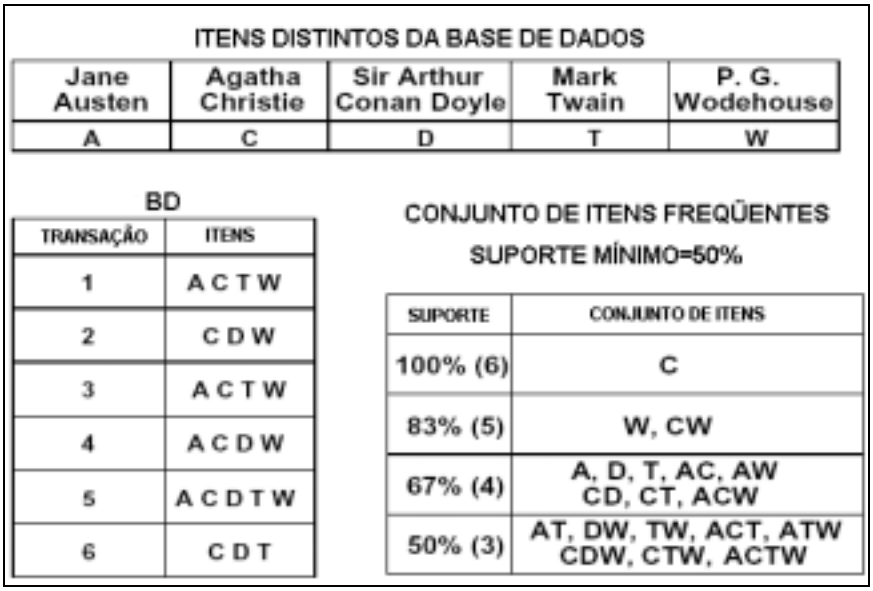


Figura 3.9 – Mineração de conjunto de Itens Freqüentes

Na Figura 3.10, são mostrados todos os 19 conjuntos de itens freqüentes juntamente com os identificadores das transações em que eles aparecem. Os 7 conjuntos *closed* são obtidos pelo grupamento de todos os conjuntos de itens freqüentes que possuem o mesmo conjunto de identificadores de transações (encontram-se circulados no esquema do lado esquerdo da figura). Na parte direita da figura, os *maximal itemsets* estão marcados com círculos: *ACTW* e *CDW*. Como o exemplo sugere, em geral, se *F* corresponde aos conjuntos de itens freqüentes, *C* aos *closed itemsets* e *M* aos *maximal itemsets*, temos $M \subset C \subset F$. Enquanto em *C* não existe perda de informação, no sentido em que a freqüência exata de todos os conjuntos de itens freqüentes pode ser determinada a partir de *C*, em *M* existe perda de informação. Para verificar se um conjunto *X* é freqüente, encontra-se o menor conjunto *closed* que seja um superconjunto de *X*. Se não existe tal superconjunto, então *X* não é freqüente. Por exemplo, *ATW* é

freqüente e possui a mesma freqüência do conjunto *closed ACTW*, enquanto *DT* não é freqüente, pois não existe nenhum conjunto *closed* que o contém.

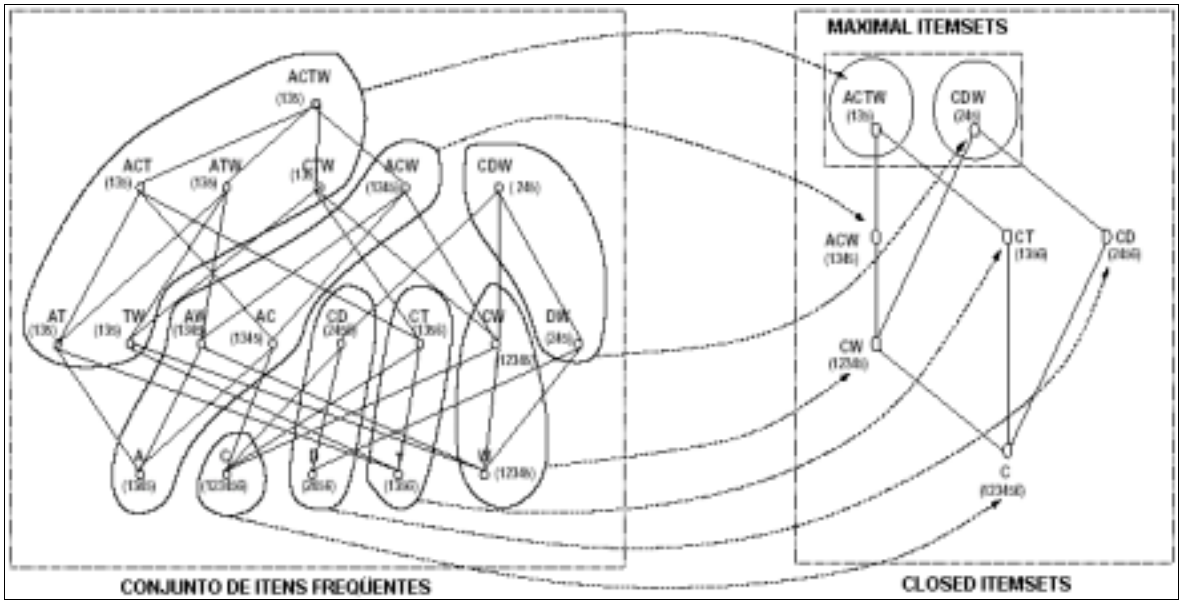


Figura 3.10 - Conjunto de Itens Freqüentes, *Maximal Itemsets*, *Closed Itemsets*

3.9 ATRIBUTOS NÃO BINÁRIOS

Quando os atributos são categóricos, ao invés de binários, a mineração de regras de associação pode ser efetuada da mesma forma. As regras que utilizam atributos categóricos são da seguinte forma:

$X_1 = x_1 \wedge ... X_n = x_n \rightarrow Y_1 = y_1 \wedge ... Y_m = y_m$, onde $X_1,...,X_n$ e $Y_1,...,Y_m$ são atributos categóricos e $x_1,...,x_n$ e $y_1,...,y_m$ são os valores respectivos dos atributos.

O suporte da expressão anterior é definido por:

$s(X_1 = x_1 \wedge ... X_n = x_n \wedge Y_1 = y_1 \wedge ... Y_m = y_m) =$

$|\{t \in D \,|\, \pi_{X_1 ... X_n, Y_1 ... Y_m} = (x_1,..., x_n, y_1,..., y_m)\}| / |D|$, onde t é uma transação do banco de dados D.

A confiança da regra é dada por:

$$\frac{s(X_1 = x_1 \wedge ... X_n = x_n \wedge Y_1 = y_1 \wedge ... Y_m = y_m)}{s(X_1 = x_1 \wedge ... X_n = x_n)}.$$

As fórmulas acima são equivalentes às formulas de suporte e de confiança da seção 3.2.

Atributos categóricos podem ser transformados em atributos binários, onde o atributo de cardinalidade *k* é substituído por *k* atributos binários. O valor binário 1 será posicionado no atributo binário gerado equivalente ao valor do atributo categórico, os outros atributos binários serão gerados com o valor 0.

Considere a tabela *T* abaixo [Figura 3.11] composta pelos atributos *SEXO*, *ESTADO_CIVIL* e *UF*. Os atributos possuem os seguintes domínios:

- *SEXO (S)*: F (feminino) e M (masculino).
- *ESTADO_CIVIL (EC)*: C (casado), S (solteiro), D (divorciado) e V (viúvo).
- *UF*: RJ, SP, MG.

t_{ID}	SEXO	EC	UF
1	F	C	RJ
2	M	S	SP
3	M	S	RJ
4	M	D	RJ
5	F	S	RJ
6	F	C	MG
7	M	V	SP
8	M	S	RJ
9	M	S	RJ

Figura 3.11 – Tabela *T* com atributos categóricos

A tabela *T'* transformada para atributos binários é exibida na figura 3.12.

t_{ID}	SEXO=F	SEXO=M	EC=C	EC=S	EC=D	EC=V	UF=RJ	UF=SP	UF=MG
1	1	0	1	0	0	0	1	0	0
2	0	1	0	1	0	0	0	1	0
3	0	1	0	1	0	0	1	0	0
4	0	1	0	0	1	0	1	0	0
5	1	0	0	1	0	0	1	0	0
6	1	0	1	0	0	0	0	0	1
7	0	1	0	0	0	1	0	1	0
8	0	1	0	1	0	0	1	0	0
9	0	1	0	1	0	0	1	0	0

Figura 3.12 – Tabela *T'* com atributos binários

Entretanto, se o domínio de um atributo for muito grande, tratando-se cada valor separadamente, ou os valores terão um suporte muito baixo e não atingirão o suporte mínimo necessário ou o número de regras resultantes irá ser enorme, tornando difícil a compreensão do total de regras [SIE03]. Enquanto os atributos categóricos representam variáveis discretas, os atributos quantitativos representam variáveis contínuas.

Uma maneira de tratar os atributos quantitativos é utilizar a discretização. Se os atributos forem pré-processados antes da mineração, o problema estará transformado no caso de atributos categóricos. Existe ainda a possibilidade de efetuar a discretização durante a própria fase de mineração (vide [SIE03] para maiores informações). Entretanto, as regras geradas com a efetuação de discretização de atributos passam a ser mais gerais, perdendo-se, desta forma, a capacidade de detalhamento do problema.

Considere o atributo idade composto de variáveis contínuas, podendo variar de 0 a 40. Uma discretização possível seria: faixa 1 = idade{0..18}, faixa 2 = idade{19..25}, faixa 3 = idade{26..40}. Neste exemplo, os valores contínuos de idade foram convertidos em 3 atributos categóricos representados pelas 3 faixas [LOP99].

Os artigos [SRI96, ZHA97, AUM99, FUK96], entre outros, possuem como objeto a extração de regras de associação de atributos quantitativos.

CAPÍTULO 4. SERVIÇOS IMPLEMENTADOS

Este capítulo apresenta diversas propostas relacionadas com a tarefa de geração de regras de associação que foram implementadas. Essas propostas foram implementadas com o objetivo de aprimorar a busca por regras de associação, fornecendo ao usuário da aplicação uma maior variedade de parâmetros para possibilitar um melhor controle sobre as regras de associação a serem geradas.

A implementação tem como princípio fundamental a idéia de fornecer serviços para que a camada superior aos serviços possa utilizá-los da forma mais conveniente. O serviço está relacionado com uma interface entre duas camadas, sendo a camada inferior, a de geração de regras de associação, e a camada superior, a utilizadora do serviço.

A implementação do serviço de geração de regras foi baseada no algoritmo *Apriori* [Figura 3.4, AGR94], utilizando-se a linguagem de programação *Visual Basic 6.0*, e as propostas que foram incorporadas ao algoritmo original serão discutidas nas seções deste capítulo.

4.1 CARACTERÍSTICAS DAS PROPOSTAS E DOS SERVIÇOS IMPLEMENTADOS

O serviço básico implementado foi o de geração de regras de associação. Como subprodutos deste serviço, também foram implementados os serviços de geração de itens freqüentes e de diagnósticos.

Estes serviços e as propostas relacionadas aos mesmos serão discutidos nas seções seguintes.

4.1.2 BASE DE DADOS DE ENTRADA

A base de dados que será analisada por um algoritmo de geração de regras de associação pode estar em diversos formatos [ZAK01]. O formato utilizado pelo algoritmo *Apriori* é o "*basket data*", também conhecido na literatura como formato horizontal [AGR94, ZAK01]. Nesse formato, cada transação é representada por um

registro do banco de dados que possui o identificador da transação, e, em seguida, os itens compreendidos pela transação. O banco de dados *D* da Figura 3.2 está neste formato.

A base de dados utilizada pelo serviço implementado neste trabalho utiliza o formato semelhante ao de uma tabela de um banco de dados relacional. Os atributos da tabela representam os itens e os registros na tabela representam as transações. Desta forma, a geração do arquivo a ser utilizado pela implementação não necessita de nenhuma transformação em especial se a fonte dos dados é um banco de dados relacional. Nesse formato, o banco de dados da Figura 3.2 seria representado pela tabela *T* da seguinte forma [Figura 4.1]:

A	B	C	D	E	F
1	0	1	1	1	0
1	1	1	1	0	1
1	0	0	1	1	0
0	0	0	1	0	0
1	1	0	0	0	0
1	1	0	1	1	0

Figura 4.1 – Tabela *T* que representa a base de dados de entrada para o serviço

Como os atributos A, B, C, D, E e F são atributos binários, o valor 1 representa a ocorrência de um atributo em uma determinada transação e o valor 0 a não ocorrência do mesmo, ou seja, a presença ou não de determinado item em uma transação, respectivamente.

Para utilização do serviço, é necessária a passagem de um arquivo texto contendo a base de dados e de um arquivo de informações sobre os atributos. Para o exemplo anterior, o arquivo de dados seria o seguinte [Figura 4.2]:

```
101110
111101
100110
000100
110000
110110
```

Figura 4.2 – Exemplo de arquivo texto para a entrada de dados

As informações básicas do arquivo sobre os atributos são o nome do atributo, a posição inicial do atributo no arquivo de dados, a quantidade de posições ocupadas pelo atributo e uma informação indicando que o atributo é binário. Na quarta posição de cada linha do arquivo é indicado um “B” para cada um dos atributos binários. Para o banco de dados da Figura 4.2, o arquivo de especificação dos atributos seria da forma dada pela Figura 4.3.

```
[ATRIBUTOS]
A|1|1|B
B|2|1|B
C|3|1|B
D|4|1|B
E|5|1|B
F|6|1|B
```

Figura 4.3 – Exemplo de arquivo texto para a especificação dos atributos

Entretanto, os atributos a serem utilizados pelo serviço implementado não precisam ser atributos binários, podem ser, também, atributos categóricos. Dessa forma, a tabela relacional que representa os dados não possuirá apenas atributos com domínio 0 ou 1.

Considere a tabela *T’* do banco de dados fictício *D’* abaixo [Figura 4.4], onde existem os atributos A, B, C, D, E e F. A, B, E e F são binários e C e D são categóricos. O domínio do atributo C é composto pelas categorias CatC1, CatC2, CatC3, CatC4 e CatC5. O domínio do atributo D é formado pelos elementos CatDX, CatDY e CatDW.

A	B	C	D	E	F
1	0	CatC1	CatDX	1	0
1	1	CatC5	CatDZ	0	1
1	0	CatC3	CatDW	1	0
0	0	CatC2	CatDW	0	0
1	1	CatC1	CatDZ	0	0
1	1	CatC1	CatDX	1	0

Figura 4.4 – Tabela *T’* que representa a base de dados de entrada para atributos binários e categóricos

O arquivo de dados e o arquivo de especificação dos atributos seriam representados respectivamente pelas figuras 4.5 e 4.6

```
10CatC1CatDX10
11CatC5CatDZ01
10CatC3CatDW10
00CatC2CatDW00
11CatC1CatDZ00
11CatC1CatDX10
```

Figura 4.5 – Arquivo texto para a entrada de dados do banco *D'*

Para atributos categóricos, não se exige, além do nome do atributo, da posição inicial do atributo no arquivo de dados e da quantidade de posições ocupadas pelo atributo, a informação indicando que o atributo é categórico. Se a quarta posição da linha do arquivo não for utilizada, o atributo é identificado como um atributo categórico.

```
[ATRIBUTOS]
A|1|1|B
B|2|1|B
C|3|5
D|8|5
E|13|1|B
F|14|1|B
```

Figura 4.6 – Arquivo texto para a especificação dos atributos do banco *D'*

4.1.3 AMOSTRA

O modelo proposto para a implementação supõe que todas as operações necessárias para a geração de regras de associação serão executadas na memória principal. Quanto menos acessos ao disco forem efetuados pelo serviço de geração de regras, menor o tempo de execução do mesmo, pois o tempo de acesso à memória principal é muito menor que o tempo de acesso ao disco.

Para atender a esta particularidade foi utilizada a técnica de amostragem da base de dados. A amostra da base de dados deve necessariamente caber na memória principal do computador onde o serviço de geração de regras será empregado.

Como o tempo de execução do *Apriori* é linear em relação ao tamanho da amostra, a utilização de amostras pequenas terá uma performance melhor que a utilização da base de dados inteira ou de amostras grandes. Entretanto, a amostra utilizada deve refletir as características de toda a base de dados para que a geração de resultados incorretos seja minimizada.

A amostragem é realizada através da seleção aleatória de registros da base de dados. Esta seleção não é realizada de uma só vez em toda a base de dados. Cada registro é selecionado de uma determinada fração da base de dados.

Considere a variável *fator*. Esta variável representa a porcentagem da base de dados *D* que será utilizada como amostra. A base de dados é dividida em *f* frações seqüenciais, onde $f = fator * |D|$. Esta fórmula representa, na verdade, o tamanho da amostra. Em cada uma destas frações, da primeira à última, um registro é selecionado de forma aleatória. Desta forma, qualquer sazonalidade da base de dados provavelmente será mantida em sua amostra. A base de dados em disco só é percorrida uma única vez para a geração da amostra.

Este critério permite que a leitura do arquivo contendo a base de dados seja efetuada de forma muito simples. Se os registros fossem aleatorizados de uma vez só, teríamos que lidar com a situação da seleção do registro na posição *n* e depois do registro na posição *m*, onde $n>m$. Para que o arquivo fosse percorrido somente uma vez, seria exigido um controle adicional para reordenação das posições dos registros a serem selecionados, antes de ser efetuada a seleção propriamente dita.

4.1.4 LEITURA DA BASE DE DADOS DE ENTRADA

Um registro da base de dados de entrada representa uma transação. Como o serviço trabalha com atributos que podem ser categóricos, um item é a combinação do atributo e seu valor respectivo em determinado registro [Seção 3.2]. Desta forma, um determinado atributo que assume *k* valores distintos será interpretado como *k* itens distintos. Para atributos categóricos, todos os valores distintos do domínio que são utilizados irão gerar um item, inclusive o valor nulo.

A exceção ocorre com os atributos identificados como binários [Seção 4.1.2]. Neste caso, um atributo binário da base de dados irá gerar apenas um item, pois somente

o valor “1” do atributo binário é considerado para compor o item. Os valores nulos ou “0” são ignorados para atributos binários.

Para cada registro lido da base de dados de entrada, identifica-se se este registro será aproveitado ou não para a confecção da amostra a ser utilizada pelo serviço de geração de regras [Seções 4.1.9 e 4.1.13]. Ainda que o registro não seja aproveitado para a amostra, o mesmo será tratado para a confecção de estatísticas [Seção 4.1.16] e para o cálculo de erro da amostra [Seção 4.1.7], se o mesmo não for descartado pela segmentação da base de dados [Seção 4.1.9].

Ainda para cada registro lido da base de dados, identificam-se todos os itens encontrados no mesmo. Se o registro for selecionado para compor a amostra, a identificação do registro e todos os seus itens serão incluídos em uma estrutura de dados em memória. Esta estrutura ficará na memória principal até o término da execução do serviço e representará a amostra da base de dados a ser utilizada por todo o serviço. A base de dados de entrada, em arquivo, não será mais manipulada.

4.1.5 MANIPULAÇÃO DOS VALORES NULOS

A princípio, os valores nulos são considerados para a geração de regras tal como os demais valores do domínio de um atributo categórico. Entretanto, se o atributo for binário, apenas o valor “1” é considerado válido. Desta forma, os nulos são automaticamente descartados.

Considerando-se atributos categóricos, existe um parâmetro que pode ser acionado para que não ocorra a geração de regras com valores nulos de atributos. Assim, os valores nulos assumidos pelos atributos serão ignorados e não serão carregados para a base de dados em memória. Para todo o serviço de geração de regras, os valores nulos não existirão, pois os mesmos não irão pertencer à base de dados em memória se o parâmetro afim estiver especificado. Ainda que os nulos sejam ignorados nas regras e não carregados para a base de dados em memória, eles serão lidos normalmente no caso de retorno de estatísticas [Seção 4.1.16].

4.1.6 GERAÇÃO DA LISTA DE ITENS FREQÜENTES F_1

A lista de itens freqüentes com apenas um item (F_1, [Seção 3.4.2]) é extraída de informações resultantes da leitura da base de dados de entrada. Como a base de dados deve ser percorrida para a extração da amostra, este procedimento é aproveitado para outras finalidades. Nesta fase de leitura, a quantidade de registros em que um determinado item aparece é registrada. Desta informação é calculado o suporte de cada um dos itens em relação ao tamanho da amostra. Assim, a lista F_1 de itens freqüentes é extraída dessas informações sobre o suporte dos itens.

Na fase de leitura da base de dados de entrada, também são geradas algumas outras estatísticas sobre a base de dados [Seção 4.1.16].

4.1.7 CÁLCULO DE ERRO DA AMOSTRA

Já que a base de dados é percorrida em sua íntegra na fase de geração da amostra, armazena-se, também, para cada item, a quantidade de registros em que ele aparece na base de dados inteira. Dessa informação, podemos gerar uma lista F_1^T relativa não à amostra, mas em relação a toda a base de dados. Essas informações relativas à base de dados inteira são utilizadas para rejeitar ou aprovar uma amostra.

O serviço implementado possibilita a rejeição de uma amostra se a lista F_1 for diferente da lista F_1^T. Permite, ainda, rejeitar a amostra se a diferença entre o suporte de cada um dos itens freqüentes em F_1 e o suporte em F_1^T for maior que ε (ε é o valor do erro a ser tolerado). Esse valor deve ser passado como parâmetro para o serviço de geração de regras.

Após a extração da amostra, se a mesma for aprovada, a geração dos demais conjuntos de itens freqüentes utiliza apenas a amostra dos dados que se encontra na memória principal.

4.1.8 ESTRUTURA DA BASE DE DADOS EM MEMÓRIA

A amostra dos dados é armazenada em memória na forma de uma lista de vetores de *bits*, aos quais chamaremos de palavras. Cada item distinto identificado na

base de dados lida é associado a uma posição a ser ocupada na palavra. Como cada transação da base de dados é um conjunto de itens, esta transação será armazenada em um conjunto de *bits*, ou seja, em palavras. Cada transação pode ocupar uma ou mais palavras da lista.

Considere que a amostra selecionada seja idêntica ao banco de dados *D'* [Figura 4.4]. Na leitura do primeiro registro, ao ler o item A=1, o serviço associaria a posição 0 da palavra a este item, por ser este o primeiro a ser lido. O próximo item a ser identificado é o B=0, mas como B é um atributo binário, este item seria ignorado. O item seguinte a ser lido, item C=CatC1, seria associado à posição 1. O item D=CatDX seria associado à posição 2, e, por fim, o item E=1 seria associado à posição 3.

Na leitura do segundo registro, o item A=1 já estaria identificado através da leitura da transação anterior e ocuparia a posição 0 da palavra no registro 2. O item B=1 seria associado à próxima posição disponível (posição 4), e assim por diante. As associações entre itens e posições na palavra seriam as seguintes [Tabela 4-1]:

Item	Posição
A=1	0
C=CatC1	1
D=CatDX	2
E=1	3
B=1	4
C=CatC5	5
D=CatDZ	6
F=1	7
C=CatC3	8
D=CatDW	9
C=CatC2	10

Tabela 4-1 – Associações entre itens e posições na palavra dos serviços implementados

Cada uma destas posições representa um *bit* ligado no vetor de *bits*, ou seja, na palavra. O registro 1 possui, portanto, os *bits* 0, 1, 2, e 3 ligados. Significa que as posições 0, 1, 2 e 3 da palavra estarão preenchidas com 1 e as demais com 0. Considerando uma palavra que ocupe, no máximo, 5 posições, de 0 a 4, a palavra relativa à transação 1 seria a "01111". Para efeito de armazenamento, optou-se por armazenar números inteiros. Convertendo a palavra "01111" para inteiro, temos o valor

$1*2^0+1*2^1+1*2^2+1*2^3+0*2^4=15$. Este valor é o que será armazenado para representar o registro lido.

A segunda transação ocuparia as posições 0, 4, 5, 6 e 7. Como a palavra possui apenas 5 posições, não poderíamos utilizar as posições 5, 6 e 7. Neste caso, utiliza-se mais de uma palavra da lista para armazenar o registro. A posição 5 se tornaria a posição 0 da segunda palavra, a posição 6 se tornaria a posição 1 da segunda palavra, e assim por diante. Dessa forma, o registro 2 seria identificado por duas palavras: “10001” e “00111”, o que equivaleria aos valores 17 e 7, respectivamente. A lista de palavras contendo a amostra em memória seria da seguinte forma:

Palavra	Valor
1	15
2	17
3	7
4	9
5	24
6	0
7	16
8	8
9	19
10	2
11	31

Figura 4.7 – Lista de palavras representando a amostra de dados

Existe uma tabela auxiliar, que indica a posição inicial do registro na lista anterior. Seria da seguinte forma:

Transação	Posição Inicial
1	1
2	2
3	4
4	6
5	9
6	11

Figura 4.8 – Tabela auxiliar para a lista de palavras representando a amostra de dados

Como se optou por este tipo de armazenamento, todas as operações de comparação realizadas pelo serviço são feitas através de operadores *booleanos*. Por, exemplo, para identificar se um conjunto de itens candidatos estará presente ou não em uma determinada transação, basta efetuar um "*AND*" entre os dois valores. Neste caso, o resultado deve ser igual ao valor associado ao conjunto candidato. Por exemplo, o item candidato A=1 e B=1, equivale ao valor, $2^0 + 2^4 = 17$. Para o registro 1, fazendo 15 *AND* 17, tem-se o valor 1. Logo, o conjunto de candidatos não está presente neste registro. Para o registro 2, fazendo 17 *AND* 17, tem-se 17. Como o retorno da segunda palavra é 0, resultado de 0 *AND* 7, verifica-se que o conjunto candidato está presente na transação 2, assim como nas transações 5 e 6. Nas transações 3 e 4, verifica-se que o conjunto candidato não está presente.

Na implementação realizada, foi utilizado um vetor de números inteiros para armazenar, em memória, a amostra selecionada a partir dos dados. Cada número inteiro possui 15 posições a serem ocupadas. Embora esta estrutura seja armazenada em um vetor de inteiros, podemos considerá-la como uma base de dados do tipo *Vertical Bitvectors* [ZAK01].

4.1.9 SEGMENTAÇÃO

O serviço implementado permite que seja especificado qual o segmento da base de dados que servirá para a mineração das regras de associação. Esta especificação é feita relacionando-se quais são os domínios válidos para determinados atributos. Se não existir a especificação, todos os registros da base de dados estarão aptos a serem selecionados para a amostra que será utilizada no serviço. Havendo a especificação, ocorre uma pré-seleção dos registros. A pré-seleção de registros é feita verificando se os itens existentes nos registros estão de acordo com a segmentação especificada. Somente os registros pré-selecionados poderão ser utilizados na amostra. Desta forma, a base de dados não precisa ser segmentada fisicamente para ser utilizada pelo serviço. Apenas a especificação do segmento é o bastante. A amostra será formada somente por registros dentro da especificação do segmento.

Considere uma base de dados, por exemplo, contendo o atributo "UF" que se refere à unidade de federação brasileira. O usuário poderia querer segmentar a base de

dados para, depois, realizar a mineração de dados. Esta segmentação poderia ser feita com o objetivo de selecionar somente os registros relacionados às unidades da federação do Sudeste. Registros relacionados com unidades da federação de outras regiões devem, então, ser descartados. Uma abordagem seria segmentar fisicamente a base de dados para a retirada dos registros e utilizar a mesma como origem de dados para a amostra. O que o serviço implementado se propõe, no entanto, é não segmentar a base de dados fisicamente. E sim, especificar quais são os atributos (no caso, somente o "UF") e quais são os valores de domínio válidos para os atributos (no caso, "ES", "MG", "RJ" e "SP"). A partir destas especificações, sem nenhuma alteração na base de dados original, o serviço será capaz de selecionar apenas registros com as condições relacionadas. Desta forma, poderão ser utilizadas várias segmentações diferentes da base de dados sem a necessidade de alteração da mesma.

O serviço não carrega para a base de dados em memória os atributos e seus valores, ou seja, os itens que são especificados para a segmentação da base de dados. Esta política é adotada para evitar que sejam geradas regras triviais com os itens da segmentação no conseqüente da regra.

Uma outra característica da utilização da segmentação é que o tamanho da amostra sugerida pelo usuário pode não ser atingido. Neste caso, o tamanho da amostra será o maior possível desde que menor ou igual ao valor especificado pelo usuário.

O serviço implementado requer um arquivo texto a ser enviado como parâmetro. Para o banco de dados *D'* [Figura 4.4] e para uma segmentação dada pelo aproveitamento dos registros com o atributo "B", com valor igual a "1", e com o atributo "C", com valor igual a "CatC1" ou "CatC5", o arquivo texto seria dado por [Figura 4.9]:

```
[SEGMENTAÇÃO]
B|1
C|CatC1|CatC5
```

Figura 4.9 – Arquivo texto para a especificação do segmento do banco *D'* que servirá para a mineração das regras de associação

4.1.10 DESCARTE DE REGISTROS

O serviço implementado realiza o descarte de registros da base de dados em memória com a finalidade de redução da base de dados a ser analisada. Com uma base de dados menor, o teste dos conjuntos de itens candidatos se torna mais rápido.

A partir do conjunto de candidatos, na etapa de avaliação dos conjuntos de itens freqüentes de tamanho *k*, os registros que possuem pelo menos um conjunto de itens freqüentes é marcado. Somente os registros marcados serão utilizados na etapa posterior, a de avaliação dos itens freqüentes *k+1*. Os registros não marcados são descartados automaticamente na etapa de avaliação. Desta forma, a cada passo do algoritmo, a base de dados sob avaliação se torna menor, diminuindo o tempo de execução total do algoritmo.

4.1.11 HIERARQUIA

Uma hierarquia é uma coleção de conjuntos de itens. Cada conjunto de itens representa um nível da hierarquia. O nível 1, o maior da hierarquia, possui apenas um elemento. Cada elemento do nível *n* pertence a um único elemento do nível *n-1*. O nível mais baixo é composto de itens simples e não de conjunto de itens.

Permite-se que seja enviada ao serviço uma relação de hierarquias para serem consideradas na geração de regras de associação.

No serviço implementado, a hierarquia é útil para organizar o domínio de um atributo. Considere o atributo ESTADO_CIVIL, que possui o seguinte domínio: SOLTEIRO, CASADO, DIVORCIADO, DESQUITADO, UNIDO ESTAVELMENTE, SEPARADO DE FATO e VIÚVO. Podemos organizar o domínio em dois grupos. O primeiro, o grupo dos *unidos*, e o segundo, o grupo dos *não unidos*. Os valores de domínio CASADO e UNIDO ESTAVELMENTE pertencem ao grupo dos *unidos*. Já os valores SOLTEIRO, DIVORCIADO, DESQUITADO, SEPARADO DE FATO e VIÚVO pertencem aos *não unidos*. Podemos, ainda, dizer que DIVORCIADO, DESQUITADO e SEPARADO DE FATO pertencem ao grupo dos *separados*, e este grupo é que pertence ao grupo no nível acima dos *não unidos*. A Figura 4.10 representa a hierarquia formada por estes grupos.

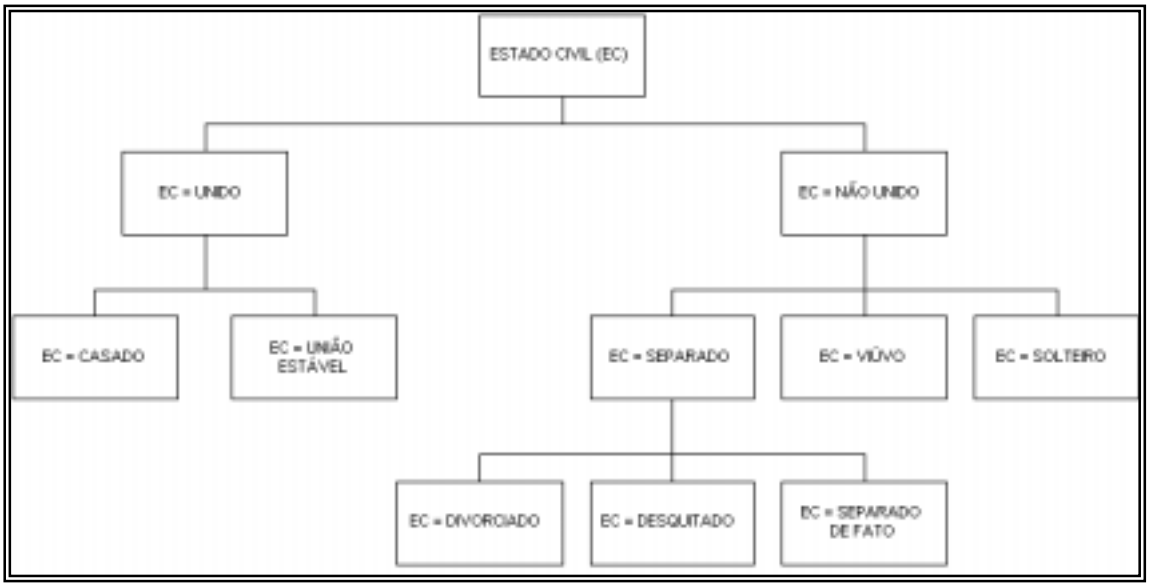


Figura 4.10 – Hierarquia do domínio do atributo ESTADO_CIVIL

Como já visto, a definição da hierarquia pressupõe que um item pertença a apenas um grupo do nível acima. Isto significa que os grupos devem ser mutuamente exclusivos.

A hierarquia que pode ser enviada ao serviço pode ser compreendida na forma de simples agrupamentos de itens de um mesmo atributo, o que seria semelhante a uma discretização no domínio do atributo. Mas não é simplesmente uma discretização, visto que o valor do atributo pertencente ou não a uma hierarquia continua sendo válido para a mineração de dados. Na discretização, ocorre a substituição do valor do atributo por outro valor já discretizado. Na hierarquia, ocorre o acréscimo de mais um item no domínio do atributo. Este item é associado a um valor definido pelo usuário e tem seu acontecimento relacionado à ocorrência de qualquer item que compõe a hierarquia em questão. No exemplo do atributo ESTADO_CIVIL, o grupo dos *unidos* é associado ao valor UNIDO e este valor ocorrerá se o valor CASADO ou se o valor UNIDO ESTAVELMENTE ocorrer. A idéia da hierarquia é permitir o agrupamento de itens, mas não é necessário que os valores agrupados sejam desconsiderados na análise.

A hierarquia também pode servir para gerar regras do tipo $Não\ A \rightarrow B$. Neste caso, basta criar um grupo que defina o domínio $Não\ A$. No caso do exemplo do estado civil, a hierarquia *não unidos* é um exemplo.

A princípio, para o exemplo da Figura 4.10, todos os valores (SOLTEIRO, CASADO, DIVORCIADO, DESQUITADO, UNIDO ESTAVELMENTE, SEPARADO DE FATO, VIÚVO, UNIDO, NÃO UNIDO e SEPARADO) serão válidos como valores de domínio do atributo ESTADO_CIVIL. O usuário, entretanto, pode informar se os valores de domínio agrupados serão ou não considerados na geração de regras de associação. Se apenas os valores relativos aos grupos forem considerados (UNIDO, NÃO UNIDO e SEPARADO), a utilização da hierarquia funcionará como uma discretização. A utilização da hierarquia, tal como uma discretização, pode ser utilizada para atingir o suporte e reduzir as variáveis sob consideração, aumentando a chance de ocorrência de itens freqüentes.

Considere um banco de dados que possua, entre outros atributos, o atributo A e o atributo B. Os valores a_1, a_2 e a_3 fazem parte do domínio do atributo A. Os valores b_1, b_2 e b_3 fazem parte do domínio do atributo B. Considere, ainda, as hierarquias especificadas na Figura 4.11. O serviço exigirá um arquivo texto no formato da figura 4.12 para a especificação das hierarquias a serem consideradas.

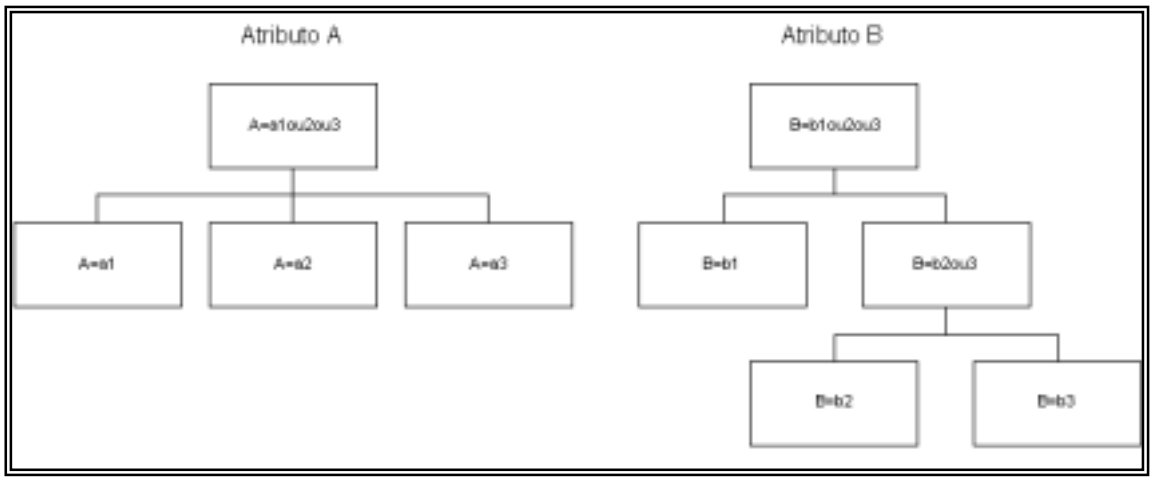


Figura 4.11 – Hierarquia do domínio dos atributos A e B

```
[HIERARQUIA]
A|I|a1ou2ou3|a1|a2|a3
B|H|b2ou3|b2|b3
B|H|b1ou2ou3|b1|b2ou3
```

Figura 4.12 – Arquivo texto para a especificação da hierarquia dos atributos A e B

Na Figura 4.12, o primeiro campo das linhas do arquivo deve ser preenchido com o nome do atributo. O segundo campo especifica se os itens agrupados deverão ser levados em conta ou não para a geração das regras. O “H” indica que os itens agrupados não serão considerados e somente o valor da hierarquia será levada em consideração. Já o “I” significa que, além do valor da hierarquia ser levado em consideração, todos os valores dos itens agrupados também poderão estar presentes nas regras. O terceiro campo é reservado para o valor da hierarquia. Os campos seguintes são reservados aos itens a serem agrupados.

Para cada hierarquia especificada, um item é acrescentado ao domínio do atributo em questão. Na geração da amostra de dados, estes itens são armazenados em memória como se fossem mais um valor do domínio do atributo, tal como os demais valores do domínio. O serviço, então, irá tratar a hierarquia como um item comum na base de dados em memória. Se não for necessário que os itens agrupados em uma hierarquia estejam presentes nas regras, então, esses valores não serão armazenados na base de dados em memória. Somente os valores associados às hierarquias e os demais itens que não tenham sido agrupados serão armazenados.

A quantidade de registros associados a uma hierarquia é dada pela quantidade de registros onde ocorre pelo menos um item que pertença à hierarquia, em qualquer nível inferior. Considere BD_{mem} a amostra em memória do banco de dados e t cada uma das transações. No caso da hierarquia $b_{1ou2ou3}$, a quantidade de registros seria dada por:

$$| b_{1ou2ou3} | = | \{ t \in BD_{mem} : b_1 \subset t \vee b_2 \subset t \vee b_3 \subset t \} | .$$

Entretanto, como o valor da própria hierarquia é armazenado na base de dados em memória, o serviço calcularia diretamente:

$$| b_{1ou2ou3} | = | \{ t \in BD_{mem} : b_{1ou2ou3} \subset t \} | .$$

4.1.12 ATRIBUTOS MULTIVALORADOS

Se a base de dados a ser minerada possuir um atributo multivalorado, este atributo deverá ser decomposto para poder ser enviado ao serviço de geração de regras. Uma primeira abordagem seria a identificação de todo o domínio do atributo multivalorado, onde cada valor distinto se tornaria um atributo binário a mais na tabela.

Entretanto, se o domínio for extenso ou difícil de ser identificado em sua totalidade, podem ser criados *n* atributos do mesmo tipo na tabela, onde cada atributo seria preenchido com um valor distinto do atributo multivalorado (esta seria uma segunda abordagem do problema).

Considere um banco de dados com as informações sobre clientes, onde existe um atributo multivalorado que relaciona os bens do cliente. Além deste atributo, existem atributos que armazenam a idade do cliente e a cidade onde o cliente mora, além de um identificador único. A Figura 4.13 representa a tabela com estas informações.

ID Cliente	Idade	Cidade	Bens
1	25	RIO DE JANEIRO	CARRO
2	30	RIO DE JANEIRO	CARRO APARTAMENTO
3	45	NITERÓI	CARRO CASA SÍTIO
4	40	NITERÓI	MOTO LANCHA APARTAMENTO
5	38	MACAÉ	CARRO CASA

Figura 4.13 – Tabela com informações de clientes

A primeira abordagem, levaria ao envio do seguinte arquivo com a descrição dos atributos para o serviço de geração de regras [Figura 4.14]:

```
[ATRIBUTOS]
IDADE|1|2|C
CIDADE|3|20|C
CARRO|23|1
APARTAMENTO|24|1
CASA|25|1
SÍTIO|26|1
MOTO|27|1
LANCHA|28|1
```

Figura 4.14 – Primeira abordagem para especificação de atributos para banco de dados de clientes

O arquivo de dados a ser enviado deveria ser o seguinte [Figura 4.15]:

```
25RIO DE JANEIRO      100000
30RIO DE JANEIRO      110000
45NITERÓI             101100
40NITERÓI             010011
```

Figura 4.15 – Primeira abordagem para especificação do arquivo de dados para banco de dados de clientes

A segunda abordagem levaria ao envio do arquivo de especificação dos atributos e do arquivo de dados na seguinte forma [Figura4.16; Figura 4.17]:

```
[ATRIBUTOS]
IDADE|1|2|C
CIDADE|3|20|C
BEM1|23|11
BEM2|34|11
BEM3|45|11
```

Figura 4.16 – Segunda abordagem para especificação de atributos para banco de dados de clientes

```
25RIO DE JANEIRO      CARRO
30RIO DE JANEIRO      CARRO      APARTAMENTO
45NITERÓI             CARRO      CASA       SÍTIO
40NITERÓI             MOTO       LANCHA     APARTAMENTO
```

Figura 4.17 – Segunda abordagem para especificação do arquivo de dados para banco de dados de clientes

Para a segunda abordagem, poderia, ainda, ser definido um grupo de atributos que agruparia todos os atributos provenientes da coluna multivalorada. Este grupo serviria para prover o serviço de um mecanismo que identificasse todos os atributos do grupo como um único grande atributo multivalorado. Os atributos pertencentes ao grupo podem ser combinados entre si para a geração de regras.

O agrupamento de atributos pode ser utilizado para atributos multivalorados, o mesmo se aplica para relacionamentos 1-N entre tabelas do banco de dados.

Considere que no banco de dados de clientes [Figura 4.13], o cliente 1 possua um terreno. Neste caso, o arquivo de definição de atributos da segunda abordagem não precisaria mudar e o arquivo de dados seria o da Figura 4.18.

```
25RIO DE JANEIRO      CARRO      TERRENO
30RIO DE JANEIRO      CARRO      APARTAMENTO
45NITERÓI             CARRO      CASA       SÍTIO
40NITERÓI             MOTO       LANCHA     APARTAMENTO
```

Figura 4.18 – 2a abordagem para especificação do arquivo de dados para banco de dados de clientes

O grupo chamado BEM poderia ser especificado para o agrupamento dos atributos BEM1, BEM2 e BEM3. Com esta informação, o serviço de geração de regras utilizaria somente o grupo BEM para a geração de regras se associação. Os atributos BEM1, BEM2 e BEM3 seriam levados em consideração, apenas, para a composição do grupo BEM e não seriam alvo da geração de regras. O arquivo a ser enviado para o serviço de geração de regras com a especificação do grupo seria o seguinte [Figura 4.19]:

```
[GRUPO]
BEM|BEM1|BEM2|BEM3
```

Figura 4.19 – Arquivo de especificação para grupo de atributos

A armazenagem da base de dados em memória seria feita utilizando-se as informações do grupo BEM. Para o registro 1, seriam armazenados os seguintes itens: IDADE=25, CIDADE=RIO DE JANEIRO e BEM=CARRO. Para o registro 2, seriam armazenados: IDADE=30, CIDADE=RIO DE JANEIRO, BEM=CARRO e BEM=APARTAMENTO, e assim sucessivamente [Figura 4.17]. Logo, estes itens é que seriam alvo das regras de associação.

O agrupamento de atributos pode servir para a detecção de relacionamentos entre eventos que ocorreram em tempos diferentes, onde a seqüência dos fatos não é importante.

Diferentemente da hierarquia, o grupo de atributos é considerado um novo atributo e é composto pelos valores existentes no domínio dos atributos que o compõem. A hierarquia é um novo valor de um atributo já existente e é composta pelos valores do domínio do atributo em questão. Ainda de forma diferente da hierarquia, um atributo não pode estar em mais de um grupo diferente.

4.1.13 DIRECIONAMENTO E REDUÇÃO DO NÚMERO DE REGRAS

Para a redução do número de regras de associação a serem geradas pelo algoritmo implementado, existem diversos parâmetros que podem ser utilizados. O objetivo da redução do número de regras geradas é possibilitar, para o usuário, a manipulação de um volume menor de informações ao final da execução do algoritmo. Além deste aspecto, alguns parâmetros estão aptos a direcionar os resultados em relação a determinados atributos previamente especificados.

4.1.13.1 NÚMERO MÁXIMO DE ITENS

O primeiro parâmetro que pode ser empregado é o número máximo de itens de um conjunto de itens freqüentes. Como uma regra com grande quantidade de itens pode se tornar de difícil interpretação, este parâmetro pode ser usado. Se o parâmetro especificado for menor que o tamanho máximo possível de um conjunto de itens freqüentes em uma determinada base de dados, ocorrerá uma redução na quantidade de conjuntos de itens freqüentes. Esta redução implicará em um menor tempo de busca na base de dados, já que um menor número de conjuntos de itens candidatos será testado.

O número máximo de itens no antecedente das regras e o número máximo de itens no conseqüente das regras também podem ser especificados. Estes parâmetros direcionam o formato das regras geradas, facilitando a sua compreensão. Muito comum é a utilização de apenas um único item no conseqüente da regra. Da mesma forma que o parâmetro anterior, esses novos parâmetros também são úteis na fase de geração dos itens freqüentes, pois são utilizados para gerar o parâmetro de número máximo de itens de um conjunto freqüente. Alguns preceitos foram estipulados para reger esses parâmetros:

- Se o número máximo de itens de um conjunto freqüente não for especificado ou for maior que o número máximo de itens no antecedente mais o número máximo de itens no conseqüente, passa a ser fixado por esta soma.

- Se, no entanto, os valores especificados para os parâmetros de número máximo de itens no antecedente ou no conseqüente não forem estipulados ou forem maior ou igual ao valor estipulado para o máximo de itens em um conjunto de itens freqüentes, passam a ser fixados por este valor máximo menos 1.

Os parâmetros que regem o número de itens no antecedente e no conseqüente também reduzem o custo computacional na fase de geração de regras. Por exemplo, se um conjunto $\{A\ B\ C\}$ é freqüente e o número de itens no conseqüente deve ser 1, então, apenas as regras $A, B \rightarrow C$, $A, C \rightarrow B$ e $B, C \rightarrow A$ deverão ter suas confianças testadas. As regras $A \rightarrow B, C$, $B \rightarrow A, C$ e $C \rightarrow A, B$ não serão nem consideradas na verificação.

4.1.13.2 ESPECIFICAÇÃO DE ITENS

Para que as regras geradas sejam direcionadas para o objetivo do usuário, podem ser especificados, também, quais atributos deverão ser utilizados no antecedente das regras e quais atributos deverão ser utilizados no conseqüente das regras. Em se tratando de atributos categóricos, pode, ainda, ser especificado quais são os valores do domínio dos atributos que deverão ser considerados. Se os valores do domínio do atributo não forem determinados, o serviço considera todo o domínio como sendo válido. Ou seja, podem ser especificados somente atributos ou os itens completos. As hierarquias e grupos também podem ser especificados, sendo os mesmos tratados como se fossem atributos ou itens comuns.

Os antecedentes e conseqüentes estipulados são utilizados na fase de leitura da base de dados. Nesta fase, um item é carregado para a base de dados em memória apenas se o atributo respectivo ou item respectivo estiver sido especificado para utilização como antecedente ou como conseqüente.

A especificação destes atributos também reduz a quantidade de conjuntos de itens candidatos. Qualquer conjunto de itens candidatos deve ser um subconjunto do conjunto formado pela união dos atributos e itens especificados para serem levados em conta no antecedente e no conseqüente das regras. Também são utilizados na fase de geração das regras para a verificação se o antecedente e o conseqüente da regra se

encontram respectivamente contidos nos conjuntos de valores possíveis para os antecedentes e para os conseqüentes.

Considere o banco de dados *D'* [Figura 4.4]. Supondo que o antecedente deva ser um subconjunto do conjunto formado pelo atributo A e pelos itens C=CatC1 e C=CatC2 e que o conseqüente deva ser um subconjunto do conjunto formado pelo atributo D, os arquivos a serem enviados para o serviço de geração de regras, com a especificação dos antecedentes e dos conseqüentes, seriam, respectivamente, os seguintes [Figura 4.20; Figura 4.21]:

```
[ANTECEDENTES]
A
C|CatC1|CatC2
```

Figura 4.20 – Arquivo de especificação para os antecedentes das regras

```
[CONSEQUENTES]
D
```

Figura 4.21 – Arquivo de especificação para os conseqüentes das regras

Estes parâmetros serão considerados sempre dentro dos limites dos parâmetros de números máximos.

4.1.13.3 NÚMERO MÁXIMO DE REGRAS

Além dos parâmetros das seções 4.1.13.1 e 4.1.13.2, ainda pode ser especificado o número máximo de regras a serem geradas. Este parâmetro é utilizado após a utilização de todos os outros, inclusive dos parâmetros de seleção de regras interessantes [Seção 4.1.15].

4.1.14 DESCARTE DE COMBINAÇÕES E REGRAS TRIVIAIS

Podem ser enviadas ao serviço de geração de regras, combinações de itens que são triviais para determinado problema. As combinações podem ser enviadas mesmo que não sejam triviais, desde que o usuário não tenha interesse em regras que possuam determinados itens juntos.

As combinações triviais passadas como parâmetro são descartadas pelo serviço na etapa de geração de candidatos. Se um determinado conjunto de itens candidatos possuir um subconjunto que seja uma combinação trivial, o candidato não será avaliado. Desta forma, nenhuma regra contendo os itens de uma combinação trivial será gerada. A implementação resolve, apenas, combinações triviais entre dois itens.

Podem ser enviados dois tipos de combinações possíveis: a primeira, refere-se a combinação entre atributos, e a segunda, refere-se a combinação entre valores de atributos, ou seja, entre itens.

Como exemplo, podemos citar o atributo PAÍS e o atributo ESTADO. De maneira mais genérica, pode ser informado ao serviço que a combinação do atributo PAÍS e do atributo ESTADO é trivial. Refinando mais, pode-se informar que apenas a combinação dos itens PAÍS=“BRASIL” e ESTADO=“RJ” é uma combinação trivial.

De modo semelhante, as regras triviais também podem ser enviadas para o serviço implementado. As regras triviais passadas como parâmetro também serão descartadas pelo serviço de geração de regras, desta vez, na própria fase de geração de regras. De forma similar às combinações triviais, se alguma regra possuir como subconjunto uma regra trivial, então, a regra em avaliação será descartada. Se $A \rightarrow B$ for trivial, então as regras na forma $A,X \rightarrow B,Y$ também serão descartadas. Também pode ser informado se a regra refere-se genericamente a atributos ou a valores específicos dos atributos. Podem ser informadas que as regras serão triviais quando possuírem determinado conseqüente e determinado antecedente, ou ainda, que as regras serão triviais quando possuírem determinado conseqüente, sem informação sobre o antecedente.

Visualizando o exemplo anterior, é fácil perceber que a regra ESTADO=“RJ” $\rightarrow$ PAÍS=“BRASIL” é uma regra trivial. Já a regra PAÍS=“BRASIL” $\rightarrow$ ESTADO=“RJ” não é trivial à primeira vista, mas pode ser para

determinado problema. Pode ser que qualquer regra cujo antecedente seja ESTADO e o conseqüente seja PAÍS também sejam consideradas triviais, independentes do valor que assumirão estes atributos. Qualquer regra trivial citada neste exemplo pode ser enviada ao serviço. Pode-se, ainda, solicitar ao serviço que não gere nenhuma regra cujo conseqüente possua PAÍS="BRASIL". Mais genericamente, a solicitação pode se estender para o atributo PAÍS, qualquer que seja o seu valor. As informações sobre as combinações e regras triviais são passadas através de arquivos texto para o serviço de geração de regras.

```
[COMBINAÇÃO TRIVIAL]
C|1|A||B
C|2|C|c1|D|d1
R|1|D||E
R|2|F|f1|G|g1
R|3|||H
R|4|||I|i1
```

Figura 4.22 – Arquivo texto para a especificação das combinações e regras triviais

A Figura 4.22 exibe um arquivo texto contendo combinações e regras triviais a serem descartadas pelo serviço. Considere que o banco de dados de exemplo possua os atributos A, B, C, D, E, F, G, H e I, pelo menos. Considere, ainda, que c_1 é um valor do domínio do atributo C, assim como d_1 é de D, f_1 é de F, g_1 é de G e i_1 é de I.

Cada linha do arquivo refere-se a uma combinação ou uma regra trivial. Esta informação é passada através do primeiro campo de cada linha, "C" para combinação e "R" para regra. A segunda informação é o tipo de trivialidade a ser passada. Podem ser de quatro tipos:

- Tipo 1- Só é enviada a informação dos dois atributos que não podem ser combinados, no caso de combinação, ou dos dois atributos, tal que o primeiro não pode ser antecedente se o segundo for conseqüente e vice-versa, no caso de regras.
- Tipo 2- Funciona da mesma forma que o tipo 1, porém, também são informados os valores dos atributos.
- Tipo 3- Só existe para regras triviais e é informado, apenas, qual é o atributo que não pode aparecer no conseqüente da regra.

- Tipo 4- Funciona da mesma forma que o tipo 3, porém, é informado, também, o valor do conseqüente.

O descarte de combinações e de regras triviais possui um caráter subjetivo, ao contrário das medidas de interesse associadas às regras que são de caráter objetivo.

4.1.15 SELEÇÃO DE REGRAS INTERESSANTES

Como já visto, no algoritmo implementado (*Apriori*), a seleção de regras é feita, primeiramente, através do cálculo do suporte e da confiança da regra [Seção 3.2]. Logo, os valores do suporte mínimo e da confiança mínima que uma regra deve possuir devem ser especificados para que o serviço de geração de regras possa funcionar. Existem parâmetros específicos para esta finalidade no serviço implementado. Além desses cálculos, o *lift* (ou *interesse*) e a *convicção* das regras também poderão ser calculados, se assim for solicitado [Seção 3.8.1.2].

Existem dois parâmetros que podem ser passados para o serviço de geração de regras implementado em relação ao cálculo do *lift*. O primeiro parâmetro indica se o *lift* será utilizado como um recurso para a seleção de regras. Se este parâmetro for deixado vazio, o serviço não calculará o *lift*. Se o valor passado para este parâmetro for igual à "C" (Calcular), então, o *lift* será calculado e as regras geradas terão a informação sobre o *lift* da mesma. Se o parâmetro for especificado com o valor "D" (Descartar), então, além de calcular o *lift* das regras, aquelas que possuírem *lift* menor que o valor mínimo especificado serão descartadas. Somente as regras remanescentes serão geradas e possuirão a informação de seu *lift*. Por fim, o último parâmetro relacionado com o cálculo é aquele que especifica o valor mínimo do *lift*, para o caso do descarte das regras. Se for deixado em branco, será assumido o valor 1.

A convicção funciona de maneira similar. Existe o parâmetro que indica se a convicção será somente calculada ("C") ou se, além de calculada, a regra será descartada ("D") se a convicção da mesma for maior que a máxima especificada pelo parâmetro que indica a convicção máxima.

4.1.16 SERVIÇOS RELACIONADOS

A ferramenta implementada oferece um outro serviço, o de diagnóstico da base de dados, que objetiva o retorno de determinadas estatísticas identificadas a partir da leitura dos dados.

Este serviço está intimamente relacionado com o serviço de geração de regras, sendo um subproduto do mesmo. O serviço de diagnóstico é gerado na fase de leitura da base de dados para a geração da amostra que será utilizada no serviço de extração de regras de associação. Nesta fase de leitura, as estatísticas sobre os atributos e seus domínios vão sendo armazenadas como parte vital do algoritmo de geração de regras, pois o suporte de cada valor do domínio de um atributo deve ser calculado para a composição da lista F_1 de itens freqüentes. Logo, os diagnósticos relacionam-se com os atributos da base de dados e seus domínios. Tais informações podem ser úteis no processo de descoberta de conhecimento.

O serviço de diagnóstico pode ou não ser solicitado, podendo ser solicitado em conjunto com o serviço de geração de regras ou isoladamente. Pode retornar as seguintes informações úteis para análise da base de dados:

- o percentual existente na base de dados de cada um dos valores lidos do domínio de cada atributo;
- o percentual de registros com *n* atributos preenchidos com valores nulos;
- as combinações de atributos preenchidos com valores nulos e o percentual existente desta combinação.

A partir das informações sobre os atributos e seus domínios, o usuário poderá verificar situações de atributos com domínios extensos e com suporte baixo, com muitos valores nulos, etc. Essas observações podem ser utilizadas pelo usuário de diversas maneiras:

- Podem ser úteis como informação básica para a discretização do domínio do atributo ou como informação para a criação de hierarquias;

- Podem indicar o mau preenchimento de determinados atributos, auxiliando, assim, na limpeza de registros mal preenchidos ou até no seu descarte;
- Podem indicar a necessidade de eliminação do atributo na tarefa de extração de regras, se o preenchimento do mesmo estiver muito ruim e o custo da limpeza for muito alto.

As informações que dizem respeito à quantidade de registros com valores nulos em *n* atributos vêm associadas às informações sobre quais são as combinações destes *n* atributos. Este serviço é uma função adicional ao serviço de geração de regras, pois a busca pelas regras não possui um passo que retorne este tipo de informação. Ainda assim, este serviço é realizado na fase de leitura da base de dados sendo um subproduto desta fase. O conjunto de informações retornado pode sugerir a indicação de mau preenchimento dos registros. Um registro, onde a maioria dos atributos não está preenchida, pode não ser interessante para a geração de regras de associação. Pode ser útil, também, na indicação de existência de registros com nulos em atributos considerados muito importantes para a análise de regras de associação. Desta forma, a informação retornada no diagnóstico do serviço pode ser útil para a eliminação de registros da base de dados ou para a indicação de registros que devam ser devidamente preenchidos. Neste serviço, não são retornados os registros, e sim, uma estatística sobre o quantitativo destes que podem vir a ser problemas para o serviço de geração de regras.

Todos estes diagnósticos não são úteis nos *basket datas*. Nestas bases de dados, os atributos binários representam itens adquiridos em uma determinada transação e estarão sempre relacionados com informações que indicam se o item foi adquirido ou não na transação. Já em outras bases de dados, é interessante que os atributos estejam devidamente preenchidos, pois o atributo é uma informação relevante relacionada à entidade que se está tratando. Ainda que sejam especificados os atributos e os domínios que o serviço de geração de regras estará baseado, o serviço de diagnósticos operará com todos os atributos e todo o domínio dos mesmos.

4.1.17 RESULTADOS DOS SERVIÇOS

Os resultados gerados pelo serviço são armazenados em arquivos texto. São quatro arquivos texto de saída possíveis. Se for solicitado o serviço de diagnóstico da base de dados, será gerado um arquivo com o domínio dos atributos lidos. Considerando o banco de dados *D'* [Figura 4.4], o arquivo de resultados sobre o domínio de cada um dos atributos seria dado pela Figura 4.23. O primeiro campo representa o nome do atributo; o segundo, o valor do atributo; e o terceiro, retorna o percentual deste item em relação à quantidade de registros de *D'*. Se o item em questão representar uma hierarquia ou um grupo de atributos, o arquivo ainda trará um quarto campo que virá identificado por "HIERARQUIA" ou "GRUPO", conforme o caso.

```
A|1|83,33%
B|1|50,00%
C|CatC1|50,00%
C|CatC5|16,66%
C|CatC3|16,66%
C|CatC2|16,66%
D|CatDX|33,33%
D|CatDZ|33,33%
D|CatDW|33,33%
E|1|50,00%
F|1|16,66%
```

Figura 4.23 – Arquivo de saída com os resultados sobre os domínios dos atributos lidos

Ainda se for solicitado o serviço de diagnóstico da base de dados, será gerado um arquivo de saída com os resultados sobre as combinações de itens nulos lidos, contendo o percentual de registros com valores nulos em *n* atributos, as combinações destes *n* atributos e o percentual em relação ao tamanho da base de dados. Considerando uma base de dados dada pela Figura 4.24 a seguir, onde todos os atributos são categóricos, o arquivo de resultados seria dado pela Figura 4.24.

A	B	C	D	E	F
		1	1	1	2
1	1	2	2	2	1
1	2	1			2
2	2	1			2
1	1	2	2	2	2
1				1	1

Figura 4.24 – Base de dados de entrada composta de atributos binários e categóricos

```
2|50,00%
|A|B|16,66%
|D|E|33,33%
3|16,66%
|B|C|D|16,66%
```

Figura 4.25 – Arquivo de saída com os resultados sobre os combinações de itens nulos lidos

A primeira linha do arquivo [Figura 4.25] informa que existem 50% de registros com 2 atributos nulos na base de dados. Na segunda e na terceira linha, são informadas as combinações de 2 atributos que possuem nulos nos mesmos registros da base de dados. No caso exemplificado, a combinação dos atributos A e B possui nulos em 16,66% da base de dados, enquanto a combinação D e E possui 33,33% de nulos nos mesmos registros. A quarta linha do arquivo informa que existem 16,66% de registros com 3 atributos nulos. Na última linha do arquivo é informada qual é a combinação de 3 atributos que possuem nulos nos mesmos registros, que são os atributos B, C e D.

Dependendo da solicitação do usuário, podem ser armazenadas em arquivo as informações sobre os conjuntos de itens freqüentes encontrados. Considerando o banco de dados *D* [Figura 3.2], o arquivo com os itens freqüentes seria dado pela Figura 4.26. A primeira informação de cada linha do arquivo é o tamanho do conjunto de itens freqüentes, seguida pelos nomes dos atributos e seus valores. A última informação de cada linha é o suporte associado ao item freqüente. Os valores associados aos atributos do banco de dados *D* é sempre 1, pois se tratam de atributos binários.

```
2|A|1|B|1|50,00%
2|A|1|D|1|66,66%
2|A|1|E|1|50,00%
2|D|1|E|1|50,00%
3|A|1|D|1|E|1|50,00%
```

Figura 4.26 – Arquivo de saída com os itens freqüentes encontrados

Por fim, pode ser gerado um arquivo com as regras de associação encontradas. Considerando o banco de dados *D* [Figura 3.2], a Figura 4.27 exibe o formato do arquivo a ser gerado. As informações sobre cada regra gerada sempre são armazenadas em duas linhas do arquivo. A primeira é a regra propriamente dita, com as informações

sobre o nome de cada atributo e seu valor associado pertencente ao antecedente da regra. Após estes dados, as informações sobre o nome de cada atributo e seu valor associado pertencentes ao conseqüente da regra também são incluídos na primeira linha. O antecedente e o conseqüente estão separador pelo caractere "»". A segunda linha de cada regra possui informações sobre o suporte e a confiança da regra. Se forem calculados o *lift* e a convicção, ambos serão armazenados na segunda linha de cada regra, após o suporte e a confiança.

```
B|1»A|1
50,00%|100,00%
A|1»D|1
66,66%|80,00%
D|1»A|1
66,66%|80,00%
E|1»A|1
50,00%|100,00%
E|1»D|1
50,00%|100,00%
E|1»A|1|D1
50,00%|100,00%
A|1|E1»D|1
50,00%|100,00%
D|1|E|1»A|1
50,00%|100,00%
```

Figura 4.27 – Arquivo de saída com as regras geradas

4.2 PARÂMETROS DOS SERVIÇOS

A utilização do serviço de geração de regras de associação, do serviço de geração de itens freqüentes ou do serviço de diagnóstico da base de dados pressupõe o envio e o retorno de diversos parâmetros. Nesta seção, listamos todos os parâmetros que devem ou podem ser utilizados na execução dos serviços.

4.2.1 PARÂMETROS DE ENTRADA OBRIGATÓRIOS

Os parâmetros de entrada obrigatórios configuram o limite mínimo de parâmetros a serem enviados ao serviço para que este possa operar. São eles:

- ArquivoDefAtributos: deve ser preenchido com o caminho e nome completo do arquivo texto com a especificação dos atributos da base de dados a ser trabalhada pelo serviço [Seção 4.1.2; Figura 4.3; Figura 4.6].

- ArquivoDados: consiste na informação do nome e caminho completo do arquivo texto que contém a base de dados a ser minerada [Seção 4.1.2; Figura 4.2; Figura 4.5].
- TamanhoArquivoDados: indica a quantidade de transações, ou seja, de registros da base de dados [Seção 4.1.3];
- FatorAmostra: informa o percentual de registros do ArquivoDados que deve ser extraído para a composição da amostra [Seção 4.1.3];
- Suporte: é o suporte mínimo que um conjunto de itens freqüentes deve possuir [Seção 3.2]. Este parâmetro não é obrigatório se apenas o serviço de diagnóstico for solicitado.
- Confiança: é a confiança mínima que uma regra deve possuir [Seção 3.2]. Este parâmetro é obrigatório se o serviço de geração de regras for solicitado.

4.2.2 PARÂMETROS DE ENTRADA OPCIONAIS

Os parâmetros de entrada opcionais são aqueles que podem ou não ser utilizados para determinação de restrições na operação do serviço. Não são essenciais para o funcionamento do serviço em si, mas para determinados problemas devem ser enviados para a geração de regras com qualidades pré-determinadas. São eles:

- TipoResultado: determina se haverá o retorno dos itens freqüentes e das regras de associação nos arquivos de saída correspondentes. Deve ser preenchido com o valor “I” só para retorno dos itens freqüentes, com “R” para retorno somente das regras e com “IR” para retorno dos dois resultados. Se não for utilizado, não haverá retorno dos itens freqüentes e das regras.
- AtivarServiçoDiagnóstico: deve ser preenchido com o valor –1 (*true*) ou com o valor 0 (*false*) e determina se devem ser gerados e armazenados em arquivos as estatísticas relativas à base de dados [Seção 4.1.16].

- IgnorarNulos: deve ser preenchido com o valor 0 (*false*) ou valor –1 (*true*) e determina se os valores nulos encontrados na base de dados deverão ou não ser transportados para a amostra que permanecerá em memória. Se não for preenchido, o serviço assumirá o valor 0 (*false*) e tratará os nulos como qualquer outro elemento do domínio dos atributos [Seção 4.1.5].

- RejeitarAmostraSuporte: a ser preenchido com um número real que indica o erro a ser tolerado na geração da amostra. Se o erro encontrado pelo serviço for maior que o tolerado, a amostra será rejeitada [Seção 4.1.7]. Se não for preenchido, a amostra não deverá ser rejeitada pelo motivo exposto acima.

- RejeitarAmostraListaF1: é também um parâmetro que permite a rejeição da amostra, neste caso, se a lista F_1, gerada pela amostra, for diferente da lista F_1^T, gerada pela base de dados inteira, a amostra será rejeitada pelo serviço [Seção 4.1.7]. Deve ser preenchido com o valor 0 (*false*) ou valor –1 (*true*). Se não for preenchido, a amostra não deverá ser rejeitada pelo motivo exposto acima.

- ArquivoDefSegmentação: deve ser preenchido com o caminho e nome completo do arquivo texto contendo a especificação da segmentação a ser feita na base de dados [Seção 4.1.9; Figura 4.9].

- ArquivoDefHierarquias: deve ser preenchido com o caminho e nome completo do arquivo texto com a especificação das hierarquias a serem levadas em consideração pelo serviço [Seção 4.1.11; Figura 4.12].

- ArquivoDefGruposAtributos: deve ser preenchido com o caminho e nome completo do arquivo texto com a especificação dos grupos de atributos a serem levados em consideração pelo serviço [Seção 4.1.12; Figura 4.19].

- NumMaxItens: a ser preenchido com o número máximo de itens em um conjunto de itens freqüentes [Seção 4.1.13.1].

- NumMaxItensAntec: a ser preenchido com o número máximo de itens permitido no antecedente de uma regra [Seção 4.1.13.1].
- NumMaxItensConseq: a ser preenchido com o número máximo de itens permitido no conseqüente de uma regra [Seção 4.1.13.1].
- ArquivoDefItensAntec: deve ser preenchido com o caminho e nome completo do arquivo texto com a especificação dos atributos a serem incluídos no antecedente das regras [Seção 4.1.13.2; Figura 4.20].
- ArquivoDefItensConseq: deve ser preenchido com o caminho e nome completo do arquivo texto com a especificação dos atributos a serem incluídos no conseqüente das regras [Seção 4.1.13.2; Figura 4.21].
- NumMaxRegras: a ser preenchido com o número máximo de regras a serem geradas pelo serviço [Seção 4.1.13.3].
- ArquivoDefCombTrivial: deve ser preenchido com o caminho e nome completo do arquivo texto com a especificação das combinações e regras triviais a serem eliminadas pelos serviços [Seção 4.1.14; Figura 4.22].
- TipoOperaçãoLift: deve ser preenchido com “C” ou com “D”. Se for preenchido com “C”, o *lift* associado a cada uma das regras será calculado. Se for preenchido com “D”, além do *lift* da regra ser calculado, a mesma será descartada se este *lift* for menor que o parâmetro que indica o *lift* mínimo da regra [Seção 4.1.15].
- Lift: é o *lift* mínimo que uma regra deve possuir caso o parâmetro “TipoOperaçãoLift” seja especificado com “D”. É obrigatório, neste caso [Seção 4.1.15].
- TipoOperaçãoConvicção: deve ser preenchido com “C” ou com “D”. Se for preenchido com “C”, a convicção associada a cada uma das regras será calculada. Se for preenchido com “D”, além da convicção da regra ser calculada, a mesma será descartada se esta convicção for maior que o parâmetro que indica a convicção máxima de uma regra [Seção 4.1.15].

- Convicção: é a convicção máxima que uma regra deve possuir caso o parâmetro “TipoOperaçãoConvicção” seja especificado com “D”. É obrigatório, neste caso [Seção 4.1.15].

4.2.3 PARÂMETROS DE SAÍDA

Os parâmetros de saída são aqueles através dos quais o serviço apresentará os resultados encontrados. São eles:

- ArquivoRegras: deve ser preenchido com o caminho e nome completo do arquivo texto onde serão armazenadas as regras geradas pelo serviço [Seção 4.1.17; Figura 4.27].
- ArquivoItensFrequentes: deve ser preenchido com o caminho e nome completo do arquivo texto onde serão armazenados os conjuntos de itens freqüentes encontrados pelo serviço [Seção 4.1.17; Figura 4.26].
- ArquivoDomínioAtributos: deve ser preenchido com o caminho e nome completo do arquivo texto onde serão armazenadas as informações sobre o domínio dos atributos lidos pelo serviço [Seção 4.1.17; Figura 4.23].
- ArquivoNulos: deve ser preenchido com o caminho e nome completo do arquivo texto onde serão armazenadas os resultados sobre as combinações de itens nulos lidos pelo serviço [Seção 4.1.17; Figura 4.25].

Ainda como parâmetro de saída, o serviço possui o parâmetro utilizado para mensagens:

- MsgErro: é utilizado para fornecimento de mensagens de erro, como no caso de uma amostra rejeitada.

4.2.4 PARÂMETROS DE ENTRADA SOBRE FORMATAÇÃO DE ARQUIVOS

Os parâmetros para a formatação de arquivos dizem respeito aos separadores de informações utilizados nos arquivos textos, tanto nos de entrada como nos de saída. São eles:

- Separador: deve ser preenchido com um caractere. Se não for informado, o serviço utilizará o caractere "|". Este caractere deve ser utilizado para separar as informações em todos os arquivos de parâmetros de entrada, e, também, é utilizado para separar as informações nos arquivos de saída.
- SeparadorEntao: deve ser preenchido com um caractere. Se não for informado, o serviço utilizará o caractere "»". Este caractere é utilizado somente no arquivo de saída definido pelo parâmetro ArquivoRegras [Seção 4.2.3], separando o antecedente do conseqüente da regra.

4.2.5 PARÂMETRO DE APRESENTAÇÃO

O parâmetro de apresentação do serviço pode ser utilizado para exibir uma tela [Figura 4.28] com informações sobre o andamento da execução dos serviços solicitados. Através do parâmetro a seguir, pode ser feita esta solicitação:

- ExibirBarraStatus: deve ser preenchido com o valor –1 (*true*) ou com o valor 0 (*false*) e determina se a tela de andamento do serviço deve ou não ser exibida. Se não for especificado, o valor 0 (*false*) é assumido.

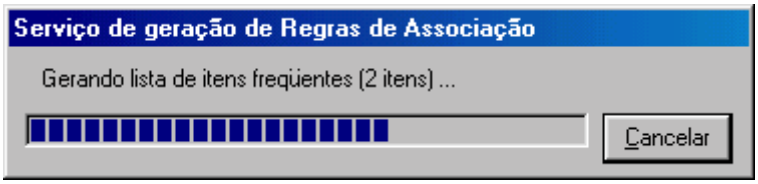


Figura 4.28 – Barra de *Status* exibida pelo serviço implementado

4.2.6 PARÂMETROS INTERNOS

Os parâmetros internos ao serviço, não disponíveis para o usuário, dizem respeito à determinadas limitações dos serviços disponibilizados:

- TamMaxAmostra: o tamanho máximo da amostra a permanecer em memória é de 300.000 transações.
- QuantidadeMaxAtributos: o serviço é limitado em 10.000 atributos por base de dados.
- QuantidadeMaxDominio: o serviço é limitado em 1.000 valores distintos para cada um dos atributos da base de dados;

4.3 ESTRUTURA DOS SERVIÇOS IMPLEMENTADOS

A estrutura dos serviços implementados encontra-se representada na Figura 4.29 a seguir. Nesta figura estão representados os parâmetros de entrada obrigatórios e opcionais e os parâmetros de saída. Existem basicamente três fases para execução dos serviços: a fase de geração da amostra, a fase de identificação dos itens freqüentes e a fase de geração de regras de associação. Os parâmetros de entrada de cada fase, a interação entre as fases e os parâmetros de saída de cada fase, que representam os produtos gerados, estão representados na figura. Todos os parâmetros exibidos na figura já foram discriminados na Seção 4.2.

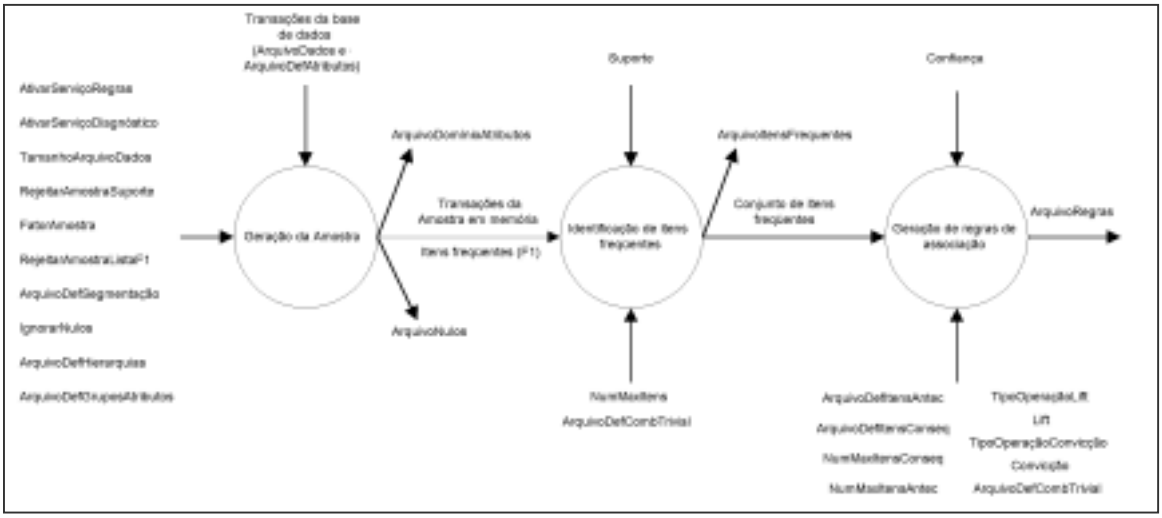


Figura 4.29 – Estrutura dos serviços implementados

4.4 INTERFACE E UTILIZAÇÃO

Os serviços implementados foram disponibilizados através de um arquivo do tipo *dll* (*dynamic link library*) da plataforma Windows. A interface fornecida pelos serviços é o nome da *dll*, *AprioriBit.dll*, e o método *ServicoRegrasAssociacao*, além dos parâmetros, já previamente explicados em seção anterior [Seção 4.2].

A seguir, exibimos um trecho de código em *Visual Basic*, onde o serviço é utilizado [Figura 4.30].

```
Dim ClassApriori As New AprioriBit

Sub AprioriBitDLL()

    ClassApriori.ArquivoDados = "C:\AprioriBitDLL\TesteServiços.txt"
    ClassApriori.ArquivoDefAtributos = "C:\AprioriBitDLL\TesteServiços.atr"
    ClassApriori.TamanhoArquivoDados = 30000
    ClassApriori.FatorAmostra = 1
    ClassApriori.TipoResultado = "IR"
    ClassApriori.IgnorarNulos = True

    ClassApriori.RejeitarAmostraSuporte = 0
    ClassApriori.RejeitarAmostraListaF1 = True

    ClassApriori.GerarEstatísticas = True

    ClassApriori.Suporte = 0.0001
    ClassApriori.Confiança = 0.001
    ClassApriori.TipoOperaçãoLift = "C"
    ClassApriori.Lift = 1.01
    ClassApriori.TipoOperaçãoConvicção = "C"
    ClassApriori.Convicção = 2000000

    ClassApriori.NumMaxItensAntec = 1
    ClassApriori.NumMaxItensConseq = 2
    ClassApriori.ArquivoDefItensAntec = "C:\AprioriBitDLL\TesteServiços.ant"
    ClassApriori.ArquivoDefItensConseq = "C:\AprioriBitDLL\TesteServiços.con"

    ClassApriori.ArquivoDefHierarquias = "C:\AprioriBitDLL\TesteServiços.hrq"
    ClassApriori.ArquivoDefGruposAtributos = "C:\AprioriBitDLL\TesteServiços.gru"

    ClassApriori.ArquivoDefSegmentação = "C:\AprioriBitDLL\TesteServiços.seg"
    ClassApriori.ArquivoDefCombTrivial = "C:\AprioriBitDLL\TesteServiços.trv"

    ClassApriori.Separador = "|"
    ClassApriori.SeparadorEntao = Chr(187)

    ClassApriori.ArquivoRegras = "C:\AprioriBitDLL\Regras.doc"
    ClassApriori.ArquivoItensFrequentes = "C:\AprioriBitDLL\ItensFrq.doc"
    ClassApriori.ArquivoDomínioAtributos = "C:\AprioriBitDLL\Dominio.doc"
    ClassApriori.ArquivoNulos = "C:\AprioriBitDLL\Nulos.doc"

    ClassApriori.ExibirBarraStatus = True

    ClassApriori.ServicoRegrasAssociacao

End Sub
```

Figura 4.30 – Exemplo de código em *Visual Basic* relativo à utilização dos serviços implementados

Não foi construída uma interface gráfica para o usuário final. O objetivo é que os serviços disponíveis através do arquivo *AprioriBit.dll* implementado possam ser incorporados a outros sistemas. Estes sistemas, por sua vez, deveriam prover a funcionalidade gráfica necessária.

A utilização dos serviços pode servir para implementação de diversos sistemas com diferentes objetivos: para retorno de estatísticas relacionadas aos atributos da base de dados e seus domínios; para a geração de itens freqüentes da base de dados; para a geração de regras de associação; para a construção de um classificador utilizando as regras de associação encontradas [SIE03]; para a limpeza da base de dados [ROD03]; para a construção de um modelo para *Cross-selling* fazendo uso das regras de associação [SIE03]; etc.

CAPÍTULO 5. ESTUDOS DE CASOS

Este capítulo descreve os testes realizados utilizando os serviços implementados que foram descritos no Capítulo 4. Foi utilizada uma base de dados do comércio varejista, por ser este tipo de base de dados o alvo inicial dos estudos de extração de regras de associação. Além desta base de dados, utilizou-se a base de dados da Vara de Execuções Penais do Tribunal de Justiça do Estado do Rio de Janeiro [VEP03], experimentou-se, desta forma, mais uma área de atuação para as regras de associação.

Para realização de todos os testes foi utilizado um micro-computador Pentium III de 700 MHz, com 256 MB de memória RAM e sistema operacional Windows 98.

5.1 COMÉRCIO VAREJISTA

A base de dados do comércio varejista utilizada para avaliação dos serviços implementados foi obtida através da Internet. A página do *"KDD Cup 2000"* [KDDCUP03], que é associada à Conferência Internacional em Descoberta de Conhecimento e em Mineração de Dados (*ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*), disponibiliza para a comunidade acadêmica três bases de dados para serem utilizadas na descoberta de regras de associação. A base de dados escolhida para testes contém informações de vários meses de *clickstreams* (registro de acesso) efetuados pelos usuários através do *site Gazelle.com*, uma rede varejista que encerrou as atividades de sua loja virtual em 08/18/2000. A base de dados é chamada de BMS-WebView-1 [BMS03; ZHE01]. Nesta base de dados, cada detalhe de um produto visitado é considerado um item, a Tabela 5-1 exibe informações adicionais sobre esta base de dados.

	Nº de transações	Nº de Itens	Tamanho máximo da transação	Tamanho médio da transação
BMS-WebView-1	59.602	497	267	2,5

Tabela 5-1 – Informações sobre a base de dados BMS-WebView-1

A Tabela 5-2 a seguir apresenta informações referentes ao número de itens freqüentes, ao tamanho do item freqüente mais longo e ao número de regras de

associação geradas pelo algoritmo *Apriori* a partir da base de dados BMS-WebView-1, utilizando valores distintos para o suporte mínimo [ZHE01.2].

Suporte Mínimo (%)	No de Itens Freqüentes	Tamanho do Item Freqüente mais longo	No de Regras de Associação
1,00	77	2	87
0,80	105	2	122
0,60	162	3	195
0,40	286	3	404
0,20	798	4	1.516
0,10	3.991	6	10.360

Tabela 5-2 – Informações sobre a aplicação do algoritmo *Apriori* na base de dados BMS-WebView-1

Para a mesma base de dados, BMS-WebView-1, se o suporte mínimo for muito reduzido, ocorrerá a geração de um número excessivo de itens freqüentes, o que torna impossível a manipulação dos dados. A Figura 5.1 [ZHE01.3], mostra o número de itens freqüentes gerados utilizando o suporte mínimo de 0,06%, 0,04%, 0,02% e 0,01%.

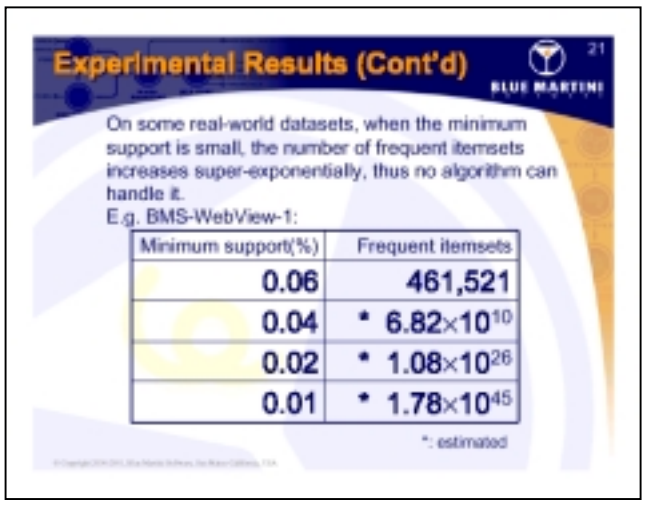


Figura 5.1 – Suporte mínimo aplicado à base de dados BMS-WebView-1

Para validar e verificar a precisão dos serviços implementados, foi utilizada a base de dados BMS-WebView-1 com os mesmos suportes mínimos de 1%, 0,8% e 0,6% e os resultados gerados foram os mesmos da Tabela 5-2 tanto em relação ao

número de itens freqüentes quanto ao tamanho do item freqüente mais longo. Quanto ao número de regras de associação, o resultado dos serviços implementados difere do apresentado na Tabela 5-2 apenas porque não considera os itens freqüentes de tamanho 1 para geração de regras, por supor que os mesmos não podem gerar regras. O artigo [ZHE03_02], no entanto, os considera. Tal fato não indica que os resultados dos serviços estejam incorretos, e sim, mais apurados. Além disto, o artigo também não considera regras com mais de um item no conseqüente das regras, enquanto o serviço implementado pode gerar qualquer quantidade de itens no conseqüente das regras. O número de itens freqüentes de tamanho 1 somado com o número de regras de associação com apenas um item no conseqüente encontradas pelos serviços implementados retorna exatamente o número de regras de associação apresentado no artigo. A Tabela 5-3 resume estas informações.

	Artigo [ZHE01.2]	Serviços implementados		
Suporte Mínimo (%)	No de Regras de Associação	No de Regras de Associação	No de Regras de Associação com apenas 1 item no conseqüente	No Itens Freqüentes com tamanho 1
1,00	87	20	20	67
0,80	122	34	34	88
0,60	195	68	65	130

Tabela 5-3 – Número de regras de associação geradas pelos serviços implementados utilizando a base de dados BMS-WebView-1

5.2 VARA DE EXECUÇÕES PENAIS

A Vara de Execuções Penais (VEP) é uma serventia do Tribunal de Justiça do Estado do Rio de Janeiro (TJRJ) com jurisdição em todo o território do Estado. A VEP tem competência para executar as penas privativas de liberdade e as medidas de segurança detentivas que importem no recolhimento dos réus em estabelecimentos do sistema penitenciário do Estado, para executar as penas restritivas de direito impostas pelas Varas e Juizados Criminais da Comarca da Capital, além de outras atribuições [VEP03].

O banco de dados da VEP é composto por informações relativas aos apenados, que são os indivíduos que possuem alguma pena ou medida a eles aplicada, informações

sobre os processos dos apenados e, também, sobre as ocorrências e os incidentes ocorridos durante a execução da pena, além de contar com as informações sobre o andamento processual.

Para os testes, utilizamos dois subconjuntos da base de dados da VEP: o primeiro teste foi realizado com os dados dos apenados e o segundo com os dados dos processos. Embora os testes realizados e o comportamento tenham sido similares, não irá ser descrito todo o processo para ambos os casos. Como o processo de descoberta de conhecimento foi similar, até mesmo em virtude da semelhança dos dois subconjuntos, fez-se a opção por exibir alguns resultados interessantes relativos a um subconjunto e outros resultados em relação ao outro. Para o caso de apenados será mostrado como o serviço de geração do domínio [Seção 4.1.16] se comportou em relação a diferentes tamanhos de amostras [Seção 4.1.3] e será dada uma ênfase maior no processo de geração de regras de associação. Para o caso de processos iremos mostrar o comportamento do serviço em relação ao número de regras geradas e o tempo de execução em relação a diferentes tamanhos de amostras. Em ambos os casos, algumas regras de associação encontradas pelo serviço implementado serão exibidas.

O Apêndice II contém informações sobre os atributos das bases de dados utilizadas e sobre seus respectivos domínios.

5.2.1 APENADOS

O primeiro subconjunto selecionado do banco de dados da VEP para a mineração de regras de associação foi enfocado, principalmente, nos dados sobre os apenados. Foram considerados dados pessoais sobre o apenado, e, ainda, dados relativos à capitulação dos crimes cometidos pelos mesmos em diversas épocas. Os dados considerados foram relativos ao período de anos de 2000 até 2003.

Após a preparação, a seleção, a limpeza e a transformação dos dados selecionados do banco de dados da VEP, a base de dados dos apenados habilitada para a mineração contou com 26 atributos [Apêndice II] e 35.616 registros. Nesta base de dados, aplicamos os serviços implementados [Capítulo 4].

O serviço de retorno do domínio dos atributos [Seção 4.1.16] mostrou que mesmo em amostras pequenas, o percentual relativo a cada um dos itens permanece

semelhante. Para exemplificar, vide a Tabela 5-3 a seguir, com informações sobre os atributos COR, ESTADO CIVIL, GRAU DE INSTRUÇÃO, IDADE e SEXO:

Item (Atributo/Valor Atributo)		Amostra de 10%	Amostra de 50%	Base de Dados Inteira
COR	BRANCA	34,76%	35,57%	35,42%
COR	PARDA	32,48%	32,71%	32,62%
COR	PRETA	21,86%	20,83%	21,14%
COR	NÃO CONSTA	10,90%	10,89%	10,82%
ESTADO_CIVIL	SOLTEIRO	82,51%	82,42%	82,38%
ESTADO_CIVIL	CASADO	13,33%	13,08%	13,27%
ESTADO_CIVIL	OUTROS	01,42%	01,52%	01,47%
ESTADO_CIVIL	SEPARADO	00,65%	00,95%	00,91%
ESTADO_CIVIL	DIVORCIADO	00,86%	00,83%	00,78%
ESTADO_CIVIL	<NÃO CONSTA>	00,55%	00,60%	00,60%
ESTADO_CIVIL	VIUVO	00,65%	00,55%	00,56%
ESTADO_CIVIL	UNIÃO ESTÁVEL	00,03%	00,05%	00,04%
ESTADO_CIVIL	SEPARADO DE FATO	-	00,01%	00,00%
GRAU_INST	1.GRAU INCOMPLETO	67,89%	65,91%	65,86%
GRAU_INST	1.GRAU COMPLETO	08,96%	10,51%	10,35%
GRAU_INST	ANALFABETO	05,14%	05,59%	05,78%
GRAU_INST	2.GRAU COMPLETO	03,79%	03,63%	03,68%
GRAU_INST	2.GRAU INCOMPLETO	01,82%	01,87%	01,80%
GRAU_INST	SUPERIOR	01,05%	01,03%	00,98%
GRAU_INST	SUPERIOR INCOMPLETO	00,37%	00,36%	00,37%
GRAU_INST	NÃO CONSTA	10,99%	11,09%	11,18%
IDADE	18_24	23,28%	22,23%	22,32%
IDADE	25_30	30,27%	30,05%	30,31%
IDADE	31_40	25,41%	26,92%	26,47%
IDADE	41_50	13,06%	13,08%	13,14%
IDADE	51_...	07,98%	07,73%	07,76%
SEXO	MASCULINO	94,03%	93,55%	93,76%
SEXO	FEMININO	05,97%	06,45%	06,24%

Tabela 5-4 – Configuração do domínio de atributos da base de dados de apenados da VEP

A configuração em percentual sobre o domínio dos atributos auxiliou na tarefa de geração de hierarquias, visto que itens que possuem um suporte muito pequeno não resultariam em seu aparecimento em nenhuma regra válida. Então, foram criadas hierarquias para estes itens que podem ser logicamente agrupados para garantir um

suporte maior através da hierarquia [Seção 4.1.11]. O arquivo de hierarquias utilizado está explícito na Figura 5.2.

```
[HIERARQUIA]
ESTADO_CIVIL|H|NAO_SOLTEIRO|E|V|O|I|P|F|C
ESTADO_CIVIL|H|UNIDOS|CASADO|UNIÃO ESTÁVEL
GRAU_INST|I|ATE_1.GRAU INCOMPLETO|ANALFABETO|1.GRAU INCOMPLETO
GRAU_INST|H|DE_1._A_2.GRAU COMPLETO|1.GRAU COMPLETO|2.GRAU INCOMPLETO|2.GRAU COMPLETO
GRAU_INST|H|SUPERIOR_|SUPERIOR|SUPERIOR INCOMPLETO
IDADE|I|41_...|41_50|51_...
NURC|I|NURC_Interior|NURC 10 - SEDE ITAPERUNA|NURC 11 - SEDE CABO FRIO|NURC 3 - SEDE
PETROPOLIS|NURC 5 - SEDE VOLTA REDONDA|NURC 6 - SEDE CAMPOS|NURC 7 - SEDE
VASSOURAS|NURC 8 - SEDE ITAGUAI|NURC 9 - SEDE NOVA FRIBURGO
NURC|I|NURC_Metropolitano||NURC 2 - SEDE NITEROI|NURC 4 - SEDE DUQUE DE CAXIAS
NURC|I|NURC_GrandeRio|NURC 1 - SEDE RIO DE JANEIRO|NURC 2 - SEDE NITEROI|NURC 4 - SEDE
DUQUE DE CAXIAS
UF|H|DIEFERENTE_RJ|AC|AL|AM|AP|BA|CE|DF|ES|GO|MA|MG|MS|MT|PA|PB|PE|PI|PR|RN|RO|RR|RS|S
C|SE|SP|TO
UF|I|SUDESTE|ES|MG|SP
UF|H|NORDESTE|AL|BA|CE|MA|PB|PE|PI|RN|SE|SP
```

Figura 5.2 – Hierarquias utilizadas na base de dados de apenados da VEP

Além dos atributos e das hierarquias já mencionadas, a mineração através dos serviços utilizou o parâmetro de grupo para reunir todos os atributos relativos à capitulação dos crimes em um grupo chamado CRIME [Seção 4.1.12]. O grupo CRIME reúne os seguintes atributos multivalorados: CRIME1, CRIME2, CRIME3, CRIME4, CRIME5, CRIME6, CRIME7 e CRIME8.

Na base de dados dos apenados, já considerando as hierarquias e grupos, foram encontrados 322 itens (combinação de cada atributo com cada elemento componente do seu domínio).

Foi utilizada, também, a segmentação da base de dados através do domínio FEMININO do atributo SEXO [Seção 4.1.9], pois 93,76% da base de dados utilizada é composta de indivíduos do sexo masculino, portanto, em virtude do suporte mínimo exigido nas regras, os apenados de sexo feminino raramente apareceriam em alguma regra ou seria exigido um esforço computacional maior para minerar tais regras.

Em relação às regras de associação, foram geradas muitas regras óbvias, tais como as regras com UF no antecedente e nacionalidade no conseqüente, estas regras foram aos poucos sendo filtradas através dos parâmetros para redução do número de regras de associação [Seção 4.1.13], inclusive com a utilização do arquivo de combinações triviais [Figura 5.3; Seção 4.1.14]. No decorrer dos testes, os valores de suporte mínimo e de confiança mínima também foram amplamente manipulados para

atingir um resultado satisfatório [Seção 3.2]. Até mesmo o arquivo de definição dos atributos [Seção 4.1.2] foi manipulado no decorrer dos testes, para conter somente os atributos de interesse no momento, diminuindo muito o tempo de execução dos serviços.

```
[COMBINAÇÃO TRIVIAL]
C|1|MUNICIPIO||NURC
C|1|NACIONALIDADE||UF|
C|1|QT_PROCESSOS||QT_PROCESSOS_TOTAL
R|3|||ESTADO_CIVIL|S
R|3|||NACIONALIDADE
R|3|||SEXO|M
R|3|||UF|RJ
```

Figura 5.3 – Arquivo de combinações triviais utilizado na mineração base de dados de apenados da VEP

Para a execução dos serviços de geração de itens freqüentes e de extração de regras de associação a seguir exibidos, o tamanho da amostra utilizada foi de 50% da base de dados. A utilização dos demais parâmetros de entrada dos serviços já foi registrada nos parágrafos anteriores.

5.2.1.1 ITENS FREQÜENTES

Para exemplificar a utilização dos itens freqüentes gerados pelo serviços implementados, exibe-se na Tabela 5-5 o mapeamento dos tipos de delitos mais freqüentes das transações do banco de dados A Figura 5.4 exibe graficamente os valores da tabela.

Item Freqüente	Suporte
CRIME|Contra o Patrimônio	54,88%
CRIME|Contra a Pessoa	26,31%
CRIME|Tráfico de entorpecentes	23,40%
CRIME|Contra a Pessoa|CRIME|Contra o Patrimônio	19,53%
CRIME|Contra o Patrimônio|CRIME|Tráfico de entorpecentes	01,25%
CRIME|Contra a Pessoa|CRIME|Tráfico de entorpecentes	00,55%
CRIME|Contra a Pessoa|CRIME|Contra o Patrimônio|CRIME|Tráfico de entorpecentes	00,43%

Tabela 5-5 – Itens freqüentes da base de apenados relacionados ao grupo CRIME

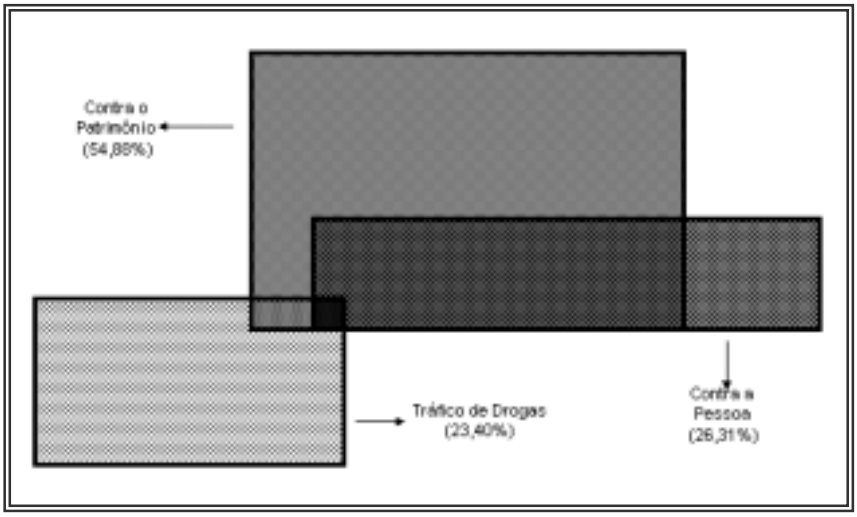


Figura 5.4 – Participação dos delitos contra o patrimônio, contra a pessoa e de tráfico de entorpecentes na base de dados de apenados

5.2.1.2 REGRAS DE ASSOCIAÇÃO

A seguir são exemplificadas algumas regras geradas pelo serviço [Tabela 5-6]. Para apresentação nesta seção foram selecionadas aquelas cuja capitulação dos crimes estivesse presente no conseqüente das regras (grupo CRIME ou atributos a este grupo relacionados). Esta opção se deve ao fato do crime violado pelo apenado ser um conseqüente natural em relação aos demais aspectos existentes e, portanto, as regras com este formato são regras que despertam interesse.

	Regra	Suporte|Confiança da Regra	Observação
1	CRIME|Usuário de drogas»CRIME|Roubo CRIME|Usuário de drogas»CRIME|Tráfico de entorpecentes CRIME|Usuário de drogas»CRIME|Furto CRIME|Usuário de drogas»CRIME|Relacionado com arma de fogo	01,05%|13,10% 00,74%|09,24% 00,55%|06,93% 00,17%|02,16%	Somente crimes ocorridos em processos distintos
2	CRIME|Ocupação ilícita»CRIME|Ocupação ilícita +X CRIME|Furto»CRIME|Furto +X CRIME|Roubo»CRIME|Roubo +X CRIME|Usuário de drogas»CRIME|Usuário de drogas +X CRIME|Homicídio»CRIME|Homicídio +X CRIME|Tráfico de entorpecentes»CRIME|Tráfico de entorpecentes +X	00,10%|32,29% 01,48%|13,09% 03,73%|09,87% 00,49%|05,98% 00,16%|03,17% 00,66%|02,84%	Somente crimes ocorridos em processos distintos
3	CRIME|Lei 9437»CRIME|Tráfico de entorpecentes CRIME|Usuário de drogas»CRIME|Roubo CRIME|Contra a Pessoa »CRIME|Roubo CRIME|Contra a Pessoa »CRIME|Furto CRIME|Roubo»CRIME|Contra a Pessoa CRIME|Furto»CRIME|Contra a Pessoa	03,03%|29,79% 01,36%|15,78% 13,37%|58,77% 06,12%|26,89% 13,37%|34,92% 06,12%|52,97%	
4	NACIONALIDADE|EST»CRIME|Tráfico de entorpecentes NACIONALIDADE|EST»CRIME|Contra o Patrimônio NACIONALIDADE|EST»CRIME|Contra a Pessoa	00,63%|50,50% 00,34%|27,72% 00,25%|19,80%	

	Regra	Suporte|Confiança da Regra	Observação
5	NACIONALIDADE|BRA»CRIME|Tráfico de entorpecentes NACIONALIDADE|BRA»CRIME|Roubo NACIONALIDADE|BRA»CRIME|Contra o Patrimônio NACIONALIDADE|BRA»CRIME|Contra a Pessoa	22,71%|22,99% 39,53%|39,98% 53,87%|54,55% 25,81%|26,14%	
6	NACIONALIDADE|BRA»CRIME|Tráfico de entorpecentes NACIONALIDADE|EST»CRIME|Tráfico de entorpecentes	30,55%|31,53% 02,42%|77,78%	Segmentação da base de dados para apenados do sexo feminino
7	NURC|NURC_Interior»CRIME|Tráfico de entorpecentes NURC|NURC_Interior»CRIME|Roubo NURC|NURC_Interior»CRIME|Contra o Patrimônio NURC|NURC_Metropolitano»CRIME|Tráfico de entorpecentes NURC|NURC_Metropolitano»CRIME|Roubo NURC|NURC_Metropolitano»CRIME|Contra o Patrimônio NURC|NURC_GrandeRio»CRIME|Tráfico de entorpecentes NURC|NURC_GrandeRio»CRIME|Roubo NURC|NURC_GrandeRio»CRIME|Contra o Patrimônio	01,78%|48,00% 00,30%|08,00% 01,23%|33,33% 10,46%|35,45% 04,00%|13,52% 12,59%|42,64% 22,06%|31,24% 14,01%|19,85% 33,02%|46,75%	Segmentação da base de dados para apenados do sexo feminino
8	NURC|NURC_Interior»CRIME|Tráfico de entorpecentes NURC|NURC_Interior»CRIME|Roubo NURC|NURC_Interior»CRIME|Contra o Patrimônio NURC|NURC_Metropolitano»CRIME|Tráfico de entorpecentes NURC|NURC_Metropolitano»CRIME|Roubo NURC|NURC_Metropolitano»CRIME|Contra o Patrimônio NURC|NURC_GrandeRio»CRIME|Tráfico de entorpecentes NURC|NURC_GrandeRio»CRIME|Roubo NURC|NURC_GrandeRio»CRIME|Contra o Patrimônio	01,56%|25,47% 02,20%|35,85% 03,13%|50,94% 04,93%|20,41% 08,93%|37,00% 13,72%|56,81% 17,50%|23,41% 31,06%|41,56% 41,72%|55,84%	Segmentação da base de dados para apenados do sexo masculino
9	QT_PROCESSOS_TOTAL|1»CRIME|Tráfico de entorpecentes QT_PROCESSOS_TOTAL|2»CRIME|Tráfico de entorpecentes QT_PROCESSOS_TOTAL|3+»CRIME|Tráfico de entorpecentes QT_PROCESSOS_TOTAL|1»CRIME|Contra o Patrimônio QT_PROCESSOS_TOTAL|2»CRIME|Contra o Patrimônio QT_PROCESSOS_TOTAL|3+»CRIME|Contra o Patrimônio QT_PROCESSOS_TOTAL|1»CRIME|Contra a Pessoa QT_PROCESSOS_TOTAL|2»CRIME|Contra a Pessoa QT_PROCESSOS_TOTAL|3+»CRIME|Contra a Pessoa	17,76%|26,26% 03,97%|20,28% 01,59%|12,63% 33,33%|49,28% 12,17%|62,12% 09,28%|73,95% 15,53%|22,97% 06,05%|30,90% 04,68%|37,29%	

Tabela 5-6 – Regras de associação selecionadas a partir das regras geradas da base de dados de apenados da VEP

5.2.1.3 INTERPRETAÇÃO DAS REGRAS DE ASSOCIAÇÃO

O primeiro grupo de regras exibe o comportamento dos apenados condenados no Artigo 16 da Lei 6368/76, que são os usuários de drogas. O delito não é considerado um delito grave, entretanto, em muitos casos, o usuário de drogas volta a cometer outros tipos de delitos [Grupo 1 da Tabela 5-6].

O segundo grupo de regras mostra a relação entre a condenação em um determinado tipo de delito e a reincidência no mesmo tipo de delito, levando o apenado a uma nova condenação pelo mesmo motivo [Grupo 2 da Tabela 5-6]. Em geral, quanto mais rígida é a pena aplicada menor o índice de reincidência para o mesmo delito.

O terceiro grupo indica o relacionamento entre alguns tipos de crimes, alguns fortemente ligados, como é o caso do crime contra a pessoa e o crime de roubo, e ainda,

do crime de porte de arma e do crime de tráfico de entorpecentes [Grupo 3 da Tabela 5-6]. A primeira associação pode ser em virtude simplesmente do excesso do crime de furto na base. A segunda associação citada pode ocorrer, no mesmo processo, em virtude da natureza do delito de tráfico de entorpecentes, ou até mesmo, em processos distintos, indicar uma evolução do crime de porte de armas para o crime de tráfico.

O próximo grupo de regras torna patente que se o apenado é estrangeiro existe 50,50% de chance do crime por ele cometido ser de tráfico, entretanto, se for brasileiro, esta confiança decai para 22,99%. Para o crime de roubo ou para os crimes contra a pessoa esta tendência se inverte, já que os brasileiros possuem uma chance mais alta que os estrangeiros de praticar tais delitos [Grupos 4 e 5 da Tabela 5-6].

Com o banco de dados segmentado somente com apenados do sexo feminino, verifica-se que mesmo com nacionalidade brasileira, a probabilidade do crime ser de tráfico sobe para 31,53%, contra os 22,99% não utilizando a segmentação. Para estrangeiras este acréscimo é ainda mais acentuado, 77,78% contra 50,50% do banco todo [Grupo 6 da Tabela 5-6].

Ainda sobre o crime de tráfico cometido pelas mulheres percebe-se visualizando as regras do grupo 7 que o comportamento muda à medida que a região do delito vai saindo do interior para a Capital, embora o suporte das regras cresça, a confiança respectiva diminui [Figura 5.5]. O comportamento relativo aos crimes contra o patrimônio é bem diferente, pois tanto o suporte quanto a confiança aumentam quanto mais próximo o local do delito da Capital [Grupo 7 da Tabela 5-6].

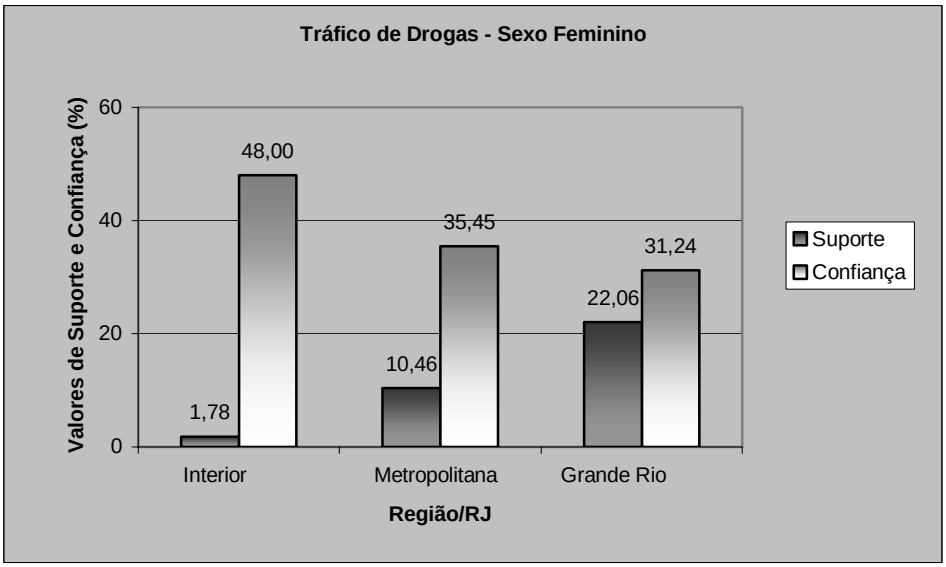


Figura 5.5 – Comportamento do tráfico de drogas com apenados do sexo feminino

Se, no entanto, forem extraídas as regras com o banco de dados segmentado pelo sexo masculino, o comportamento em relação ao tráfico de drogas nas diversas regiões do estado não ocorre semelhante ao grupo de mulheres. A confiança das regras permanece razoavelmente no mesmo patamar, embora o suporte aumente do interior para a capital em virtude do número excessivamente maior de delitos nos grandes centros urbanos [Grupo 8 da Tabela 5-6]. Pode ser observado, também, que enquanto na capital a participação das mulheres é maior no crime relacionado ao tráfico de entorpecentes, a participação masculina é maior no crime de roubo [Grupo 7 e 8 da Tabela 5-6].

As regras utilizando o recurso de segmentação do banco de dados poderiam ser extraídas como regras com mais de um item no antecedente. Por exemplo, a regra "NURC|NURC_Interior»CRIME|Tráfico de entorpecentes" do grupo 7, poderia ser expressa por "SEXO|F|NURC|NURC_Interior»CRIME|Tráfico de entorpecentes", neste caso o suporte seria 0,11% pois todo o banco de dados seria levado em consideração, entretanto, a confiança seria a mesma [Grupo 7 da Tabela 5-6].

O grupo 9 de regras, expressa a informação de que quanto maior o número de condenações maior incidência de delitos com pena mais branda e menor a incidência de delitos com pena mais rígida. Do grupo 2, pode-se extrair informação semelhante.

5.2.2 PROCESSOS

O segundo subconjunto selecionado do banco de dados da VEP para a mineração de regras de associação foi enfocado nos dados dos processos dos apenados. Além da capitulação dos crimes cometidos em cada processo, foram considerados atributos sobre o local da apuração dos fatos, o dia da semana e o mês do delito, e, também, os dados pessoais dos apenados na época do processo. Os dados considerados foram relativos ao período de anos de 2000 até 2003 e, após a preparação, a seleção, a limpeza e a transformação dos dados, a base de dados de processos permaneceu com 42.716 transações e 30 atributos [Apêndice II].

O serviço de geração de regras de associação foi testado utilizando todos os atributos, as hierarquias [Figura 5.6] e o grupo de atributos CRIME utilizado pelo teste na base de apenados [Seção 5.2.1]. Através do serviço de retorno de domínio foram encontrados 626 itens na base de dados [Seção 4.1.16].

```
[HIERARQUIA]
ESTADO_CIVIL|H|NAO_SOLTEIRO|E|V|O|I|P|F|C
ESTADO_CIVIL|H|UNIDOS|CASADO|UNIÃO ESTÁVEL
GRAU_INST|I|ATE_1.GRAU INCOMPLETO|ANALFABETO|1.GRAU INCOMPLETO
GRAU_INST|H|DE_1._A_2.GRAU COMPLETO|1.GRAU COMPLETO|2.GRAU INCOMPLETO|2.GRAU COMPLETO
GRAU_INST|H|SUPERIOR_|SUPERIOR|SUPERIOR INCOMPLETO
IDADE_DELITO|H|41_...|41_50|51_...
REGIAO|I|CIDADE_RIO|TIJUCA|OESTE|NORTE|SUL|CENTRO|MEIER|IRAJA|JACAREPAGUA|LEOPOLDINA|B
ANGU|ILHA|PAN
REGIAO|H|GRANDE_RIO|METROPOLITANA|TIJUCA|OESTE|NORTE|SUL|CENTRO|MEIER|IRAJA|JACAREPAGU
A|LEOPOLDINA|BANGU|ILHA|PAN
REGIAO|H|GRANDE_PAN|OESTE|JACAREPAGUA|PAN
REGIAO|H|GRANDE_ILHA|ILHA|LEOPOLDINA
REGIAO|I|GRANDE_NORTE|TIJUCA|NORTE|MEIER|IRAJA|LEOPOLDINA
UF|H|DIEFERENTE_RJ|AC|AL|AM|AP|BA|CE|DF|ES|GO|MA|MG|MS|MT|PA|PB|PE|PI|PR|RN|RO|RR|RS|S
C|SE|SP|TO
```

Figura 5.6 – Hierarquias utilizadas na base de dados de processos da VEP

5.2.2.1 RESULTADOS SOBRE A UTILIZAÇÃO DE AMOSTRAS

Utilizou-se o suporte em 0.005 e a confiança em 0.05 para testar o serviço utilizando amostras distintas [Seção 4.1.3], com 100%, 50%, 25% e 10% das transações da base de dados. O número de regras geradas e o tempo de processamento para estas amostras foram respectivamente 6.447 em 05:07:23, 6.537 em 02:46:28, 6.623 em 01:20:52 e 6.110 em 00:35:36. Nos mesmos testes, o número de itens freqüentes gerados foi de 4.364, 4.384, 4.356 e 4.230 para as amostras com 100%, 50%, 25% e

10% da base de dados. A Figura 5.7 resume os dados sobre o número de itens freqüentes e de regras.

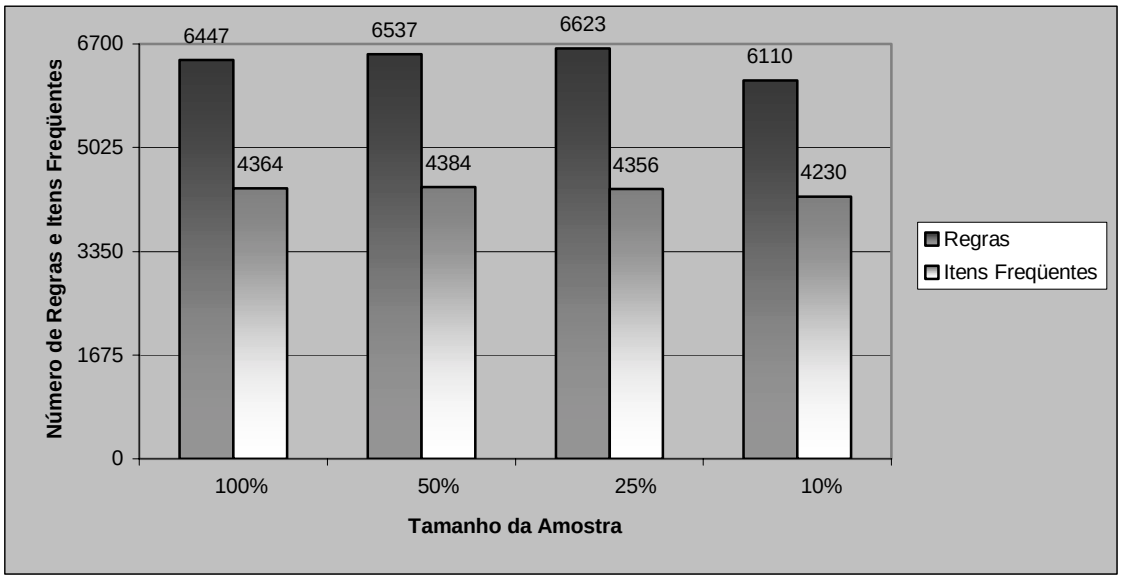


Figura 5.7 – Número de regras geradas para amostras de tamanhos diferentes (suporte em 0.005 e a confiança em 0.05)

Com a mesma configuração de atributos, hierarquias e grupo, utilizando o suporte mínimo de 0.05 e a confiança mínima de 0.30, verifica-se que o número de regras e o tempo de execução diminuem muito. Para amostras de 100%, 50%, 25% e 10% são geradas, respectivamente, 1.177 em 00:42:37, 1.207 em 00:22:34, 1.212 em 00:12:23 e 1.207 em 00:06:04. Nos mesmos testes, o número de itens freqüentes gerados foi de 1.074, 1.079, 1.087 e 1.100 para as amostras com 100%, 50%, 25% e 10% da base de dados. As Figura 5.8 e 5.9 resumem este dados.

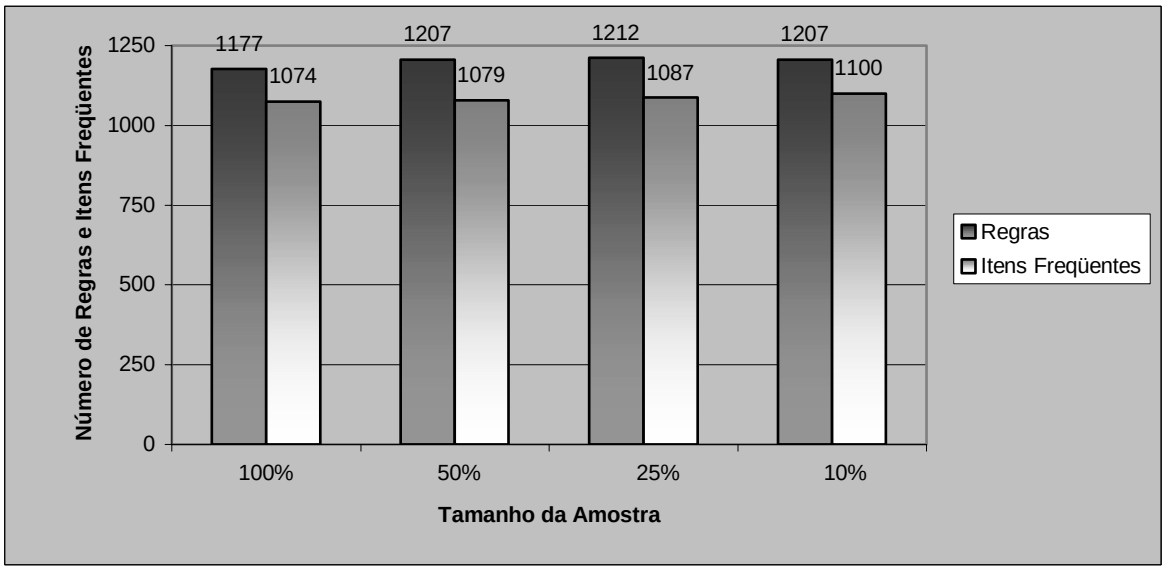


Figura 5.8 – Número de regras geradas para amostras de tamanhos diferentes (suporte em 0.05 e a confiança em 0.30)

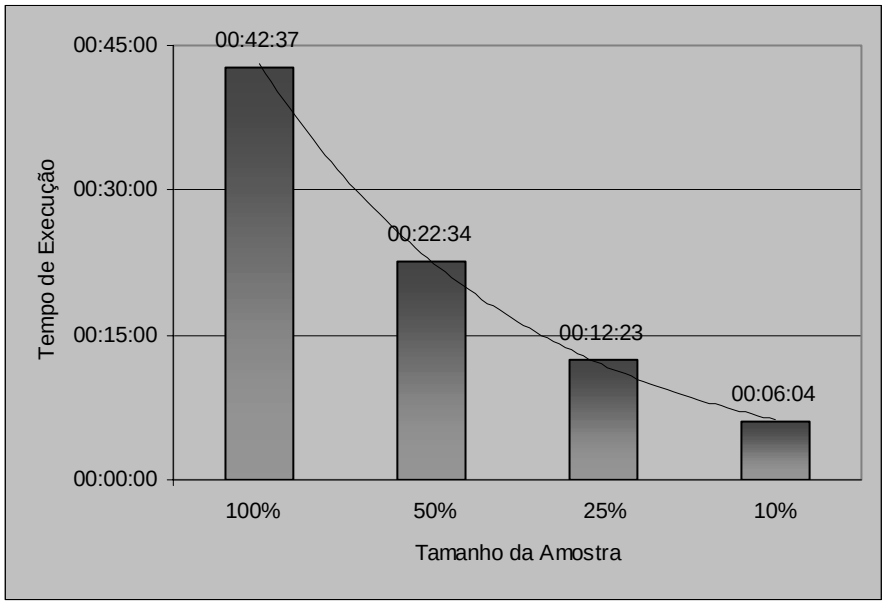


Figura 5.9 – Tempo de execução do serviço para amostras de tamanhos diferentes (suporte em 0.05 e a confiança em 0.30)

Estudando os casos acima [Figuras 5.7 e 5.8], pode-se chegar à conclusão que o número de regras geradas é muito semelhante nas várias amostras utilizadas. Além disto, o tempo de execução do serviço em uma amostra de menor tamanho é bastante

reduzido em relação a amostras maiores [Figura 5.9]. Portanto, existe um maior custo/benefício na execução dos serviços em amostras menores.

A diferença do número de itens freqüentes e de regras é sempre relativo a itens com suporte muito próximo do limite de suporte mínimo ou de regras com confiança muito próxima da confiança mínima. Em uma amostra menor, uma determinada regra que existiria utilizando a base de dados inteira pode não ser gerada ou, ainda, pode existir a geração de uma regra que não iria existir na base de dados inteira. A solução para o primeiro problema é diminuir um pouco o suporte mínimo e a confiança mínima, para o segundo problema, pode-se utilizar a validação das regras na base de dados inteira [Seção 3.5].

Quando o suporte é bem reduzido, amostra muito pequenas, como a de 10%, tendem a gerar um número menor de regras, pois o fato de determinadas transações não serem consideradas na amostra pode impedir a geração de alguma regra. Utilizando amostras maiores este efeito é reduzido, podendo ocorrer, inclusive, o efeito inverso, pois a consideração de algumas transações pode gerar algum tipo de regra, que não seria gerada se todas as transações fossem consideradas. Entretanto, a diferença no número de regras é muito pequena e, além disto, este efeito pode ser eliminado utilizando-se a validação mencionada no parágrafo anterior. Quando se eleva a confiança mínima e o suporte mínimo tal efeito não é mais percebido [Figura 5.8].

5.2.2.2 REGRAS DE ASSOCIAÇÃO

Para a extração das regras, o método de trabalho foi realizado de forma semelhante à realizada no cadastro de apenados, as regras foram sendo filtradas e induzidas através dos parâmetros de geração de regras de associação: do suporte, da confiança, da definição dos conseqüentes e dos antecedentes, das combinações triviais [Figura 5.10], do arquivo de definição de atributos, enfim, através dos parâmetros de entrada.

```
[COMBINAÇÃO TRIVIAL]
C|1|NACIONALIDADE||UF|
C|1|MUNICIPIO||UF
C|1|MUNICIPIO||NURC
C|1|ORGAO_INQ_PROC||UF_INQ_PROC
C|1|PENA||CRIME
R|3|||NACIONALIDADE
R|3|||SEXO|M
R|3|||ESTADO_CIVIL|S
R|3|||UF|RJ
```

Figura 5.10 – Arquivo de combinações triviais utilizado na mineração da base de dados de processos da VEP

Várias regras envolvendo os diversos itens foram geradas e foram avaliadas, principalmente as regras que envolvem os atributos e grupos relacionados ao crime cometido. A seguir, são listados alguns exemplos de regras selecionadas [Tabela 5-7]:

Grupo de Regras	Regra	obs	Suporte|Confiança da Regra
1.	CRIME|Furto»CRIME|Contra a Pessoa (Outros)		05,70%|48,45%
	CRIME|Furto»CRIME|Contra a Pessoa		05,74%|48,72%
	CRIME|Contra a Pessoa (Outros)»CRIME|Furto		05,70%|26,73%
	CRIME|Contra a Pessoa»CRIME|Furto		05,74%|22,30%
	CRIME|Contra a Pessoa (Outros)»CRIME|Roubo		11,44%|53,79%
	CRIME|Contra a Pessoa»CRIME|Roubo		11,48%|44,47%
	CRIME|Roubo»CRIME|Contra a Pessoa (Outros)		11,44%|31,10%
	CRIME|Roubo»CRIME|Contra a Pessoa		11,48%|30,99%
	CRIME|121 do CP»CRIME|Contra a Pessoa (Outros)		01,14%|26,04%
	CRIME|Lei 9437»CRIME|Tráfico de entorpecentes		02,42%|26,47%
	CRIME|Público »CRIME|Contra o Patrimônio		04,55%|52,23%
	CRIME|Contra a Pessoa »CRIME|Contra o Patrimônio		17,90%|72,81%
	CRIME|Contra o Patrimônio »CRIME|Contra a Pessoa		17,90%|31,99%
	CRIME|Contra a Pessoa |CRIME|Público »CRIME|Contra o Patrimônio		01,09%|63,26%
2.	DIA_SEMANA|DOMINGO»CRIME|Roubo		04,01%|36,85%
	DIA_SEMANA|SEGUNDA»CRIME|Roubo		04,65%|37,01%
	DIA_SEMANA|TERÇA»CRIME|Roubo		04,94%|35,66%
	DIA_SEMANA|QUARTA»CRIME|Roubo		05,25%|35,28%
	DIA_SEMANA|QUINTA»CRIME|Roubo		05,25%|34,59%
	DIA_SEMANA|SEXTA »CRIME|Roubo		06,34%|36,87%
	DIA_SEMANA|SÁBADO»CRIME|Roubo		05,03%|37,77%
3.	DIA_SEMANA|DOMINGO »CRIME|Tráfico de entorpecentes		01,96%|18,00%
	DIA_SEMANA|SEGUNDA »CRIME|Tráfico de entorpecentes		02,18%|17,33%
	DIA_SEMANA|TERÇA »CRIME|Tráfico de entorpecentes		02,81%|20,30%
	DIA_SEMANA|QUARTA »CRIME|Tráfico de entorpecentes		03,08%|20,70%
	DIA_SEMANA|QUINTA »CRIME|Tráfico de entorpecentes		03,34%|21,99%
	DIA_SEMANA|SEXTA »CRIME|Tráfico de entorpecentes		03,70%|21,49%
	DIA_SEMANA|SÁBADO »CRIME|Tráfico de entorpecentes		02,72%|20,42%
4.	DIA_SEMANA|DOMINGO»CRIME|Contra a Pessoa		02,85%|26,13%
	DIA_SEMANA|SEGUNDA »CRIME|Contra a Pessoa		02,98%|23,74%
	DIA_SEMANA|TERÇA »CRIME|Contra a Pessoa		02,96%|21,35%
	DIA_SEMANA|QUARTA »CRIME|Contra a Pessoa		03,20%|21,52%
	DIA_SEMANA|QUINTA »CRIME|Contra a Pessoa		03,49%|23,03%
	DIA_SEMANA|SEXTA »CRIME|Contra a Pessoa		03,86%|22,42%
	DIA_SEMANA|SÁBADO »CRIME|Contra a Pessoa		03,31%|24,85%
5.	IDADE_DELITO|18_24»CRIME|Roubo		20,82%|42,34%

Grupo de Regras	Regra	obs	Suporte|Confiança da Regra
	IDADE_DELITO|18_24»CRIME|Tráfico de entorpecentes		11,34%|23,07%
	IDADE_DELITO|18_24»CRIME|Contra a Pessoa		10,74%|21,84%
6.	IDADE_DELITO|25_30»CRIME|Roubo		09,14%|38,47%
	IDADE_DELITO|25_30»CRIME|Tráfico de entorpecentes		04,07%|17,15%
	IDADE_DELITO|25_30»CRIME|Contra a Pessoa		05,84%|24,58%
7.	IDADE_DELITO|31_40»CRIME|Roubo		04,64%|27,27%
	IDADE_DELITO|31_40»CRIME|Tráfico de entorpecentes		02,88%|16,95%
	IDADE_DELITO|31_40»CRIME|Contra a Pessoa		04,34%|25,48%
8.	IDADE_DELITO|41_...»CRIME|Contra o Patrimônio (Outros)		01,81%|17,98%
	IDADE_DELITO|41_...»CRIME|Contra o Patrimônio		03,73%|37,06%
	IDADE_DELITO|41_...»CRIME|Tráfico de entorpecentes		01,60%|15,86%
	IDADE_DELITO|41_...»CRIME|Contra a Pessoa		02,21%|21,94%
9.	REGIAO|IRAJA»CRIME|Contra o Patrimônio		02,34%|59,51%
	REGIAO|MEIER»CRIME|Contra o Patrimônio		02,35%|40,46%
	REGIAO|BANGU»CRIME|Contra o Patrimônio		01,35%|40,56%
	REGIAO|NORTE»CRIME|Contra o Patrimônio		02,65%|49,81%
	REGIAO|TIJUCA»CRIME|Contra o Patrimônio		02,16%|70,91%
	REGIAO|CENTRO»CRIME|Contra o Patrimônio		06,25%|59,73%
	REGIAO|LEOPOLDINA»CRIME|Contra o Patrimônio		01,11%|51,20%
	REGIAO|JACAREPAGUA»CRIME|Contra o Patrimônio		01,15%|50,51%
	REGIAO|SUL»CRIME|Contra o Patrimônio		05,20%|63,32%
	REGIAO|ILHA»CRIME|Contra o Patrimônio		00,79%|55,84%
	REGIAO|OESTE»CRIME|Contra o Patrimônio		01,95%|46,95%
	REGIAO|PAN»CRIME|Contra o Patrimônio		01,16%|74,71%
	REGIAO|GRANDE_PAN»CRIME|Contra o Patrimônio		04,26%|53,37%
	REGIAO|GRANDE_ILHA»CRIME|Contra o Patrimônio		01,90%|53,04%
	REGIAO|GRANDE_NORTE»CRIME|Contra o Patrimônio		10,51%|51,84%
	REGIAO|CIDADE_RIO»CRIME|Contra o Patrimônio		28,36%|54,88%
10.	REGIAO|IRAJA»CRIME|Contra a Pessoa		01,02%|25,87%
	REGIAO|MEIER»CRIME|Contra a Pessoa		01,16%|19,92%
	REGIAO|BANGU»CRIME|Contra a Pessoa		00,46%|13,96%
	REGIAO|OESTE»CRIME|Contra a Pessoa		00,93%|22,44%
	REGIAO|NORTE»CRIME|Contra a Pessoa		01,09%|20,52%
	REGIAO|TIJUCA»CRIME|Contra a Pessoa		01,11%|36,30%
	REGIAO|CENTRO»CRIME|Contra a Pessoa		03,19%|30,45%
	REGIAO|LEOPOLDINA»CRIME|Contra a Pessoa		00,54%|24,82%
	REGIAO|JACAREPAGUA»CRIME|Contra a Pessoa		00,55%|24,12%
	REGIAO|SUL»CRIME|Contra a Pessoa		02,69%|32,70%
	REGIAO|ILHA»CRIME|Contra a Pessoa		00,39%|27,74%
	REGIAO|PAN»CRIME|Contra a Pessoa		00,57%|36,44%
	REGIAO|GRANDE_PAN»CRIME|Contra a Pessoa		02,05%|25,65%
	REGIAO|GRANDE_ILHA»CRIME|Contra a Pessoa		00,93%|25,98%
	REGIAO|GRANDE_NORTE»CRIME|Contra a Pessoa		04,91%|24,22%
	REGIAO|CIDADE_RIO»CRIME|Contra a Pessoa		13,69%|26,49%
11.	REGIAO|IRAJA»CRIME|Tráfico de entorpecentes		00,70%|17,71%
	REGIAO|MEIER»CRIME|Tráfico de entorpecentes		01,28%|22,01%
	REGIAO|BANGU»CRIME|Tráfico de entorpecentes		01,27%|38,14%
	REGIAO|OESTE»CRIME|Tráfico de entorpecentes		01,05%|25,25%
	REGIAO|NORTE»CRIME|Tráfico de entorpecentes		01,16%|21,78%
	REGIAO|TIJUCA»CRIME|Tráfico de entorpecentes		00,31%|10,01%
	REGIAO|CENTRO»CRIME|Tráfico de entorpecentes		01,10%|10,52%
	REGIAO|LEOPOLDINA»CRIME|Tráfico de entorpecentes		00,49%|22,90%
	REGIAO|JACAREPAGUA»CRIME|Tráfico de entorpecentes		00,52%|22,98%
	REGIAO|ILHA»CRIME|Tráfico de entorpecentes		00,23%|16,42%
	REGIAO|SUL»CRIME|Tráfico de entorpecentes		00,70%|08,51%
	REGIAO|JACAREPAGUA»CRIME|Tráfico de entorpecentes		00,54%|23,43%
	REGIAO|GRANDE_PAN»CRIME|Tráfico de entorpecentes		01,62%|20,27%
	REGIAO|GRANDE_ILHA»CRIME|Tráfico de entorpecentes		00,73%|20,33%
	REGIAO|GRANDE_NORTE»CRIME|Tráfico de entorpecentes		03,93%|19,40%
	REGIAO|CIDADE_RIO»CRIME|Tráfico de entorpecentes		08,85%|17,13%
12.	REGIAO|MEIER»CRIME|Usuário de drogas		01,50%|25,89%
	REGIAO|BANGU»CRIME|Usuário de drogas		00,33%|09,91%
	REGIAO|CENTRO»CRIME|Usuário de drogas		01,03%|09,88%
	REGIAO|SUL»CRIME|Usuário de drogas		01,04%|12,68%
	REGIAO|IRAJA»CRIME|Usuário de drogas		00,22%|05,60%
	REGIAO|OESTE»CRIME|Usuário de drogas		00,40%|09,54%
	REGIAO|NORTE»CRIME|Usuário de drogas		00,51%|09,60%

Grupo de Regras	Regra	obs	Suporte|Confiança da Regra
	REGIAO|LEOPOLDINA»CRIME|Usuário de drogas		00,16%|07,55%
	REGIAO|TIJUCA»CRIME|Usuário de drogas		00,25%|08,31%
	REGIAO|JACAREPAGUA»CRIME|Usuário de drogas		00,18%|08,08%
	REGIAO|PAN»CRIME|Usuário de drogas		00,08%|05,32%
	REGIAO|ILHA»CRIME|Usuário de drogas		00,10%|06,75%
	REGIAO|GRANDE_PAN»CRIME|Usuário de drogas		00,66%|08,30%
	REGIAO|GRANDE_ILHA»CRIME|Usuário de drogas		00,26%|07,24%
	REGIAO|GRANDE_NORTE»CRIME|Usuário de drogas		02,65%|13,06%
	REGIAO|CIDADE_RIO»CRIME|Usuário de drogas		05,81%|11,24%

Tabela 5-7 – Tabela de Regras de Associação selecionadas a partir das regras geradas em relação a base de dados de processos da VEP

5.2.2.3 INTERPRETAÇÃO DAS REGRAS DE ASSOCIAÇÃO

O primeiro grupo de regras refere-se aos crimes que estão relacionados entre si em um mesmo ato criminoso. Para exemplificar, pode-se citar que se um delito de furto é infringido então um delito contra a pessoa também é, com 48,45% de confiança. Generalizando esta regra, verifica-se que se um crime qualquer contra o patrimônio é cometido então um delito contra pessoa também é, com 31,99% de confiança. O inverso desta regra, informa que se um crime contra a pessoa ocorre, então também ocorre um delito contra o patrimônio, em 72,81% dos casos, sendo que em 44,47% deles ocorre o delito de roubo [Grupo 1 da Tabela 5-7].

A partir da avaliação dos grupos de regras 2, 3 e 4, verifica-se que não existe muita variação em relação aos dias da semana e a incidência de crimes neste dias. Tanto o suporte quanto a confiança sofrem alterações pequenas entre os dias da semana. A incidência de delitos diminui um pouco no domingo, segunda-feira e terça-feira, mas a diferença ainda assim é pequena. A partir de quarta-feira, o suporte das regras aumenta e chega ao ápice na sexta-feira, a partir daí, começa a cair novamente. O domingo é o dia de menores quantidades de delitos. A Figura 5.11 mostra a incidência dos delitos entre os dias da semana, estes dados foram extraídos do serviço de retorno de domínio implementado [Seção 4.1.16]. Da mesma maneira, a Figura 5.12 mostra a incidência dos delitos entre os meses do ano, onde se enxerga que também não existe uma variação muito grande no percentual de delitos em relação aos meses, sendo que, de uma forma geral, os meses próximos ao final e ao início do ano contribuem ligeiramente menos que os demais meses.

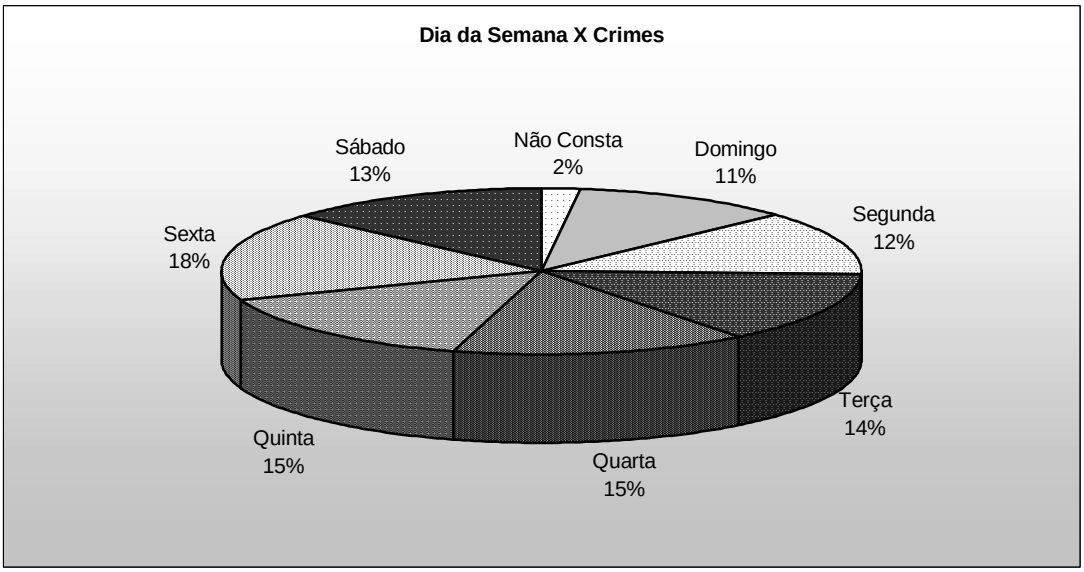


Figura 5.11 – Incidência de delitos em relação aos dias da semana

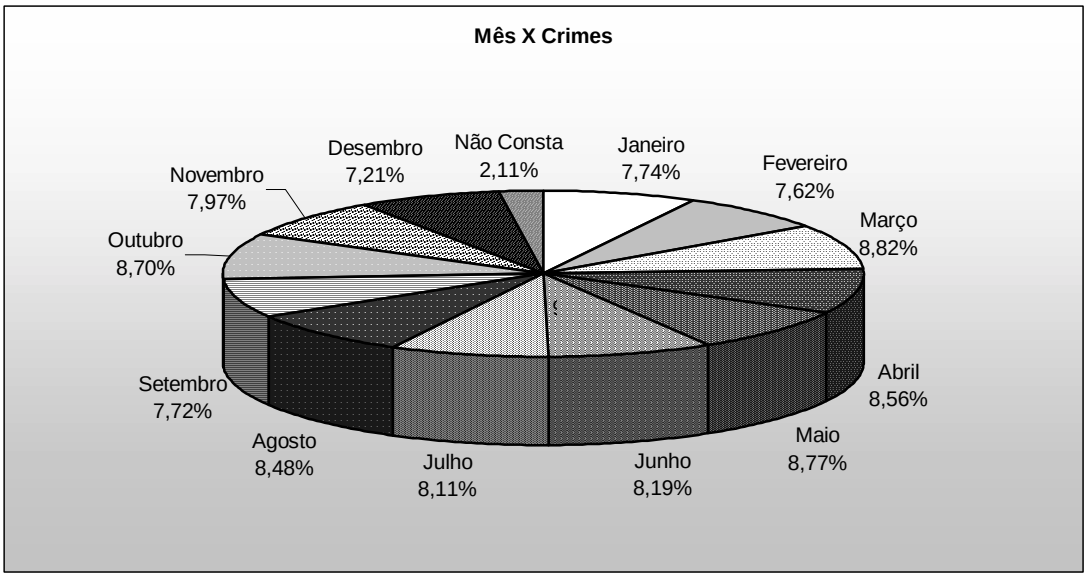


Figura 5.12 – Incidência de delitos em relação aos meses do ano

As regras dos grupos 5, 6, 7 e 8 da tabela 5-7 são relativas aos crimes cometidos e a idade dos apenados à época do delito. Os crimes de maior suporte em todas as faixas etárias são crimes contra o patrimônio. Apenas na faixa etária dos maiores de 40 anos o crime de maior incidência não é o roubo, porém é, ainda, relacionado aos crimes cometidos contra o patrimônio. Os crimes contra o patrimônio constituem o item dominante na base de dados, estando presentes em 55,07% das transações. Para a faixa

de apenados entre 18 e 24 anos, o roubo é seguido do crime de tráfico de entorpecentes e dos crimes contra a pessoa. Nas faixa de apenados mais velhos, esta tendência se inverte. Somente nos crimes contra o patrimônio encontra-se confianças maiores que 30%, para os outros crimes, a confiança é muito baixa. A faixa etária mais nova é responsável pelo maior número de crimes cometidos, a medida que a idade aumenta, a incidência de crimes diminui [Figura 5.13].

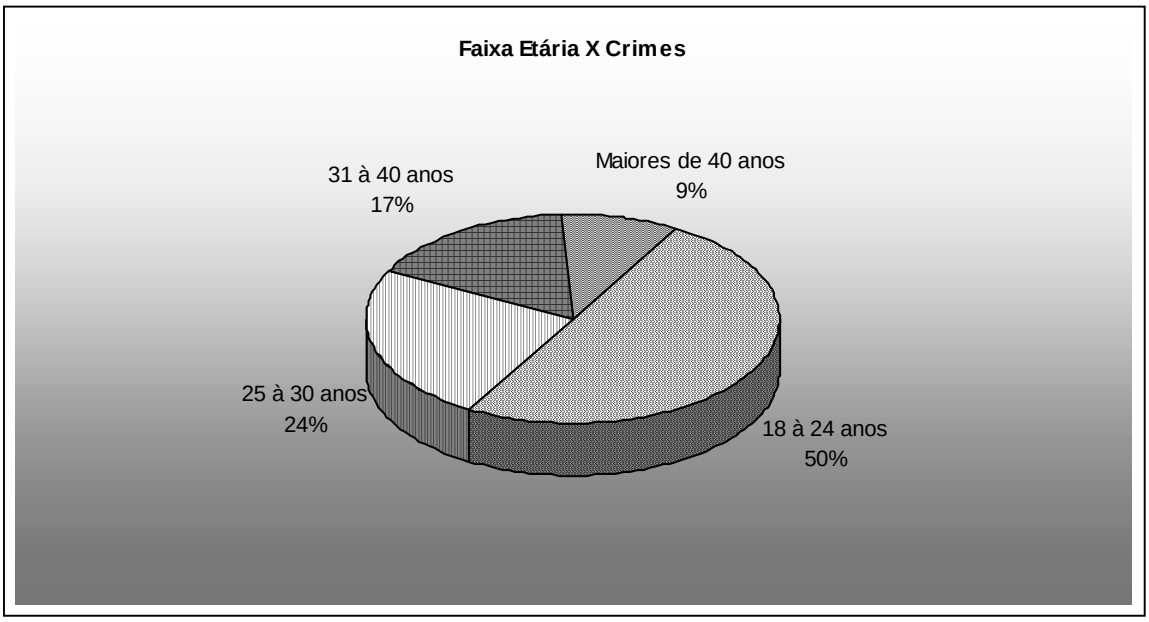


Figura 5.13 – Incidência de delitos em relação à faixa etária do apenado

A Figura 5.14 detém-se apenas na faixa etária dos apenados entre 18 e 24 anos, que é a faixa etária dominante da base de dados, e mostra os percentuais do número de transações existentes de cada tipo de delito cometido por estes indivíduos. As informações foram extraídas do serviço de geração de itens freqüentes [Seção 4.1.17].

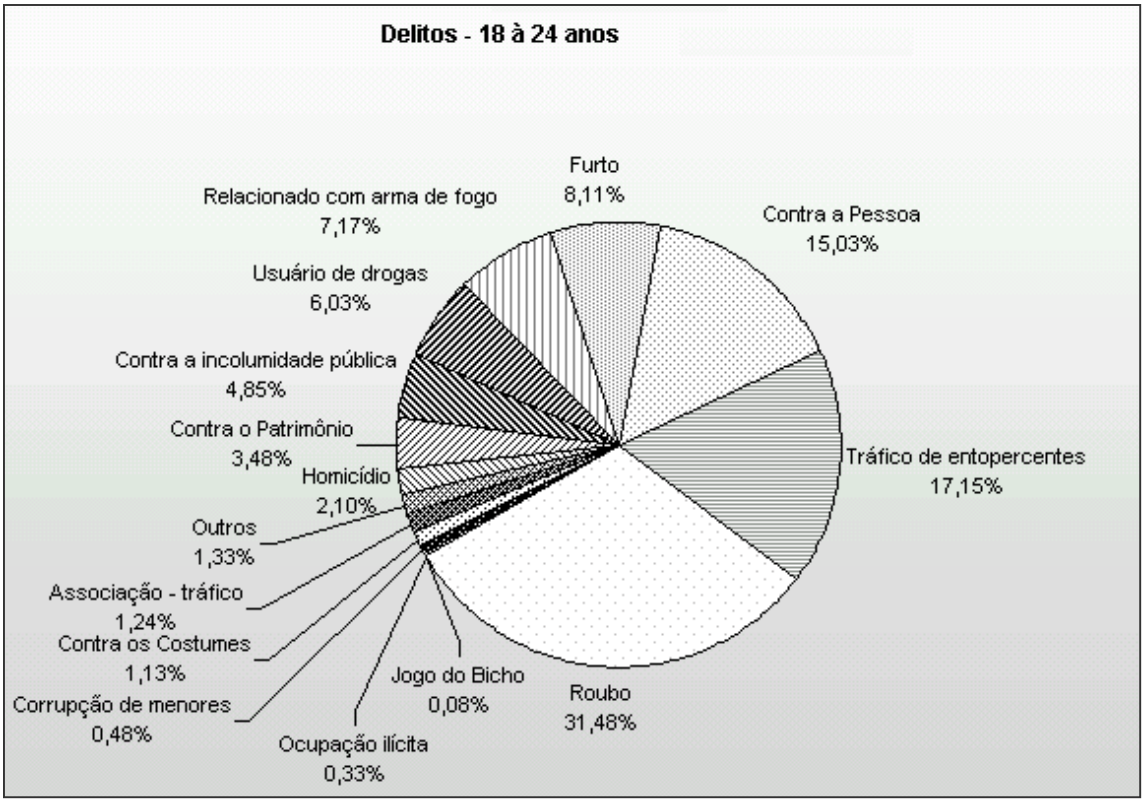


Figura 5.14 – Delitos relativos aos apenados da faixa etária de 18 à 24 anos

As regras dos grupos 9, 10, 11 e 12 são relativas aos delitos cometidos pelos apenados em relação à região do órgão processante do crime e são exibidas, somente, as regras relacionadas com as regiões da cidade do Rio de Janeiro [RIO03]. Observa-se que de acordo com a região, a confiança das regras é bastante alterada.

O crime contra o patrimônio, por exemplo, possui uma confiança muito alta na região da TIJUCA, do CENTRO, do PAN, e da zona SUL, respectivamente 70,91%, 59,73%, 74,71% e 63,32%. Ainda que a confiança relativa a este delito permaneça alta em todas as regiões, esta confiança cai bastante nas outras regiões, por exemplo, no MÉIER onde cai para 46,10% ou em BANGU, onde chega ao patamar de 40,56%. É importante frisar que os delitos relacionados ao patrimônio está presente em 55,07% das transações da base de dados, somente por este fato, a confiança das regras já seria alta.

Os crimes contra a pessoa também possuem confiança mais alta na regiões da TIJUCA, do PAN, do CENTRO e da zona SUL . Em BANGU também ocorre a menor confiança deste tipo de delito, 13,96%, mas cai em outras regiões também, por exemplo,

no MÉIER e na zona NORTE. Os crimes contra a pessoa estão presentes em 24,85% das transações da base de dados.

O crime de tráfico de entorpecentes possui confiança mais alta em BANGU, 38,14%, enquanto as mais baixas se localizam na TIJUCA, no CENTRO e na zona SUL. Nas outras regiões, tal como a região NORTE, a região OESTE e a região de JACAREPAGUÁ, a confiança fica acima de 20%. Já o crime de consumo de entorpecentes tem uma confiança no MÉIER muito mais alta que em todas as demais regiões, 25,89%.

Estas diferenças podem estar relacionadas a diversos fatores: existência de delegacias especializadas na região para apuração de determinados crimes, apuração dos delitos mais ou menos eficiente dependendo das regiões, condenações distintas nas diferentes regiões, existência de delitos pertinentes às regiões, etc. Embora presente na Tabela 5-7, o suporte não foi considerado na análise destas regras pois as dimensões geográficas e populacionais de cada região é muito diversificada, portanto, deveriam ser levadas em consideração para análise do suporte das regras.

5.2.3 COMENTÁRIOS

Algumas regras e conclusões citadas durante toda a seção 5.2 poderiam ter sido tomadas aplicando-se consultas diretamente no banco de dados, entretanto, o especialista deveria ter a capacidade de imaginar todas as situações possíveis para então codificar as consultas a serem aplicadas. Até mesmo como uma opção de substituição das análises construídas diretamente no banco de dados, os serviços se mostraram eficientes.

A forma mais automatizada proporcionada pelos serviços implementados, ainda requer um especialista, até mesmo para o direcionamento dos resultados, mas, principalmente, para a interpretação dos mesmos.

CAPÍTULO 6. CONCLUSÕES E TRABALHOS FUTUROS

O processo de DCBD, como já visto, é composto de diversas etapas ao final das quais é esperado um conhecimento novo, útil e compreensível. Entre as etapas deste processo, destaca-se a mineração de dados sobre a qual foi dirigido este trabalho. É nesta etapa que ocorre a aplicação de algoritmos para extração de padrões que depois de interpretados, transformar-se-ão em conhecimento.

A realização deste trabalho envolveu o estudo de algumas das principais tarefas de mineração de dados, com o enfoque principal na tarefa de mineração de regras de associação e observação das principais características que envolvem esta tarefa. A partir do estudo realizado, um dos objetivos desta dissertação foi definir e propor um ambiente de extração de regras de associação que permitisse flexibilidade para a geração das regras.

O trabalho envolveu 4 atividades principais:

1) Estudo das principais tarefas de mineração de dados e das características fundamentais da mineração de regras de associação.

2) Implementação de propostas disponíveis na literatura e proposição de novos mecanismos capazes de auxiliar o especialista na geração de regras, através do conceito de disponibilização de serviços.

3) Proposta de utilização de extração de regras de associação em base de dados distinta da rede varejista.

4) Estudo de caso que envolveu a elaboração e a execução de todo o processo de DCBD.

CONTRIBUIÇÕES

Podemos citar as seguintes contribuições na área de extração de regras de associação atingidas com a execução de todo o trabalho relacionado com esta dissertação:

- O processo de DCBD e de mineração de dados foi descrito, identificando as operações inerentes ao processo e realizando uma descrição das técnicas que podem ser utilizadas.

- Foi realizado um levantamento bibliográfico específico sobre o tema de mineração de regras de associação, procurando reunir as mais diversas características e propostas relacionadas a esta tarefa.
- Foi realizado um levantamento sobre as ferramentas existentes no mercado para a mineração de regras de associação.
- Foi implementada, através do conceito de serviço, uma solução para a extração de regras de associação, a partir do estudo das técnicas já conhecidas, buscando verificar a aplicabilidade e eficácia do serviço.
- Diversas propostas já existentes ou novas potencialidades foram agregadas ao serviço de geração de regras de associação em forma de parâmetros de entrada do serviço. Podemos citar como exemplo, os parâmetros de grupo de atributos, de descarte de combinações triviais, de descarte de regras triviais, de hierarquia de domínios, de especificação da estrutura, do antecedente e do conseqüente das regras desejadas, de segmentação da base de dados e de utilização de amostragem para permitir a manipulação de um conjunto menor de dados de forma mais eficiente. O serviço implementado permite que o processo de mineração de regras de associação possa ser realizado de forma mais eficiente através da manipulação destes diversos parâmetros, capacitando o especialista na seleção de regras potencialmente mais interessantes.
- Utilizaram-se vetores de *bits* para armazenar os dados em memória, permitiu-se, desta forma, a utilização de apenas operações *booleanas* para comparações de vetores candidatos [Seção 4.1.8].
- Empregou-se no presente trabalho um método de amostragem que, pelo menos nos estudos de casos, mostrou-se apto a gerar amostras com características bem semelhantes à base de dados integral [Seção 4.1.3].
- Identificaram-se e implementaram-se outros serviços que são diretamente relacionados ao processo de extração de regras de associação: geração de itens freqüentes, retorno do domínio dos atributos, hierarquias e grupos, retorno dos itens nulos da base de dados.

- Embora relacionados entre si, os serviços podem ser utilizados de forma independente. O método proposto através da disponibilização de serviços apresenta os seguintes benefícios: reutilização dos parâmetros para bases de dados distintas, reutilização de mesma base de dados para diversos fins e de distintas formas, diversas ferramentas com diferentes propósitos e interfaces podem utilizar os serviços.
- Através do estudo de caso foi possível demonstrar a aplicabilidade e a utilidade de todos os serviços implementados. Além disto, o estudo de caso confirmou que a tecnologia de mineração de dados é adequada para produzir conhecimento a partir de base de dados da área criminal e pode ser um instrumento para avaliar o planejamento e as políticas de segurança pública.

SUGESTÕES E TRABALHOS FUTUROS

A área de mineração de dados e, em especial, a de regras de associação, possui um vasto campo para pesquisas futuras. No contexto dos serviços e estudos de casos verificados no presente trabalho, identificamos as seguintes sugestões para trabalhos futuros:

- A existência de poucos trabalhos relacionados com a base de dados da VEP dificulta comparações com o mecanismo utilizado. Sugerem-se novos estudos para confirmar os resultados desta pesquisa.
- Realizar estudo do custo computacional dos serviços implementados.
- Incorporar outros algoritmos de mineração de regras de associação ao serviço.
- Incorporar no serviço algoritmos para a mineração de *Maximal Frequent Itemsets* e de *Closed Itemsets*.
- Permitir que novos formatos da base de dados de entrada possam ser utilizados pelos serviços.
- Estudar e permitir novos mecanismos de triagem de regras interessantes.

- Realizar a validação das regras geradas pelo serviço em toda a base de dados, o que deverá ser feito automaticamente através de uma nova funcionalidade do serviço.
- Estudar as regras sob o aspecto da temporalidade numa tentativa de agregar a funcionalidade de geração de regras de associação temporais ao serviço.
- Realizar estudo sobre o aspecto incremental das bases de dados com o objetivo de permitir que o serviço receba, apenas, a parte da base de dados em que se deu o incremento, e, ainda assim, possa produzir regras relativas à toda base de dados.
- Acrescentar ao serviço implementado um novo processo para auxílio na análise das regras geradas [Figura 4.29].

APÊNDICE I. FERRAMENTAS EXISTENTES

Este apêndice se propõe a listar algumas ferramentas disponíveis no mercado que trabalham com a geração de regras de associação. É feito, também, um detalhamento maior de duas ferramentas escolhidas dentre as listadas. As ferramentas foram escolhidas por terem distribuição livre e serem referenciadas e utilizadas em teses e artigos sobre a extração de regras de associação [GON01, JOR03].

7.1 LISTA DE FERRAMENTAS

A Tabela 7-1 lista algumas ferramentas existentes para a mineração de regras de associação. A tabela possui informações sobre o nome da ferramenta e de seu desenvolvedor, sobre a forma de distribuição da ferramenta e, também, exibe o endereço na *Web* com maiores informações sobre a ferramenta.

	Nome (Desenvolvedor)	Distribuição	Informações
1.	Aira Data Mining (Hycones IT)	Comercial	www.hycones.com.br
2.	Apriori (Christian Borglet)	Livre	fuzzy.cs.uni-magdeburg.de/~borgelt/software.html#assoc
3.	ARMiner (UMass-Boston)	Livre	www.cs.umb.edu/~laur/ARMiner/
4.	Artool (UMass-Boston)	Livre	www.cs.umb.edu/~laur/ARtool/
5.	CBA (Bing Liu/ National University of Singapure)	Livre	www.comp.nus.edu.sg/~liub/
6.	Clementine (SPSS)	Comercial	www.spss.com/clementine/
7.	D-Miner (Dialogis)	Comercial	www.dialogis.de/
8.	Data Mining Suite (Information Discovery Inc.)	Comercial	www.datamining.com/dmsuite.htm
9.	Easy Miner (UMIST, United Kingdom)	Livre	www.co.umist.ac.uk/~koundour/index.html
10.	Eclat (Christian Borglet)	Livre	fuzzy.cs.uni-magdeburg.de/~borgelt/software.html#assoc
11.	Enterprise Miner (SAS)	Comercial	www.sas.com/products/miner/
12.	HyperParallel Inc.	Comercial	www.hyperparallel.com
13.	Intelligent Miner (IBM)	Comercial	www-3.ibm.com/software/data/iminer/
14.	IREX (G.I. Webb and Associates Pty Ltd)	Comercial	www.giwebb.com/
15.	Magnus Opus (GI Weeb & Associates)	Comercial	www.giwebb.com/
16.	Mineset (Silicon Graphics)	Comercial	www.sgi.com/software/mineset.html
17.	Nuggets Suite (Data Mining Technologies Inc.)	Comercial	www.data-mine.com/bin/site/wrappers/splash.asp
18.	PolyAnalist (Megaputer)	Comercial	www.megaputer.com/
19.	SuperQuery (Azmy)	Comercial	www.azmy.com/
20.	TMiner Personal Edition (University of Granada)	Livre	frontdb.ugr.es

	Nome (Desenvolvedor)	Distribuição	Informações
21.	Xaffinity (Exclusive Ore)	Comercial	www.xore.com/
22.	Xpertrule Miner (Attar)	Comercial	www.attar.com/
23.	WEKA (Univ. Waikato)	Comercial	www.cs.waikato.ac.nz/ml/weka/
24.	WizRule (Wizsoft)	Comercial	www.wizsoft.com/

Tabela 6-1 - Ferramentas para extração de regras de associação

Além da utilização dos *sites* da Tabela 7-1, a pesquisa sobre estas ferramentas foi realizada, principalmente, através dos *sites* www.kdnuggets.com [KDN03], dos artigos [GOE99] e[JOR03], e da tese [GON01].

7.2 DETALHAMENTO DO MAGNUS OPUS®

A ferramenta testada foi a Magnus Opus Demo Version 1.3, ano: 1999-2001. A versão demo só pode ser utilizada para banco de dados com até 1000 transações.

O produto reconhece três tipos de arquivos texto para a entrada dos dados.

Existem diversos parâmetros que podem ser utilizados para configurar a mineração de regras de associação. É possível escolher os itens que poderão aparecer no antecedente e no conseqüente da regra, e, também, o número máximo de itens permitidos no antecedente. Existe a possibilidade de descarte de regras triviais, improdutivas ou insignificantes, para maiores detalhes ver [MAG03]. O número máximo de regras a ser gerado também pode ser especificado.

As medidas de interesse de suporte, confiança e *lift* também podem ser utilizadas como parâmetros. Além destas, o *coverage* (quantidade de transações que contém o antecedente da regra), entre outras, também podem ser utilizados.

A tela de configuração dos parâmetros do Magnus Opus é mostrada na figura 7.1.

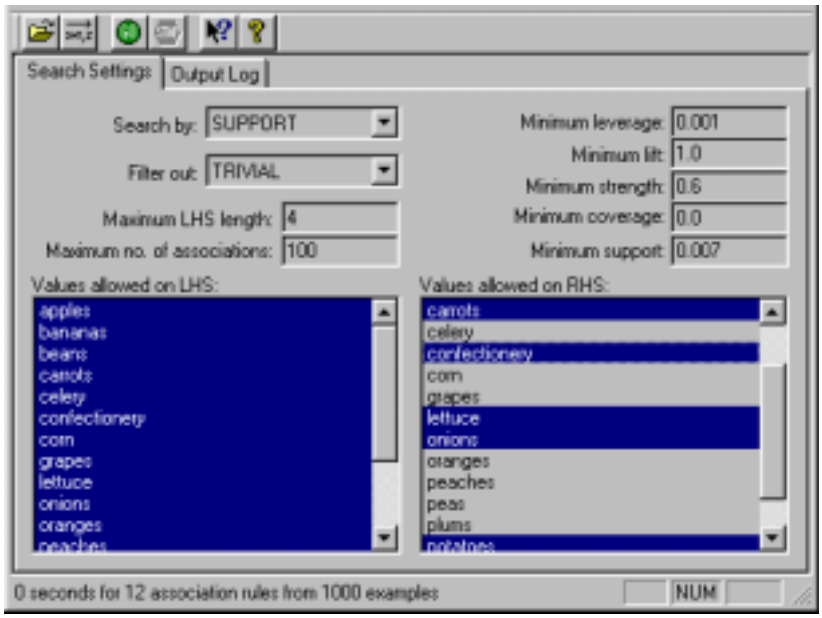


Figura 1.1– Parâmetros do Magnus Opus

A base de dados utilizada foi a “Tutorial.idi”, que consiste em um arquivo texto disponível como exemplo pela ferramenta.

As regras são geradas juntamente com os cálculos de suas medidas de interesse, por exemplo, “carrots & onions -> potatoes [Coverage=0.041 (41); Support=0.025 (25); Strength=0.610; Lift=2.15; Leverage=0.0134 (13)]”. Essas informações podem ser exportadas para o excel ou para um arquivo texto.

A principal limitação da ferramenta é a geração de regras com somente um item em cada conseqüente.

7.4 DETALHAMENTO DO WIZRULE®

A ferramenta testada foi a WizRule version 3.00 Demo, 1994-1997, que suporta até 1000 transações.

O produto reconhece arquivos de entrada do tipo *dbf* (dBase, FoxPro, Clipper), *mdb* (Access), *xls* (Excel), dois formatos de arquivos texto e também tabelas via ODBC.

O WizRule minera regras de associação (*se...então...*), regras com fórmulas (descrevendo uma função entre atributos quantitativos), e, também, exceções às regras encontradas.

As regras de associação podem ser da forma:

- Básica: *If the value of X1 is a, then the value of X2 is b.*
- Dia/Mês/Ano: *If the value of date field X1 contains the nth day of the month, then the value of X2 is a.*
- Inicia por …: *If the value of X1 starts with the string c1c2c3 . . . then the value of X2 is b.*

É possível escolher os itens que deverão ser ignorados no antecedente e os itens que deverão ser ignorados no conseqüente da regra. Pode também ser definido se os valores vazios de um determinado atributo devem ou não ser levados em consideração, o padrão é não considerar tais valores. A tela para efetuar estas configurações é a exibida pela figura 7.2. O arquivo *Demo1.dbf*, que é um exemplo de base de dados da ferramenta, foi o utilizado.

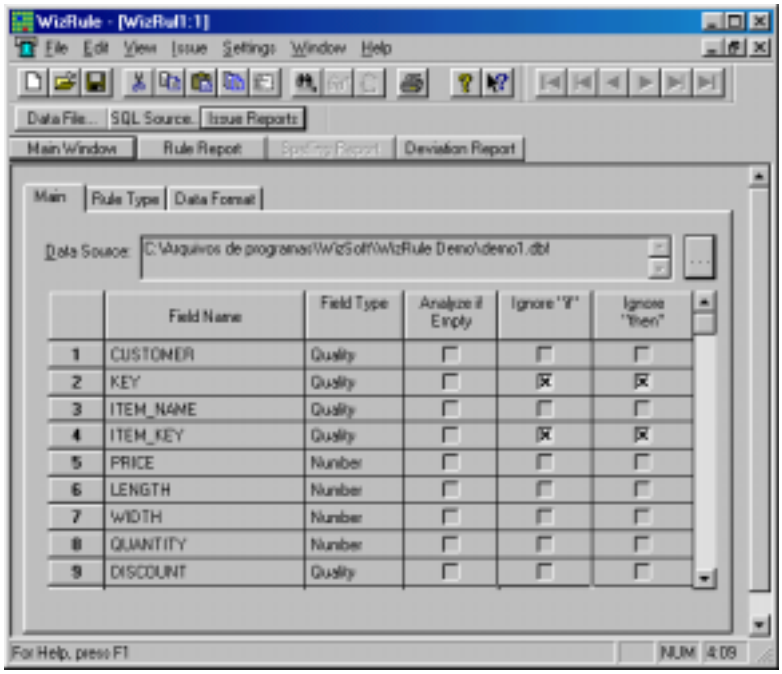


Figura 1.2 – Primeira tela de configuração do WizRule

Para a mineração de regras de associação pode ser utilizada a confiança (*Minimum probability*) e o número mínimo de casos do banco de dados onde a regra deve ser válida (*Minimum number of cases of a rule*).

As regras de associação geradas podem ser exportadas para um arquivo texto. Como exemplo de regra gerada pelo WizRule, temos:

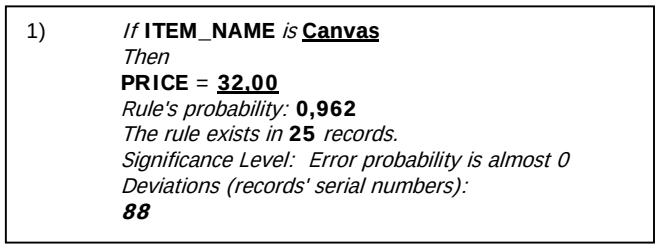


Figura 1.3 – Regra gerada pelo WizRule

A principal limitação do WizRule é a pequena quantidade de parâmetros e de medidas de interesse associadas à mineração de regras de associação, sendo, portanto, apresentado ao usuário final um elevado número de regras.

APÊNDICE II. DESCRIÇÃO DAS BASES DE DADOS UTILIZADAS

Este apêndice lista, para a base de dados de apenados e para a base de dados de processos da VEP/TJRJ, utilizadas para testes dos serviços implementados, os atributos e o seu respectivo domínio.

8.1 APENADOS

A base de dados de apenados, completa, possui informações sobre 111.316 apenados. Após as fases de preparação, seleção, limpeza e transformação, a base de dados utilizada na mineração permaneceu com 35.616 registros. Alguns registros foram retirados da base de dados em virtude da falta de informações importantes e ou por causa de ruídos. Para esta limpeza os serviços de retorno de domínio e de nulos foram utilizados. O principal corte de registros foi feito em virtude da falta de informação estruturada sobre os delitos anteriores ao ano de 2000. Foram utilizados os atributos que constam na Tabela 8-1.

Nome do Atributo	Domínio	Descrição do Atributo
NACIONALIDADE	BRA – brasileiro; EST – estrangeiro;	Nacionalidade do apenado
UF	99 – não consta informação; AC AL, AM, AP, BA, CE, DF, ES, GO, MA, MG, MS, MT, NC, PA, PB, PE, PI, PR, RJ, RN, RO, RR, RS, SC, SE, SP, TO	Naturalidade (Estado da Federação) do apenado
MUNICIPIO	NC – não consta informação, ANGRA DOS REIS, APERIBE, ARARUAMA, AREAL, ARRAIAL DO CABO, BARRA DO PIRAI, BARRA MANSA, BELFORD ROXO, BOM JARDIM, BOM JESUS DO ITABAPOANA, CABO FRIO, CACHOEIRAS DE MACACU, CAMBUCI, CAMPOS DOS GOYTACAZES, CANTAGALO, CARDOSO MOREIRA, CARMO, CASIMIRO DE ABREU, CONCEICAO DE MACABU, CORDEIRO, DUAS BARRAS, DUQUE DE CAXIAS, ENGENHEIRO PAULO DE FRONTIN, GUAPIMIRIM, ITABORAI, ITAGUAI, ITALVA, ITAOCARA, ITAPERUNA, ITATIAIA, JAPERI, LAJE DO MURIAE, MACAE, MAGE, MANGARATIBA, MARICA, MENDES, MIGUEL PEREIRA, MIRACEMA, MUNICIPIO INFORMADO, NATIVIDADE, NILOPOLIS, NITEROI, NOVA FRIBURGO, NOVA IGUACU, PARACAMBI, PARAIBA DO SUL, PARATI, PATY DO ALFERES,	Naturalidade (Município) do apenado

Nome do Atributo	Domínio	Descrição do Atributo
	PETROPOLIS, PIRAI, PORCIUNCULA, QUEIMADOS, QUISSAMA, RESENDE, RIO BONITO, RIO CLARO, RIO DAS FLORES, RIO DAS OSTRAS, RIO DE JANEIRO, SANTA MARIA MADALENA, SANTO ANTONIO DE PADUA, SAO FIDELIS, SAO GONCALO, SAO JOAO DA BARRA, SAO JOAO DE MERITI, SAO JOSE DE UBA, SAO JOSE DO VALE DO RIO PRETO, SAO PEDRO DA ALDEIA, SAO SEBASTIAO DO ALTO, SAPUCAIA, SAQUAREMA, SILVA JARDIM, SUMIDOURO, TANGUA, TERESOPOLIS, TRAJANO DE MORAIS, TRES RIOS, VALENCA, VARRE E SAI, VASSOURAS, VOLTA REDONDA	
NURC	NURC 1 - SEDE RIO DE JANEIRO, NURC 10 - SEDE ITAPERUNA, NURC 11 - SEDE CABO FRIO, NURC 2 - SEDE NITEROI, NURC 3 - SEDE PETROPOLIS, NURC 4 - SEDE DUQUE DE CAXIAS, NURC 5 - SEDE VOLTA REDONDA, NURC 6 - SEDE CAMPOS, NURC 7 - SEDE VASSOURAS, NURC 8 - SEDE ITAGUAI, NURC 9 - SEDE NOVA FRIBURGO	Núcleos Regionais da Corregedoria relativos ao MUNICÍPIO (o NURC que agrupa várias comarcas, que é a divisão de distrito judicial)
IDADE	18_24 – faixa etária entre 18 e 24 anos, 25_30 – faixa etária entre 25 e 30 anos, 31_40 – faixa etária entre 31 e 40 anos, 41_50 – faixa etária entre 41 e 50 anos, 51_... – faixa etária acima de 51 anos	Idade atual do apenado
SEXO	F – feminino, M - masculino	Sexo do apenado
COR	Branca, Parda, Preta, Não Consta informação	Cor do apenado
ESTADO_CIVIL	<NÃO CONSTA>, CASADO, DIVORCIADO, OUTROS, SEPARADO, SEPARADO DE FATO, SOLTEIRO, UNIÃO ESTÁVEL, VIUVO	Estado Civil do apenado
GRAU_ INSTRUCAO	1.GRAU COMPLETO, 1.GRAU INCOMPLETO, 2.GRAU COMPLETO, 2.GRAU INCOMPLETO, <NÃO CONSTA>, ANALFABETO, SUPERIOR, SUPERIOR INCOMPLETO	Grau de instrução do apenado
QT_PROCESSOS	1, 2, 3+ - mais de 3 processos	Quantidade de Processos em execução do apenado
QT_PROCESSOS_TOTAL	1, 2, 3+ - mais de 3 processos	Quantidade de Processos do apenado
REGIME	A – aberto, F – fechado, I – integralmente fechado, S – semi-aberto	Regime de cumprimento da pena
SITUACAO_APENADO	Evadido, Expulso, Falecido, Indultado, Livramento Condicional, PRD, Preso, Restabelecimento, Revogação, Solto, Sursis	Situação relativa ao cumprimento da pena do apenado
UNIDADE_PRISIONAL	BATPM, CAALB, DELEGACIA, DESIPE, INSTITUTO PENAL, MINMA, ORDIV, PATPM, PENITENCIÁRIA, PMERJ, PREMA, PRESÍDIO, OUTROS	Unidade de cumprimento da pena do apenado
PRESO_SOLTO	Evadido, Livramento Condicional, Preso, Solto, Outras Situações	Situação relativa ao cumprimento da pena do apenado (somente situações de prisão e liberdade)

Nome do Atributo	Domínio	Descrição do Atributo
REINCIDENCIA	S – sim, N – não	Indica se o apenado é reincidente ou não
CRIME1 à CRIME8	Associação - tráfico, Contra a Pessoa, Contra a incolumidade pública, Contra o Patrimônio, Contra os Costumes, Corrupção de menores, Furto, Homicídio, Jogo do Bicho, Ocupação ilícita, Outros, Relacionado com arma de fogo, Roubo, Tráfico de entorpecentes, Usuário de drogas	Tipo de crime cometido nos diversos processos

Tabela 1-2 - Tabela com os atributos da base de dados de apenados da VEP

8.2 PROCESSOS

A base de dados de apenados, completa, possui informações sobre 155.484 processos. Após as fases de preparação, seleção, limpeza e transformação, a base de dados utilizada na mineração permaneceu com 42.716 registros. Para a eliminação de registros da base de dados foram utilizados critérios semelhantes ao realizado na base de dados de apenados. Foram utilizados os atributos que constam na Tabela 8-2.

Nome do Atributo	Domínio	Descrição do Atributo
NACIONALIDADE	BRA – brasileiro; EST – estrangeiro;	Nacionalidade do apenado no processo
UF	99 – não consta informação; AC, AL, AM, AP, BA, CE, DF, ES, GO, MA, MG, MS, MT, NC, PA, PB, PE, PI, PR, RJ, RN, RO, RR, RS, SC, SE, SP, TO	Naturalidade (Estado da Federação) do apenado no processo
MUNICIPIO	NC – não consta informação, ANGRA DOS REIS, APERIBE, ARARUAMA, AREAL, ARRAIAL DO CABO, BARRA DO PIRAI, BARRA MANSA, BELFORD ROXO, BOM JARDIM, BOM JESUS DO ITABAPOANA, CABO FRIO, CACHOEIRAS DE MACACU, CAMBUCI, CAMPOS DOS GOYTACAZES, CANTAGALO, CARDOSO MOREIRA, CARMO, CASIMIRO DE ABREU, CONCEICAO DE MACABU, CORDEIRO, DUAS BARRAS, DUQUE DE CAXIAS, ENGENHEIRO PAULO DE FRONTIN, GUAPIMIRIM, ITABORAI, ITAGUAI, ITALVA, ITAOCARA, ITAPERUNA, ITATIAIA, JAPERI, LAJE DO MURIAE, MACAE, MAGE, MANGARATIBA, MARICA, MENDES, MIGUEL PEREIRA, MIRACEMA, MUNICIPIO INFORMADO, NATIVIDADE, NILOPOLIS, NITEROI, NOVA FRIBURGO, NOVA IGUACU, PARACAMBI, PARAIBA DO SUL, PARATI, PATY DO ALFERES, PETROPOLIS, PIRAI, PORCIUNCULA, QUEIMADOS, QUISSAMA, RESENDE, RIO BONITO, RIO CLARO, RIO DAS FLORES, RIO DAS	Naturalidade (Município) do apenado no processo

Nome do Atributo	Domínio	Descrição do Atributo
	OSTRAS, RIO DE JANEIRO, SANTA MARIA MADALENA, SANTO ANTONIO DE PADUA, SAO FIDELIS, SAO GONCALO, SAO JOAO DA BARRA, SAO JOAO DE MERITI, SAO JOSE DE UBA, SAO JOSE DO VALE DO RIO PRETO, SAO PEDRO DA ALDEIA, SAO SEBASTIAO DO ALTO, SAPUCAIA, SAQUAREMA, SILVA JARDIM, SUMIDOURO, TANGUA, TERESOPOLIS, TRAJANO DE MORAIS, TRES RIOS, VALENCA, VARRE E SAI, VASSOURAS, VOLTA REDONDA	
NURC	NURC 1 - SEDE RIO DE JANEIRO, NURC 10 - SEDE ITAPERUNA, NURC 11 - SEDE CABO FRIO, NURC 2 - SEDE NITEROI, NURC 3 - SEDE PETROPOLIS, NURC 4 - SEDE DUQUE DE CAXIAS, NURC 5 - SEDE VOLTA REDONDA, NURC 6 - SEDE CAMPOS, NURC 7 - SEDE VASSOURAS, NURC 8 - SEDE ITAGUAI, NURC 9 - SEDE NOVA FRIBURGO	Núcleos Regionais da Corregedoria relativos ao MUNICÍPIO (o NURC que agrupa várias comarcas, que é a divisão de distrito judicial)
IDADE	18_24 – faixa etária entre 18 e 24 anos, 25_30 – faixa etária entre 25 e 30 anos, 31_40 – faixa etária entre 31 e 40 anos, 41_50 – faixa etária entre 41 e 50 anos, 51_... – faixa etária acima de 51 anos	Idade atual do apenado
IDADE_DELITO	18_24 – faixa etária entre 18 e 24 anos, 25_30 – faixa etária entre 25 e 30 anos, 31_40 – faixa etária entre 31 e 40 anos, 41_50 – faixa etária entre 41 e 50 anos, 51_... – faixa etária acima de 51 anos	Idade do apenado na época do delito
SEXO	F – feminino, M - masculino	Sexo do apenado no processo
COR	Branca, Parda, Preta, Não Consta informação	Cor do apenado no processo
ESTADO_CIVIL	<NÃO CONSTA>, CASADO, DIVORCIADO, OUTROS, SEPARADO, SEPARADO DE FATO, SOLTEIRO, UNIÃO ESTÁVEL, VIUVO	Estado Civil do apenado no processo
GRAU_ INSTRUCAO	1.GRAU COMPLETO, 1.GRAU INCOMPLETO, 2.GRAU COMPLETO, 2.GRAU INCOMPLETO, <NÃO CONSTA>, ANALFABETO, SUPERIOR, SUPERIOR INCOMPLETO	Grau de instrução do apenado no processo
ANO_SENT_PROC	-	Ano da sentença do processo
ANO_DELITO_PROC	-	Ano do delito
ORGAO_INQ_PROC	1 DP PRACA MAUA, 10 DP BOTAFOGO, 100 DP NOVA FRIBURGO, 100 DP PORTO REAL, 101 DP TERESOPOLIS, 102 DP SUMIDOURO, 104 DP DUAS BARRAS, 104 DP SAO JOSE DO RIO PRETO, 105 DP CANTAGALO, 105 DP PETROPOLIS, 106 DP CORDEIRO, 106 DP ITAIPAVA, 107 DP PARAIBA DO SUL, 107 DP SAO SEBASTIAO DO ALTO, 108 DP SANTA MARIA MADALENA, 108 DP TRES RIOS, 109 DP SAPUCAIA, 109 DP TRAJANO DE MORAES, 11 DP JARDIM BOTANICO, 110 DP BOM JARDIM, 110 DP TERESOPOLIS, 111 DP CAMPOS, 111 DP SUMIDOURO, 112 DP CARMO, 113 DP ITAOCARA, 114 DP SANTO ANTONIO DE	Órgão processante da peça inicial do processo

Nome do Atributo	Domínio	Descrição do Atributo
	PADUA, 115 DP MIRACEMA, 117 DP PORCIUNCULA, 118 DP ARARUAMA, 118 DP NATIVIDADE, 119 DP RIO BONITO, 119 DP SAO FIDELIS, 12 DP LEME, 120 DP CAMBUCI, 120 DP SILVA JARDIM, 121 DP CASEMIRO DE ABREU, 121 DP ITAPERUNA, 122 DP BOM JESUS DE ITABAPOANA, 122 DP CONCEICAO DE MACABU, 123 DP MACAE, 123 DP SAO JOAO DA BARRA, 124 DP ARARUAMA, 124 DP SAQUAREMA, 125 DP RIO BONITO, 125 DP SAO PEDRO DE ALDEIA, 126 DO CABO FRIO, 126 DP CACHOEIRAS DE MACACU, 127 DP ARMACAO DE BUZIOS, 127 DP SILVA JARDIM, 128 DP CASEMIRO DE ABREU, 128 DP RIO DAS OSTRAS, 129 DP CONCEICAO DE MACABU, 13 DP COPACABANA, 130 DP MACAE, 131 DP SAQUAREMA, 132 DP SAO PEDRO DE ALDEIA, 133 DP CABO FRIO, 134 DP CAMPOS, 135 DP ITAOCARA, 136 DP SANTO ANTONIO DE PADUA, 137 DP MIRACEMA, 138 DP LAJE DE MURIAE, 139 DP PORCIUNCULA, 14 DP LEBLON, 140 DP NATIVIDADE, 141 DP SAO FIDELIS, 142 DP CAMBUCI, 143 DP ITAPERUNA, 144 DP BOM JESUS DE ITABAPOANA, 145 DP SAO JOAO DA BARRA, 15 DP GAVEA, 151 DP NOVA FRIBURGO, 152 DP DUAS BARRAS, 153 DP CANTAGALO, 154 DP CORDEIRO, 155 DP SAO SEBASTIAO DO ALTO, 156 DP SANTA MARIA MADALENA, 157 DP TRAJANO DE MORAES, 158 DP BOM JARDIM, 159 DP CACHOEIRAS DE MACABU, 16 DP BARRA DA TIJUCA, 165 DP MANGARATIBA, 166 DP ANGRA DOS REIS, 167 DP PARATI, 168 DP RIO CLARO, 17 DP SAO CRISTOVAO, 18 DP PRACA DA BANDEIRA, 19 DP TIJUCA, 2 DP SAUDE, 20 DP GRAJAU, 21 DP BONSUCESSO, 22 DP PENHA, 23 DP MEIER, 24 DP PIEDADE, 25 DP ENGENHO NOVO, 26 DP ENCANTADO, 27 DP VICENTE DE CARVALHO, 28 DP MADUREIRA, 29 DP MAGNO, 3 DP CASTELO, 30 DP MARECHAL HERMES, 31 DP RICARDO DE ALBUQUERQUE, 32 DP JACAREPAGUA, 33 DP REALENGO, 34 DP BANGU, 35 DP CAMPO GRANDE, 36 DP SANTA CRUZ, 37 DP ILHA DO GOVERNADOR, 38 DP IRAJA, 39 DP PAVUNA, 4 DP PRACA DA REPUBLICA, 40 DP HONORIO GURGEL, 41 DP TANQUE DELEGACIA LEGAL, 44 DP INHAUMA, 48 DP SEROPEDICA, 49 DP MANGARATIBA, 5 DP MEN DE SA, 50 DP ITAGUAI, 51 DP PARACAMBI, 52 DP NOVA IGUACU, 53 DP MESQUITA, 54 DP BELFORD ROXO, 55 DP QUEIMADOS, 56 DP COMENDADOR SOARES, 57 DP NILOPOLIS, 58 DP POSSE, 59 DP DUQUE DE CAXIAS, 6 DP CIDADE NOVA, 60 DP CAMPOS ELISEOS, 61 DP XEREM, 62 DP IMBARIE, 64 DP VILAR DOS TELES, 65 DP ARRAIAL DO CABO, 66 DP ITALVA, 67 DP PETROPOLIS, 69 DP MAGE, 7 DP SANTA TEREZA, 70 DP PIABETA, 71 DP ITABORAI, 72 DP SAO GONCALO, 73 DP NEVES, 74 DP ALCANTARA, 75 DP IPIIBA, 76 DP NITEROI, 77 DP SANTA ROSA, 78 DP FONSECA, 79 DP JURUJUBA, 8 DP RIO COMPRIDO,	

Nome do Atributo	Domínio	Descrição do Atributo
	80 DP BARRETO, 81 DP ITAIPU, 82 DP MARICA, 83 DP ANGRA DOS REIS, 84 DP PARATI, 85 DP BARRA DO PIRAI, 86 DP RESENDE, 87 DP BARRA MANSA, 88 DP BARRA DO PIRAI, 88 DP VALENCA, 89 DP RESENDE, 89 DP RIO DAS FLORES, 9 DP FLAMENGO, 90 DP BARRA MANSA, 90 DP PARAIBA DO SUL, 91 DP TRES RIOS, 91 DP VALENCA, 92 DP RIO DAS FLORES, 92 DP SAPUCAIA, 93 DP RIO CLARO, 93 DP VOLTA REDONDA, 94 DP PIRAI, 94 DP VOLTA REDONDA, 95 DP PIRAI, 95 DP VASSOURAS, 96 DP MIGUEL PEREIRA, 96 DP VASSOURAS, 97 DP MENDES, 97 DP MIGUEL PEREIRA, 98 DP ENG. PAULO DE FRONTIN, 98 DP MENDES, 99 DP ENG. PAULO DE FRONTIN, 99 DP ITATIAIA, ACADEPOL - ACADEMIA DE POLICIA, CENTRAL DE INQUERITOS / MP, CORREGEDORIA DE POLICIA, D.A.S. - DIVISAO ANTI-SEQUESTRO, D.I.E. - DIVISAO DE INVESTIGACOES, DAIRJ - DELEGACIA DO AEROPORTO, DDV - DELEGACIA DE DEFESA DA VIDA, DEAM - DP ESPECIAL ATENDIMENTO A MULHER, DEAM-DEL.ESP.AT.MULHER-DUQUE DE CAXIAS, DEAT - DELEGACIA ATENDIMENTO AO TURISTA, DECON - DP CRIMES CONTRA O CONSUMIDOR, DEL. DE HOMICIDIOS DA BAIXADA FLUMINENSE, DEL. ESP. ATEND. À 3ª IDADE, DELDIA/DPF/RJ, DELDIA/SR/DPF/RJ, DELDIA/SR/RJ, DELEGACIA DA FAZENDA, DELEGACIA DE COSTUMES E DIVERSOES, DELEGACIA DE DEFRAUDACOES, DELEGACIA DE ECONOMIA POPULAR, DELEGACIA DE FURTO DE AUTOMOVEIS, DELEGACIA DE HOMICIDIOS, DELEGACIA DE JOGOS E DIVERSOES, DELEGACIA DE POLICIA FEDERAL, DELEGACIA DE ROUBOS E FURTOS, DELEGACIA DE TOXICOS, DELEGACIA DE TRANSITO, DELEGACIA DE VIGILANCIA, DELEGACIA DE VIGILANCIA - CENTRO, DELEGACIA DE VIGILANCIA - NORTE, DELEGACIA DE VIGILANCIA - SUL, DELEGACIA EXTRAORDINARIA DE POLICIA, DELEGACIA POLICIAL DE VILA ISABEL, DELEMAF/SR/DPF/RJ, DELEPREN/SR/DPS/RJ, DEPART. FEDERAL DE SEGURANCA PUBLICA, DGPE DEL.MOVEL DE MEIO AMBIENTE, DIG - DIV. DE INFORMACOES GERAIS, DIV. FISCALIZACAO ARMAS E EXPLOSIVOS, DIV.CAPTURA E POLICIA INTERESTADUAL, DIV.CONTROLE DE DIVERSOES PUBLICAS, DIVISAO DE ROUBOS E FURTOS, DOPS - DP ORDEM POLITICA E SOCIAL, DP CRIMES CONTRA A FAZENDA PUBLICA, DP CRIMES CONTRA A SAUDE PUBLICA, DP DE GUARUS, DP DE OUTROS ESTADOS, DP DEFRAUDACOES E FALSIFICACOES, DP MARITIMA AEREA E ESTRANGEIROS, DP MARITIMA AEREA E FRONTEIRAS, DP POLITICA E SOCIAL, DP ROUBOS E FALSIFICACOES, DP ROUBOS E FURTOS DE AUTOMOVEIS, DP SEGURANCA E PROTECAO AO MENOR, DP VIGILANCIA E CAPTURAS - NITEROI, DP VIGILANCIA E CAPTURAS - RIO OESTE, DPCA - NITEROI,	

Nome do Atributo	Domínio	Descrição do Atributo
	DPCA-DIV.PROTECAO CRIANCA E ADOLESCENTE, DPE - DPTO. POLICIA ESPECIALIZADA, DPF - DELEGACIA FAZENDARIA SR RJ, DPF - DELEGACIA REPRESSAO ENTORPECENTES, DPF - DIV. POLICIA FEDERAL - N.IGUAÇU/RJ, DPF - DIVISAO DE POLICIA FEDERAL - CAMPO, DPF - DIVISAO POLICIA FEDERAL - NITEROI, DPF - DOPS SR RJ, DPF - DPRCD SR RJ, DPF - SPMAF SR RJ, DPFAZ/SR/DPF/RJ, DPFAZ/SR/RJ, DPI - DPTO. POLICIA DO INTERIOR, DPM - DPTO. DE POLICIA METROPOLITANA, DPPS - DP. POLICIA POLITICA E SOCIAL, DPRCE/SR/DPF/RJ, DPRE/DPF/RJ, DPRE/SR/DPF/RJ, DRACO-IE/REPR.AÇÕES CRIM.ORG.- INQ. ESP., DRAE-DEL. REPRES. A ARMAS E EXPLOSIVOS, DRCCP - DIV.REP. CRIME CONTRA PATRIMONIO, DRE - DIVISAO REPRESSAO ENTORPECENTES, DRE/SR/DPF/RJ, DRF - DELEGACIA DE ROUBOS E FURTOS, DRFC - DP ROUBOS E FURTOS DE CARGAS, DRFVAT - DP ROUBO E FURTO AUTO TERRESTRE, DRRF-CEF DP CONTRA ESTABEL. FINANCEIROS, DVC - DELEG.VIGIL.CAP. NOVA IGUACU, METROPOL XII, MINISTERIO PUBLICO, POLICIA MILITAR ESTADO DO RIO DE JANEIRO, POLINTER, PROJECAO DE DEFRAUDACOES DO INTERIOR, PROJECAO DE ENTORPECENTES DO INTERIOR, PROJECAO DE HOMICIDIOS DO INTERIOR	
SIGLA_INQ_PROC	99 – não consta informação; AC, AL, AM, AP, BA, CE, DF, ES, GO, MA, MG, MS, MT, NC, PA, PB, PE, PI, PR, RJ, RN, RO, RR, RS, SC, SE, SP, TO	UF do Órgão processante da peça inicial do processo
PECA_INI_PROC	PORTARIA, FLAGRANTE, INQUERITO, I.P.M., OUTROS, <NÃO CONSTA>, REGISTRO DE OCORRÊNCIA	Peça inicial do processo
PENA	Medida de Segurança, Multa, Pena Privativa de Liberdade, Pena Restritiva de Direito, Sursis	Tipo de pena aplicada
TEMPO_DELITO_SENT	-	Tempo decorrido entre o delito e a sentença
REINCIDENCIA	S – sim, N – não	Indica se o apenado no processo é reincidente ou não
REGIAO	Não Identificada, ILHA, PAN, LEOPOLDINA, OUTRO ESTADO, JACAREPAGUA, TIJUCA, BANGU, IRAJA, OESTE, NORTE, MEIER, SUL, ESPECIALIZADA, CENTRO, INTERIOR, METROPOLITANA	Região do Órgão processante da peça inicial do processo
DIA_SEMANA	Não Identificado, DOMINGO, SEGUNDA, TERÇA, QUARTA, QUINTA, SEXTA, SÁBADO	Dia da semana do delito
MÊS	JANEIRO, FEVEREIRO, MARÇO, ABRIL, MAIO , JUNHO, JULHO, AGOSTO, SETEMBRO, OUTUBRO , NOVEMBRO, DEZEMBRO	Mês da semana do delito
STATUS	Transferido, Terminado, Extinto, Em Execução	Situação do processo
CRIME1 à CRIME8	Associação - tráfico, Contra a Pessoa, Contra a incolumidade pública, Contra o Patrimônio, Contra os Costumes, Corrupção de menores, Furto, Homicídio, Jogo do Bicho, Ocupação ilícita, Outros, Relacionado com arma de fogo, Roubo, Tráfico de	Tipo de crime cometido nos diversos processos

Nome do Atributo	Domínio	Descrição do Atributo
	entorpecentes, Usuário de drogas	

Tabela 1-3 – Tabela com os atributos da base de dados de processos da VEP

REFERÊNCIAS BIBLIOGRÁFICAS

[AGR93.1] Agrawal, R.; Imielinski, T.; Swami, A. **Mining Association Rules between Sets of Items in Large Databases**. Proceedings of the ACM SIGMOD Conference. Washington, USA, May 1993.

[AGR93.2] Agrawal, R.; Imielinski, T.; Swami, A. **Database Mining: A performance perspective**. IEEE Trans. On Knowledge and Data Engineering. December, 1993.

[AGR94] Agrawal, R.; Srikant, R. **Fast Algorithms for Mining Association Rules**. Proc. Int´l Conf. Very Large Data Bases, pp. 487-499, Santiago, Chile, Sept. 1994.

[AGR96] Agrawal, R.; Mannila, H.; Srikant, R.; Toivonen, H.; Verkamo, A. Inkeri. **Fast Discovery of association Rules**. In Advances in Knowledge Discovery and Data Mining, Fayyad et al. (Eds.). AAAI Press/The MIT Presss, 1996.

[AGR96.2] - Agrawal, R.; Shafer, J. C. **Parallel Mining of Association Rules**. IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 6, Dez, 1996.

[AGR00] Agrawal, R.; Aggarwal, C.; Prasad, V.V.V. **Depth First Generation of Long Patterns**. In 7th Int'l Conference on Knowledge Discovery and Data Mining, 2000.

[AYA99] Ayan, N.; Tansel, A.; Arkun, E. **An Efficient Algorithm to Update Large Itemsets with Early Pruning**. In Proc. of the 5th ACM Intl. Conf. On Knowledge Discovery and Data Mining, 1999.

[AUM99] Aumann, Y.; Lindell, Y. **A Statistical Theory for Quantitative Association Rules**. Int'l Conference on Knowledge Discovery and Data Mining, 1999.

[BAY98] Bayardo, R. J. **Efficiently mining long patterns from databases**. In ACM SIGMOD Conf. of Management of Data, 1998.

[BAY99] Bayardo Jr., R. J.; Agrawal, R. **Mining the Most Interesting Rules**. In: ACM SIGKDD INT´L CONF. ON KNOWLEDGE DISCOVERY AND DATA MINING, 5, 1999, San Diego. Proceedings... New York: ACM Press, 1999, p. 145-154.

[BIG96] Bigues, J. P. - **Data mining with Neural Networks**, 1996. McGrawHill.

[BMS03] **KDD Cup 2000**. Disponível em: www.ecn.purdue.edu/KDDCUP/data/BMS-WebView-1.dat.gz. Acessado em 04/10/2003.

[BRIN97] Brin, S.; Motwani, R.; Ullman, J. - **Dynamic Itemset Counting and Implication Rules for Market Basket Data**. In Proc. of the ACM Int'l Conf. on Management of Data, 1997.

[BRU00] Brusso, M. J. - **Access Miner: uma proposta para a extração de regras de associação aplicada à mineração do uso da Web**. Dissertação de Mestrado, Instituto de Informática, UFRGS, 2000.

[BUR01] Burdick, D.; Calimlim, M.; Gehrke, J. - **MAFIA: a maximal frequent itemset algorithm for transactional databases**. In Intl. Conf. on Data Engineering, Apr. 2001.

[CAM00] Camargo, S.S. - **Mineração de Regras de Associação no Problema de Cesta de Compras aplicada ao Comércio Varejista de Confecção**. UFRGS, Instituto de Informática, VI Semana Acadêmica do PPGC, 2000. Disponível em: www.inf.ufrgs.br/pos/SemanaAcademica/Semana2000/SandroCamargo.

[CHE97] Cheung, D. W.; Lee, S. - **A General Incremental Technique for Maintaining Discovery Association Rules**. In Proc. of the 5th Intl. Conf. On Databases Systems for Advanced Applications, 1997.

[CHE96] Cheung, D.W.; Han, J.; Ng, V.; Wong, C.Y. - **Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique**. Proc. int'l Conf. Data Eng., New Orleans, La., Feb. 1996.

[CHE96.2] Cheung, D.W. - **Efficient Mining of Association Rules in Distributed Databases**. IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No 6, 1996.

[CHEN96] Chen, M.; Han, J.; Yu, P.S. - **Data Mining: An Overview from a Database Perspective**. IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No 6, December 1996.

[DOM98] Domingo, C.; Gavaldà, R.; Watanabe, O. - **On-line Sampling Methods for Discovering Association Rules**. Universitat Politècnica de Catalunya, Departament de Llenguatges i Sistemes Informàtics, Report Number: LSI-99-4-R, 1998.

[FAY96] Fayyad, U.; Shapiro, G.; Smyth, P. - **From Data Mining to Knowledge Discovery. in Advances in Knowledge Discovery and Data Mining**. Fayyad et al. (Eds.). AAAI Press/The MIT Presss, 1996.

[FAY96.2] Fayyad, U.; Shapiro, G.; Smyth, P. - **The KDD process for Extracting Useful Knowledge from Volumes of Data**. Communications of the ACM (Special Issue on Data mining). November, 1996.

[FEL99] Feldens, M. A.; Citolin, I. M.; Frigeri, S. R. - **Metodologias para a implementação da inteligência do negócio: desenvolvimento de sistema de informação para Database Marketing**. Caxias do Sul: Revista do CCET, 1999.

[FUK96] Fukuda, F.; Morimoto, Y.; Morishita, S. - **Data Mining using two-dimensional optimazed association rules: scheme, algorithms and visualization**. Proc. ACM SIGMOD Int'l Conf. on Management of Data, 13-23, 1996.

[GOE99] Goebel, M.; Gruenwal, L. - **A survey of data mining and knowledge discovery software tools**. ACM SIGKDD, 1999.

[GON01] Gonçalves, L.P.F. - **Avaliação de Ferramentas de Mineração de Dados como fonte de dados relevantes para tomada de decisão: aplicação na rede União de supermercados**. Dissertação de Mestrado, Escola de Administração, UFRGS, 2001.

[GOU01] Gouda, K.; Zaki, M. - **Efficiently mining maximal frequent itemsets**. In Proc. of the 1st IEEE Int'l Conference on Management of Data, 2001.

[GUN97] Gunolulos, D.; Mannila, H.; Saluja,S. - **Discovering all the most specific sentences by randomized algorithms**. In Int'l Conf. on Database Theory, 1997.
[HAN97] Han, E.-H.; Karypis, G.; Kumar, V. - **Scalable parallel data mining for association rules**. In Proc of 1997 ACM/SIGMOD Int'l Conf. On Management of Data, Arizona, 1997.

[HAN00] Han, Mao, R. - **Closet: An effecient algorithm for mining frequent closed itemsets**. In SIGMOD Int'l Workshop on Data Mining and Knowledge Discovery, 2000.

[HANPEI00] Han, J. P.; Yin, Y. - **Mining Frequent Patterns without candidate generation**. In Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, 2000.

[HOU93] Houtsma, M.; Swami, A. - **Set-oriented Mining of Association Rules**. Research Report RJ 9567, IBM Almaden Ressearch Center, San Jose, California, 1993.

[INF96] **Garimpando Dados**, Informação, Outubro 1996, Ano XIV, N. 76.

[JOR03] Jorge, A. M. – **Material de Apoio do Mestrado em Análise de Dados e Sistema de Apoio à Decisão**. Faculdade de Economia, Universidade do Porto. Disponível em: www.niaad.liacc.up.pt/~amjorge/Aulas/ madsad/ecd2. Acessado em 21/01/2003.

[KDN03] **Kdnuggets**. Disponível em: www.kdnuggets.com. Acessado em 12/05/2003.

[KDDCUP03] **KDD Cup 2000**. Disponível em: www.ecn.purdue.edu/KDDCUP/ . Acessado em 29/09/2003.

[LEE97] Lee, S. D.; Cheung, D. - **Maintenance of Association Rules: When to Update?** In proc. of SIGMOD WorkShop on Research Issues in Data Mining and Knowledge Discovery, 1997.

[LEV99] Levy, E. - **The Lowndown on Data Mining**. Teradatareview, Summer, 1999.

[LIN98] Lin, D-I.; Dunham, M. H. - **Pincer-search: A new algorithm for discovering de maximum frequent set**. In 6th Int'l Conf. Extending Database Technology, 1998.

[LIU99] Liu, B.; Hsu, W.; Ma, Y. - **Mining Association Rules with Mulitiple Minimum Supports**. Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-99), pp. 337-341, 1999.

[LOP99] Lopes, C.H.P. - **Classificação de registros em banco de dados por evolução de regras de associação utilizando algoritmos genéticos**. Dissertação de Mestrado, Dept. de Engenharia Elétrica, PUC/RJ, 1999.

[MAG03] **Magnum Opus**. Disponível em: www.giwebb.com. Acessado em 15/03/2003.

[MAH00] Mahesh V. J; Han, E.-H.; Karypis, G.; Kumar, V. - **Efficient parallel algorithms for mining associations**. In M. Zaki and C. T Ho (eds), Large-Scale Parallel Data Mining, LNAI State-of-the_Art Survey, Volume 1759, Springer-Verlag, 2000.

[MAN94] Mannila, H.; Toivonen, H.; Verkamo, A.I. - **Efficiente Algorithms for Discovering Association Rules**. AAAI Workshop on Knowledge Discovery in Databases, Eds. Usama M. Fayyad ans Ramasamy Uthurusamy, pages 181-192, Seattle, Washington, July 1994.

[MAT93] Matheus, C.J.; Chan, P. K.; Shapiro, G. P. - **Systems for knowledge Discovery in Databases**. IEEE Trans. On Knowledge and Data Engineering. December, 1993.

[MED99] Medeiros, C. M. B. – **Exemplo simples de algoritmo de clustering**. XVI JAI Jornada de Atualização Científica. XVII Congresso da SBC. NCE – UFRJ Congresso da Sociedade Brasileira de Computação-SBC'99, Rio de Janeiro.

[MES02] Mesbah, A. - **Data Mining and Parallel/Distributed Processing**. 2002. Disponível em: elektron.its.tudelft.nl/~amesbah/docs/datamining.pdf. Acessado em 13/03/2003.

[MOR98] Moraes, R. L. - **Sistemas de Data Warehouse: Estudo e Aplicação na Área da Saúde** – UFRGS – Ago/1998. Disponível em : www.marketingdeprecisao.com.br/artigos.asp?assunto=1.

[PAR97] Park, J.S.; Chen, M.-S. - **Using a Hash-Based Method with Transaction Trimming for Mining Association rules**. IEEE Transactions on Knowledge and Data Engineering, Vol. 9, No 5, 1997.

[PAR95] Park, J.S.; Chen, M.-S.; Yu, P.S. - **Efficient Parallel Data Mining for Association Rules**. Proc. of the 4th Conf. on Information and Knowledge Management, pages 31--36, November 1995.

[PAS99] Pasquier,N.; Bastide,Y.;. Taouil,R.; Lakhal,L. - **Discovering frequent closed itemsets for association rules**. In Proc of 7th Int'l Conf. on Database Theory (ICDT), Verlag, 1999.

[RIO03] **Mapa das Subprefeituras da Cidade do Rio de Janeiro** . Disponível em : www.rio.rj.gov.br/riourbe/mapaos.htm. Acessado em 22/10/2003.

[ROD03] Rodrigues, A. Q. - **Limpeza de Dados em Ambientes de Data Warehouse: Formalização e Técnicas de Implementação**. Dissertação de Mestrado, UFRJ/IM/NCE, 2003.

[SAV95] Savasere, A.; Omielinski, E.; Nvathe, S. - **An Efficient Algorithm for Mining Association Rules in Large Databases**. In Very Large Databases, 1995.

[SIE03] Siebes, A. - **Association Rules**. Institute of Information and Computing Sciences, Department of Mathematics and Computer Science, Universiteit Utrecht. Disponível em www.cs.uu.nl/docs/vakken/dm/assoc-complete.pdf. Acessado em 12/02/2003.

[SIL00] Silva, D. R. - **Análise e triagem de padrões em processos de Descoberta de Conhecimento em Bases de Dados**. Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação, PUC/RS, 2000.

[SRI96] Srikant, R.; Agrawal, R. - **Mining Quantitative Association Rules in Large Relational Databases**. In Procedings of ACM SIGMOD Conference on Management of Data (SIGMOD'96), pages 1-12, Montreal, Canada, 1996.

[SRI97] Srikant, R.; Vu, Q.; Agrawal, R. - **Mining Association Rules with Item Constraints**. IBM Almaden Research Center & American Association for Artificial Intelligence, 1997.

[TAL02] Talia, D. - **High Performance Data Mining and Knowledge Discovery**. Euro-Par'2002 - Parallel Processing. 5th International Euro-Par Conference, Paderborn, Germany, 2002.

[THO00] Thomas, S.; Chakravarthy, S. - **Incremental Mining of Constrained Associations**. In Proc. of the 7th Int'l Conf. of High Performance Computing (HiPC), p. 547-558, 2000.

[TOI96] Toiniven, H. - **Sampling Large Databases for Association Rules**. Proceedings of the 22th VLDB Conference, 1996.

[VAL01] Castelo, R.; Feelders, A.; Siebes, A. - **Mambo: Discovering Association Rules Based on Conditional Independencies**. In Lecture Notes in Computer Science, vol. 2189, pages 289-298. 2001.

[VEL01] Veloso, A.; Possas, B.; Meira Jr, W.; Carvalho, M.B. - **Knowledge Management in Association Rule Mining**. In Proc. of the 1st IEEE Intl WorkShop on Integrating Data Mining and Knowledge Management, San Jose, USA, 2001.

[VEL02] Veloso, A.; Meira Jr, W.; Carvalho, M. B. - **Mining Reliable Models of Associations in Dynamic Databases**. XVII Simpósio Brasileiro de Banco de Dados, Gramado-RS, Brasil, 2002.

[VEP03] **Vara de Execuções Penais do TJRJ**. Disponível em: www.tj.rj.gov.br/instituc/1instancia/vep2/framevep.htm. Acessado em 24/07/2003.

[ZAK97] Zaki, M. J.; Parthasarathy, S.; Li, W. - **New Algorithms for Fast Discovery of Association Rules**. In Proc. of the 3th ACM Intl. Conf. On Knowledge Discovery and Data Mining, 1997.

[ZAK97.2] Zaki, M. J.; Parthasarathy, S.; Li, W. - **A Localized Algorithm for Parallel Association Mining**. In 9th ACM Symp. Parallel Algorithms and Architectures. Jun 1997.

[ZAK99] Zaki, M. J. - **Parallel and Distributed association mining**: A survey. IEEE Concurrency, 1999.

[ZAK00] Zaki, M. J. - **Parallel sequence mining on SMP machine**. In Zaki and Ho, 2000.

[ZAK01] Zaki, M.J.; Gouda, K. - **Fast vertical mining using diffsets**. Technical Report 01-1, Department of Computer Science, Rensselaer Polytechnic Institute, 2001.

[ZAK02] Zaki, M.J.; Hsiao, C.-J. - **CHARM: An Efficient Algorithm for Closed Itemset Mining**. Proceedings of the Second SIAM International Conference on Data Mining, USA, 2002.

[ZHA97] Zhang, Z.; Lu, Y.; Zhang, B. - **An Effective Partitioning-Combining Algorithm for Discovering Quantitative Association Rules**. Proc. of the 1st Pacific-Asia Conf. on Knowledge Discovery and Data Mining, 1997.

[ZHE01] Zheng, Z.; Kohavi, R.; Mason, L. - **Real World Performance of Association Rule Algorithms**, KDD 2001.

[ZHE01.2] Zheng, Z.; Kohavi, R.; Mason, L - **Real World Performance of Association Rule Algorithms** (versão extensa: realWorldAssocLongPaper.pdf). Disponível em: [KDDCUP03]).

[ZHE01.3] Zheng, Z.; Kohavi, R.; Mason, L - **Real World Performance of Association Rule Algorithms** (versão de apresentação: realWorldAssocSlides.pdf). Disponível em: [KDDCUP03]).

[WIZ03] **WizSoft**. Disponível em www.wizsoft.com. Acessado em 03/02/2003.