UNIVERSIDADE FEDERAL FLUMINENSE – UFF
INSTITUTO DE COMPUTAÇÃO – IC
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CONHECIMENTO DO MUNDO COMO INSTRUMENTO ENRIQUECEDOR DOS RESULTADOS OBTIDOS NA MINERAÇÃO DE DADOS

INHAÚMA NEVES FERRAZ

NITERÓI
2008

INHAÚMA NEVES FERRAZ

CONHECIMENTO DO MUNDO COMO INSTRUMENTO ENRIQUECEDOR DOS RESULTADOS OBTIDOS NA MINERAÇÃO DE DADOS

Tese apresentada ao Curso de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Doutor.
Área de concentração: Otimização Combinatória e Inteligência Artificial.
Orientadora: Ana Cristina Bicharra Garcia, Ph. D.

NITERÓI
2008

F381 Ferraz, Inhaúma Neves.
Conhecimento do mundo como instrumento enriquecedor dos resultados obtidos na mineração de dados / Inhaúma Neves Ferraz. – Niterói : [s.n.], 2008.
155 f. : il.
Tese (Doutorado em Ciência da Computação) – Universidade Federal Fluminense, Instituto de Computação, 2008.

1.Mineração de dados. 2.Regras de associação. 3.Modelo SemPrune. I.Título.

CDD 005.74

INHAÚMA NEVES FERRAZ

Conhecimento do mundo como instrumento enriquecedor dos resultados obtidos na Mineração de Dados

Tese apresentada ao Curso de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Doutor.
Área de concentração: Otimização Combinatória e Inteligência Artificial.
Orientadora: Ana Cristina Bicharra Garcia, Ph. D.

BANCA EXAMINADORA:

_____

Ana Cristina Bicharra Garcia, Ph. D. – Orientadora
Universidade Federal Fluminense - UFF

_____

Aura Conci, D. Sc.
Universidade Federal Fluminense - UFF

_____

Alexandre Plastino de Carvalho, D. Sc.
Universidade Federal Fluminense - UFF

_____

Luis Miguel Parreira e Correia, Dr.
Faculdade de Ciências – Universidade de Lisboa

_____

Jesus M. de la Garza, Ph. D.
Virginia Polytechnic Institute and State University

NITERÓI
2008

À minha família

AGRADECIMENTOS

À minha família, Lúcia, Fernando, Marcelo, pelo estoicismo de suportar minha ausência e pela permanente preocupação com o cumprimento de meus encargos.

À minha orientadora, Professora Ana Cristina Bicharra Garcia, pela sabedoria, firmeza, objetividade e candura com que sempre me apoiou. Ter como orientadora a professora Cristina é um privilégio ímpar. Sua obsessão com a perfeição e seu nível de exigência chegam a ser, à primeira vista, assustadores. Com o passar do tempo, observa-se o notável efeito de alavancagem que a convivência com a professora Cristina traz para a vida profissional de seus orientandos. Associando atividades didáticas, de pesquisadora e de executiva ela consegue alcançar a excelência acadêmica, sem perder a capacidade de entregar produtos acabados da Engenheira, que nunca deixou de ser. Além da orientação e do exemplo, sou grato pela amizade com que sempre fui distinguido.

À professora Aura Conci, pelo exemplo de dedicação a todos os seus alunos, pelos ensinamentos transmitidos e pela calorosa amizade com que sempre me obsequiou.

Ao professor Alexandre Plastino de Carvalho, pela firme orientação, culminando com o redirecionamento da pesquisa com ênfase no tratamento via pós-processamento.

Ao professor Luís Miguel Parreira e Correia, pela paciência e gentileza com que acompanhou as diversas versões deste trabalho.

Ao professor Jesus M. de la Garza, que teve a especial gentileza de enfrentar a língua portuguesa para avaliar e engrandecer a presente pesquisa.

Aos criadores de programas que permitiram que eu utilizasse e modificasse seus produtos no presente trabalho, destacando:

- Paula Yamada Bürkle
- Ana Cristina Bicharra Garcia
- Rafael Heitor Correia de Melo
- Guilherme Luiz Lepsch Guedes

Aos meus colegas do Departamento de Ciência da Computação, que sempre me ofereceram generoso suporte, mesmo nos momentos em que havia menos de mim para dar em troca.

À generosa equipe do ADDLabs, que me proporcionou apoio e alegria no trabalho diário.

Aos meus alunos, cujo comportamento exemplar facilitou sobremaneira meu trabalho.

RESUMO

As regras de associação constituem uma técnica muito popular da mineração de dados. Apesar de sua simplicidade e eficiência, o método apresenta dois pontos que poderiam ser melhorados usando critérios semânticos. O primeiro deles é a geração de uma grande quantidade de associações inúteis, o que somente prejudica o trabalho dos analistas do negócio, que tentam encontrar oportunidades e/ou explicações para os dados armazenados.

O segundo ponto é o fato de muita informação presente nos repositórios de informação poder passar despercebida, pela incapacidade dos métodos de mineração captarem os relacionamentos entre os dados armazenados. Isto obriga os analistas, que buscam interpretar/avaliar os resultados da mineração, a incorporar o conhecimento do domínio em seus modelos mentais que dependem de suas experiências.

Para minimizar o excesso de regras mineradas, já são utilizadas técnicas de pós-processamento, que procuram podar os resultados da mineração, eliminando regras consideradas menos importantes. Estas técnicas baseiam-se em critérios sintáticos, que utilizam a freqüência de ocorrência de itens e a estrutura das regras, para determinar o ponto de corte.

A presente pesquisa utiliza a semântica do conhecimento de mundo, existente nas ontologias, para explorar o relacionamento entre os atributos do domínio. Assim, consegue enriquecer, em conteúdo, o conjunto de Regras de Associação mineradas e, ao mesmo tempo, reduzir a cardinalidade desse mesmo conjunto, aumentando a precisão, quando aplicado no pós-processamento, e aumentando o "recall", quando aplicado no pré-processamento.

O modelo SemPrune desenvolvido executa poda semântica no pós-processamento das Regras de Associação mineradas, e o enriquecimento semântico dos resultados no pós-processamento ou no pré-processamento, de acordo com as condições da Base de Transações analisada. Foram analisados domínios com relacionamentos dos tipos "é-um" e "parte-de", cardinalidades 1:n e m:n, desenvolvendo-se algoritmos para cada caso.

Os resultados obtidos, cuja avaliação foi feita por meio de experimentos realizados sobre Bases de Dados públicas mostraram-se satisfatórios.

Palavras-chave: Mineração de Dados. Regras de Associação. Modelo SemPrune.

ABSTRACT

Association Rules are a very popular data mining technique. Despite its simplicity and efficiency, two points in this method could be improved using semantic criteria. The first one is the generation of a large amount of unnecessary associations that only affect the work of business analysts who try to find opportunities and/or explanations for the stored data.

The second point is that a lot of information in the repositories may go unnoticed by the mining methods' inability to capture the relationships between the stored data. This forces the analysts that try to interpret and evaluate the mining results to incorporate the knowledge of the world into their mental models (which depend on their experiences).

The problem of mining large quantities of rules can be mitigated using post-processing techniques by pruning the results, thus preferably eliminating useless results. Most of these techniques propose syntactic metrics to guide the pruning of the Association Rules output using items frequency and rules' structure in determining cut-off points. This work describes a study on the use of domain ontology as a tool to enrich the mining results of the Association Rules, furthermore serving to reduce the number of association rules that are generated increasing the precision when applied on post-processing and increasing the recall when applied on pre-processing.

In SemPrune, the developed model, semantic pruning is executed in the post-processing of the mined Association Rules and semantic enrichment of the results in post-processing or pre-processing, according to the Transactions Bases data. The analysis was made on domains having "is-a" and "part-of" relationships types, with 1:n and m:n cardinalities. Algorithms have been developed in each case.

The results were evaluated by means of experiments conducted on public Databases and the results were quite satisfactory.

Keywords: Data Mining. Association Rules. SemPrune Model.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

As regras de associação constituem uma técnica muito popular da mineração dos dados (DM), que revela a correlação entre conjuntos de itens em uma série de dados ou transações. Sendo X e Y conjuntos disjuntos de itens de dados, uma regra de associação é uma regra da forma $X \Rightarrow Y$, na qual o conjunto X é chamado de **antecedente**, ou de LHS (*Left Hand Side*), e o conjunto Y de **conseqüente**, ou de RHS (*Right Hand Side*). A regra de associação significa que se todos os itens de X forem encontrados em uma transação, é provável que esta transação também contenha os itens de Y. Apesar de sua simplicidade e eficiência, o método gera ainda uma grande quantidade de associações inúteis, o que somente prejudica o trabalho dos analistas do negócio, que tentam encontrar oportunidades e/ou explicações para os dados armazenados. Para minimizar o excesso de regras, são utilizadas técnicas de pós-processamento, que procuram podar os resultados da mineração eliminando regras consideradas menos importantes. Estas técnicas baseiam-se em critérios sintáticos, que utilizam a freqüência de ocorrência de itens e a estrutura das regras, para determinar o ponto de corte.

Outro problema existente na mineração de regras de associação é que muita informação presente nos repositórios de informação pode passar despercebida, pela incapacidade de os métodos de mineração captarem os relacionamentos entre os dados armazenados. Isto obriga os analistas, que buscam interpretar/avaliar os resultados da mineração, a incorporar o conhecimento do domínio em seus modelos mentais, que dependem de suas experiências.

A presente pesquisa utiliza a semântica do conhecimento de mundo, existente nas ontologias, para explorar o relacionamento entre os atributos do domínio. Com isto, consegue enriquecer, em conteúdo, o conjunto de regras de associação mineradas e, ao mesmo tempo, reduzir a cardinalidade desse mesmo conjunto, aumentando a precisão, quando aplicado no pós-processamento e aumentando o "recall", quando aplicado no pré-processamento.

O modelo SemPrune desenvolvido executa poda semântica no pós-processamento das regras de associação mineradas, e o enriquecimento semântico

dos resultados no pós-processamento das regras, sempre que possível. Nas situações nas quais o pequeno suporte de itens individuais provocaria perda de informação, o modelo SemPrune faz o enriquecimento semântico no pré-processamento.

Diversos tipos e cardinalidades de relacionamentos, entre atributos de domínio, foram estudados e desenvolvidos algoritmos para cada caso.

Os resultados obtidos, cuja validação foi feita por meio de experimentos realizados sobre bases de dados públicas, mostraram-se satisfatórios.

Este trabalho vai mostrar que:

- a poda semântica só tem sentido se for aplicada após um enriquecimento semântico das regras de associação;
- os métodos sintáticos não conseguem sequer perceber o enriquecimento semântico e, quando podam, podem retirar o que não deveria ser podado e deixar de tirar o que poderia ser podado;
- os métodos semânticos, quando aplicáveis, podem reduzir substancialmente o número de regras de associação mineradas.

1.1 PROBLEMA

O problema que este trabalho de pesquisa evidencia é a dificuldade de interpretação da quantidade de regras de associação gerada na mineração de dados, aliada à incapacidade da mineração destas regras aproveitar o que o conhecimento de mundo pode trazer à mineração.

A esmagadora maioria das regras geradas tem pouca ou nenhuma serventia e, contudo, age de forma a dispersar a atenção do pesquisador e/ou usuário [TOI95]. O problema é bem conhecido, e tem sido exaustivamente estudado [JAG08] [NAT05] [BAR05] [XU004] [ABL03] [JAR02]. A maioria dos estudos que buscam redução do número de regras de associação a considerar baseia-se em análises sintáticas e em medidores de interesse das regras [JAG08] [NAT05] [BAR05] [JAR02].

Os analistas que utilizam as regras interpretam e/ou avaliam os resultados, utilizando modelos mentais, baseados na sua experiência (conhecimento de mundo), sobre os dados brutos. Quando uma regra apresenta como valor de um determinado item, USD 27,000, o analista tem de saber se este valor é baixo ou elevado demais. Se com o uso do conhecimento de mundo se tivesse substituído os valores numéricos desse item por faixas rotuladas, o processo seria mais eficiente.

1.2 O QUE TEM SIDO FEITO UTILIZANDO MÉTODOS SINTÁTICOS

[JAR02] faz a poda das regras de associação, utilizando o princípio da entropia máxima, e seus resultados mostram que esta técnica gera pequenos conjuntos de regras com elevado interesse. [BAR05] estuda a mineração de regras não deriváveis, caracterizadas pelo fato de sua confiança poder ser obtida de suas sub-regras (aquelas cujos antecedentes e conseqüentes sejam subconjuntos do antecedente e do conseqüente, respectivamente, da super-regra original). O interessante deste trabalho é caracterizar a poda de regras em abordagens subjetivas e objetivas. O primeiro tipo de abordagem utiliza os "templates" e as restrições. O segundo submete as regras de associação mineradas a métricas genéricas, não dependentes do usuário. Muito embora o interesse de regras seja dependente do usuário, as medidas objetivas não o são. Os autores separam também as abordagens objetivas, aquelas que avaliam cada regra independente, das demais (usando, suporte, confiança, lift, novidade, etc.) e aquelas que avaliam a redundância de uma regra, em função da condição mais geral ou mais específica que as demais. Este tipo de tratamento coincide com o da presente pesquisa. Todavia, os autores utilizam técnicas sintáticas e não semânticas. [JAG08] apresenta um processo de poda de regras de associação que visa recuperar regras que tenham sido podadas açodadamente, pela utilização exclusiva de processos que usam medidas objetivas de interesse das regras.

1.3 O QUE TEM SIDO FEITO UTILIZANDO MÉTODOS SEMÂNTICOS

O uso de conhecimentos semânticos na busca de novos conhecimentos tem ocorrido, principalmente, na mineração de textos, que analisa dados não estruturados [XIA05] [YOO06]. A mineração que analisa dados estruturados em bases de dados apresenta poucos trabalhos [CHE08]. A Tabela 1.1, retirada de [YOO06], foi utilizada por esse autor para justificar a preferência.

Tabela 1.1 Diferenças entre Mineração de Dados e Mineração de Textos

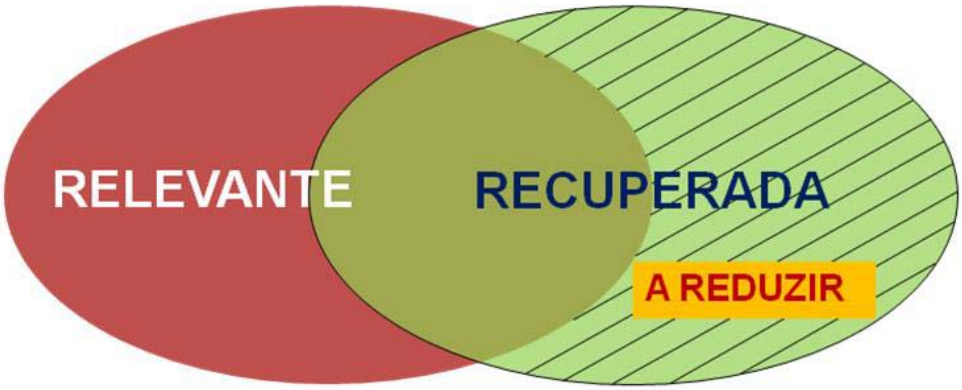
	Objeto da busca	
	Conhecimento novo	Fatos já conhecidos
Dados estruturados	Mineração de dados	Consultas a bancos de dados
Dados não estruturados (texto)	Mineração de textos	Recuperação de informações ou extração de informações

[ABL03] utiliza filtragem estatística, direcionada por uma métrica probabilística, que ajuda a ressaltar as estruturas importantes presentes nos dados. Esta técnica pode ser combinada com algoritmos de máximo, ou algoritmos heurísticos, que busquem encontrar regras relevantes, porém com pequeno suporte. [HSU03] busca o enriquecimento da mineração de regras de associação, incorporando a quantidade de itens movimentados em cada transação, criando bases de dados enriquecidas ("bag databases"). [XUA04] mistura abordagens semânticas e sintáticas. Com as primeiras, descobre e agrupa regras de associação em "clusters" de grupos de regras interessantes, que satisfaçam patamares mínimos sintáticos de suporte, confiança e chi-quadrado, em bases de dados biológicas. O sistema desenvolvido integra inspeção visual dos agrupamentos de dados sobre genes. [SON05] apresenta uma técnica de expansão de consultas que combina regras de associação com ontologias e técnicas de recuperação de informações, para encontrar termos candidatos à melhora do rendimento de recuperação de informações. Aplica semântica e propriedades lingüísticas a corpus de textos não estruturados. [TAO07] faz mineração de textos para documentos, na Web, extraindo, inicialmente, os metadados RDF ("Resource Description Framework") que representam as relações

semânticas, por técnicas de processamento de linguagem natural, gerando uma taxonomia. Esta taxonomia é usada como conhecimento de mundo, para reduzir a informação minerada utilizando processos de generalização. [CHA07] propõe uma nova abordagem ao problema da detecção de fragmentos moleculares freqüentes, recorrendo a algoritmos de descoberta de padrões, em dados tabulares enriquecidos com conhecimento de domínio existente. São criadas moléculas fictícias e, com o conhecimento dos tipos de ligações prováveis são determinados os resultados esperados para cada um dos suportes mínimos testados.

1.4 HIPÓTESE, OBJETO, OBJETIVO E LIMITAÇÕES DA PRESENTE PESQUISA

Para tentar resolver o problema anteriormente descrito, acreditamos poder, por meio da inclusão do conhecimento do mundo expresso em uma ontologia de domínio, melhorar a precisão e o "recall" da mineração. A figura 1.1 ilustra o que se deseja com o enriquecimento do conjunto de regras feito no pós-processamento: manter a informação relevante existente na base de dados de transações, reduzindo o número de regras de associação mineradas (informação recuperada). O conjunto de informações relevantes em uma base de dados independe do processo de recuperação adotado (a priori, usando ontologia do domínio ou qualquer outro). A interseção desse conjunto com o conjunto de informações recuperadas é que depende do processo de recuperação adotado. A poda de regras aumenta a precisão da recuperação, reduzindo a quantidade de informação recuperada.

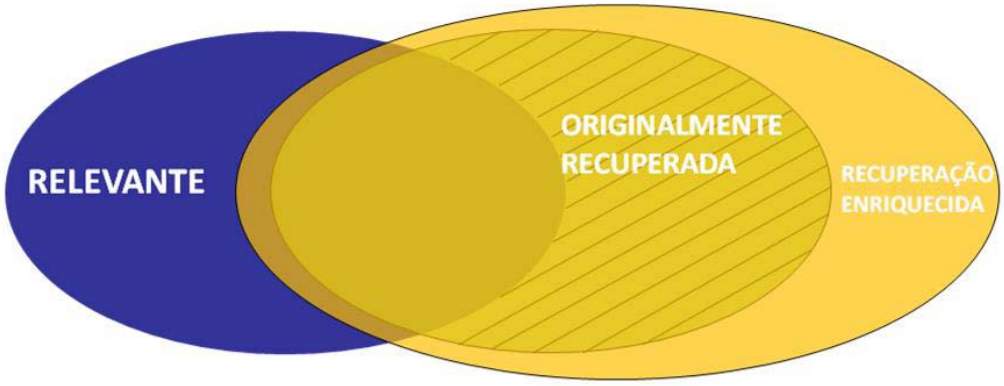


Precisão = |{informação relevante} ∩ {informação recuperada}| / |{informação recuperada}|

Recall = |{informação relevante} ∩ {informação recuperada}| / |{informação relevante}|

Figura 1.1 Precisão e "recall" com enriquecimento no pós-processamento

Como são muitas as relações semânticas presentes em uma ontologia, esta tese concentra-se na inclusão na mineração das relações do tipo "é-um" e "parte-de". Nossa hipótese é a de que o uso desses relacionamentos semânticos no pós-processamento aumenta o "recall", por meio da descoberta de informação oculta, ampliando o número de regras relevantes recuperadas. O enriquecimento do conjunto de regras, quando feito no pós-processamento, como mostrado na figura 1.2, aumenta a precisão, proporcionando substancial ganho ao analista usuário. Isto ocorre pelo incremento das regras relevantes e recuperadas, garantindo o ganho de conteúdo semântico.



Precisão = |{informação relevante} ∩ {informação recuperada}| / |{informação recuperada}|

Recall = |{informação relevante} ∩ {informação recuperada}| / |{informação relevante}|

Figura 1.2 Precisão e "recall" com enriquecimento no pré-processamento

O teste de hipóteses, para a validação deste modelo, considerou hipótese nula e hipótese alternativa:

H0 (hipótese nula)

O modelo semântico não traz nenhum ganho, em precisão ou "recall" de um conjunto de regras de associação mineradas, em relação aos métodos de filtragem sintática.

H1 (hipótese alternativa)

O método proposto enriquece um conjunto de regras de associação. Quando isto ocorre no pós-processamento da mineração aumenta a precisão, por reduzir o número de regras irrelevantes, enquanto mantém o "recall". Quando o enriquecimento ocorre no pré-processamento da mineração aumenta o "recall", por aumentar a informação relevante recuperada (anteriormente oculta), mas a manutenção da precisão depende da taxa de poda obtida, que não pode ser garantida.

Trata-se de verificar se o conhecimento do domínio, presente nas ontologias, pode contribuir, de maneira significativa, para o melhor aproveitamento das regras de associação geradas em um processo de mineração de dados. Este conhecimento pode servir de base para a descoberta de informações não encontradas pela simples mineração de regras, e para a redução "inteligente" do número de regras.

O objeto da pesquisa é a investigação do uso do conhecimento presente nas ontologias, no enriquecimento do conjunto de regras de associação mineradas, bem como na redução da cardinalidade deste conjunto.

O objetivo é estudar a exploração dos relacionamentos entre os dados do domínio e a geração de algoritmos eficazes de enriquecimento do conjunto de regras de associação mineradas com redução da cardinalidade desse conjunto. Isto deve ser feito aumentando a precisão e mantendo o "recall" quando a aplicação ocorrer no pós-processamento, e aumentando o "recall" quando a aplicação ocorrer no pré-processamento.

Uma limitação do trabalho refere-se ao seu escopo. Serão considerados dois casos do uso de ontologias que envolvam tipos específicos de relações:

- as relações de generalização-especialização do tipo "é-um",
- as relações de composição do tipo "parte-de".

Outra limitação é o fato de resultados satisfatórios serem totalmente dependentes da existência de uma ontologia do domínio.

Adicionalmente, a riqueza do conjunto de relacionamentos existentes entre os atributos pode mudar, substancialmente, os resultados esperados, pois os ganhos semânticos serão tão maiores quanto maior for o impacto causado pela consideração do conhecimento de mundo.

1.5 METODOLOGIA DE PESQUISA

Para avaliar nossa hipótese, desenvolvemos uma metodologia, com o objetivo de averiguar a eficiência e a precisão do uso de ontologia de domínio como instrumento, ao mesmo tempo, enriquecedor das regras de associação geradas, e redutor do número de regras, pela eliminação das menos relevantes. A melhora pretendida pelo enriquecimento é qualitativa, e a melhora pretendida pela poda é, essencialmente, quantitativa, medida em número de regras pertinentes. Para tal, foram desenvolvidos algoritmos para os diversos casos particulares enfocados e foi implantado um ambiente de testes. A melhoria qualitativa é subjetiva, pois a mensuração de ganho de "adolescentes são fãs de esportes radicais" sobre "pessoas de 14 anos são fãs de esportes radicais" não pode ser medida objetivamente. O que se observa é que a maioria das inferências é feita sobre agregações e busca de padrões.

Um experimento controlado foi feito para comparação, utilizando-se o ambiente desenvolvido com as técnicas tradicionais e aquelas desenvolvidas no curso do presente trabalho.

1.6 ORGANIZAÇÃO DO TRABALHO

Este trabalho está assim estruturado:

capítulo 2 - estudo sobre mineração de dados;

capítulo 3 - redução da cardinalidade do conjunto de regras de associação mineradas;

capítulo 4 - ontologias e os fundamentos teóricos que se pretende explorar no estudo de técnicas semânticas de tratamento das regras de associação geradas por mineração de dado;

capítulo 5 - apresentação do modelo proposto;

capítulo 6 - descrição dos experimentos realizados;

capítulo 7 - análise dos resultados obtidos;

capítulo 8 - conclusões a que se chegou e aponta pesquisas futuras.

Seguem-se a bibliografia e os apêndices, com detalhamento das funções chamadas pelos dois algoritmos principais do modelo.

2 MINERAÇÃO DE DADOS

Em termos coloquiais, pode-se dizer que: "Mineração de dados ou Data Mining é o processo de varrer grandes bases de dados à procura de padrões como regras de associação, seqüências temporais, para classificação de itens ou agrupamento" (Wikipedia 2008). Nada mais do que uma etapa de um processo conhecido como extração de conhecimento em bases de dados, ou "Knowledge Discovery in Databases" (KDD). Dentre as técnicas de mineração de dados, pode-se citar: regras de associação, redes neurais, análise de agrupamentos, algoritmos genéticos, árvores de decisão e outras.

A chamada mineração de regras de associação permite encontrar, em um tempo relativamente curto, as associações mais freqüentes em uma base de dados. Para tanto, a hipótese subjacente, por exemplo, é que se um conjunto de itens não é freqüente, qualquer combinação de itens que inclua este conjunto também não o será. Ou seja, se a contagem de um determinado conjunto de itens revelar que ele não é freqüente, não será mais necessário contar nenhum dos conjuntos que o incluírem, economizando-se, assim, tempo de processamento. A definição do que seja um item freqüente é função de diversos fatores, objetivos e subjetivos. A competência do analista (usuário) deve balizar esta definição.

2.1 REGRAS DE ASSOCIAÇÃO

Seja $I = \{i_1, i_2, i_3, ..., i_{n-1}, i_n\}$. um conjunto de atributos ou itens. Transação é um conjunto de itens $T=\{t_1, t_2, t_3,..., t_{n-1}, t_n\}$, $T \subset I$. D é um conjunto de transações ou dados relevantes para a tarefa. Uma regra de associação é uma implicação de forma $P \rightarrow Q$ onde $P \subset I$, $Q \subset I$ e $P \cap Q = \varnothing$ [AGR94].

O suporte de uma regra $P \rightarrow Q$ é definido como a probabilidade de que uma transação em D contenha ambos os itens P e Q, $s_D(P \rightarrow Q) = s_D(P \cup Q)$. A confiança de uma regra $P \rightarrow Q$ é definida como a probabilidade de que uma transação em D,

que já contenha os itens em P, também contenha os itens em Q, $c_D(P \rightarrow Q) = s_D(P \cup Q) / s_D(P)$.

Uma regra $P \rightarrow Q$ [s,c] é chamada de regra forte, se dados valores para os parâmetros **suporte mínimo especificado(s)** e **confiança mínima especificada (c)**, então

$s_D(P \rightarrow Q) = s$, $s \geq$ suporte mínimo especificado

$c_D(P \rightarrow Q) = c$, $c \geq$ confiança mínima especificada

Um conjunto de itens freqüentes é um conjunto de itens que têm suporte mínimo. Diz-se que os conjuntos de cardinalidade k são k-itemsets. Se um k-itemset é freqüente, todos os seus (k-i)-itemsets são também freqüentes.

A tarefa de mineração de regras de associação pode ser decomposta em dois passos [ZAK97]:

1 encontrar todos os itemsets freqüentes, ou seja, todas as combinações de itens com suporte maior que o suporte mínimo especificado pelo usuário. Este é o passo mais demorado do processo de busca de regras de associação;

2 gerar, a partir dos itemsets freqüentes, todas as regras com confiança mínima maior que a especificada pelo usuário. Isto se faz, gerando todos os subconjuntos não vazios de cada itemset freqüente x. Para cada subconjunto não vazio, s, de x, calcular a confiança c da regra $s \rightarrow (x-s)$ da forma

$c = sup(\{x,(x-s)\}) / sup(\{s\}) = sup(\{x\}) / sup(\{s\})$. Se $c \geq$ minconf então aceitar a regra $s \rightarrow (x-s)$.

Considerando a notação a seguir, pode-se escrever $A \Rightarrow B$ [s,c] em que s é o suporte da regra e c sua confiança. Suporte é a freqüência de uma regra entre as transações. Um valor elevado de suporte indica que a regra envolve boa parte da base de dados suporte $(A \Rightarrow B [s,c]) = p(A \cup B)$.

Confiança é a freqüência de transações que contém A e B. É uma estimativa da probabilidade condicional.

confiança $(A \Rightarrow B [s,c]) = p(B|A) = sup(A,B)/sup(A)$

Agrawal et al. [AGR93] propuseram um modelo matemático, no qual as regras de associação devem atender aos patamares mínimos de suporte e confiança especificados pelo usuário. A tarefa, recebida como desafio, consiste no desenvolvimento de um modelo híbrido, que atenda a requisitos matemáticos e ontológicos, de geração de regras de associação.

No item 2 apresenta-se a mineração de dados e as regras de associação. No item 3 descrevem-se os métodos adotados para redução, via pós-processamento, da cardinalidade das regras de associação mineradas. O item 4 trata do emprego do conhecimento de mundo para revelar informações que passam despercebidas, sem o uso da semântica, no processo.

2.2 MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO

A sistematização da mineração das regras de associação data de 1994, com o método Apriori (item 2.2.1). Depois deste método, muitos outros foram desenvolvidos e aplicados.

2.2.1 Algoritmo Apriori

O suporte mínimo especificado para um itemset garante a significância estatística da amostra, evitando a consideração de combinações de pequena freqüência. Já a confiança mínima especificada mostra que o resultado obtido não é ocasional, há coesão entre as regras obtidas.

Considera-se forte uma regra que retrata informação interessante, o que é medido pelo fato de a regra atender, simultaneamente, os valores mínimos especificados para suporte e confiança [AGR94]. A natureza de busca de ocorrência de instâncias, em uma base de dados, conduz à coleta de um número extremamente grande de itemsets freqüentes e, em conseqüência, de regras de associação. Isto faz

com que uma parte substancial das regras obtidas seja praticamente inútil, retratando regras óbvias, redundantes e/ou contraditórias.

O algoritmo Apriori [AGR94] busca os itemsets freqüentes, iterativamente baseado no princípio Apriori. Inicialmente, são obtidos os itemsets freqüentes de tamanho 1, a partir daí os de tamanho 2 e assim, sucessivamente.

Princípio Apriori: Qualquer subconjunto de um itemset freqüente tem de ser freqüente. Por exemplo, uma transação contendo {tomate, mussarela, cerveja} também contém {tomate, mussarela}. Se {tomate, mussarela, cerveja} for freqüente, então {tomate, mussarela} também deve ser freqüente. Não há necessidade de gerar ou testar qualquer superconjunto de itemset não freqüente.

O algoritmo Apriori calcula os itemsets freqüentes, de tamanho k (L_k), com base nos itemsets candidatos de tamanho k (C_k). Para tanto, utiliza um passo de junção e um passo de poda. No passo de junção, C_k é gerado pela junção de itemsets de L_{k-1} com outros de L_{k-1}. No passo de poda, eliminam-se alguns candidatos, pois qualquer (k-1)-itemset que não seja freqüente, não pode ser um subconjunto de um frequent k-itemset. Para a poda, os candidatos são testados por comparação, com itemsets já considerados freqüentes para determinar quais deles (candidatos) são, de fato, freqüentes.

Os itens em L_{k-1} são classificados. Assim, o Passo de Junção dos L_{k-1} tem a forma abaixo. Para inserir um candidato em C_k o que se faz é selecionar $p.item_1$, $p.item_2$, ..., $p.item_{k-1}$, $q.item_{k-1}$ de L_{k-1} p, $L_{k-1}q$, em que $p.item_1 = q.item_1$, ..., $p.item_{k-2} = q.item_{k-2}$, $p.item_{k-1} < q.item_{k-1}$., tal como visto na figura 2.1.

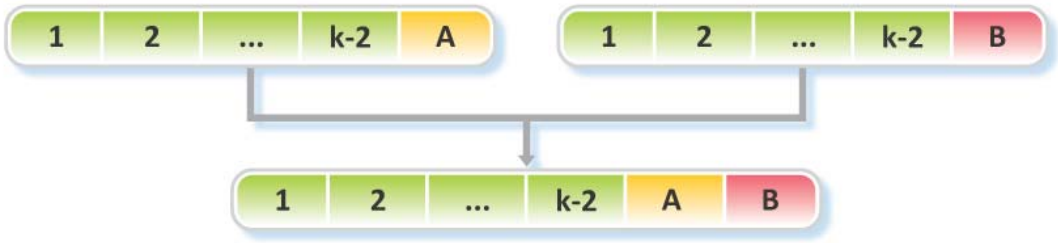


Figura 2.1 Junção de dois (k-1)itemsets

No passo de poda, todos os itemsets c em C_k serão excluídos, se qualquer um de seus (k-1) subconjuntos não pertencer a L_{k-1}.

Como exemplo de geração de candidatos, seja L3 = {abc, abd, acd, ace, bcd}. Na etapa de autojunção, $L_3 * L_3$ se obtém abcd de abc e abd; acde de acd e ace. Na etapa de poda (antes da contagem de suporte), acde é removido, pois ade não está em L_3. Em conseqüência, C_4 = {abcd}.

Para agilizar a análise dos candidatos $C_{k,}$ estes são armazenados em uma hash-tree. Os nós deste tipo de árvore podem ser folhas ou nós interiores. Cada folha contém uma lista de itemsets e cada nó interior, uma hash table. As hash tables contêm buckets, que apontam para outros nós interiores. Diz-se que a raiz da hash table está no nível 1. Nós interiores, no nível d, apontam para nós no nível d+1.

Para armazenar na árvore um itemset c, inicia-se pela raiz e percorre-se a árvore até encontrar uma folha. O percurso é feito, em cada nó, no nível d, aplicando uma função hash ao d-ésimo item do itemset, cujo valor será a entrada na hash table do nó.

Inicialmente, todos os nós da árvore são folhas. À medida que são incluídos itemsets em um nó (folha), sua população aumenta, até atingir um patamar especificado, ocasião na qual o nó se torna nó interior.

Para encontrar todos os candidatos em uma transação t, procede-se como descrito a seguir.

Em uma folha, verificam-se quais dos seus itemsets estão contidos em t e incrementa-se o contador de suporte do itemset encontrado. Tendo chegado a um nó interior, aplicando o hash ao item i, aplica-se o hash a cada elemento que sucede i em t e, recursivamente, o mesmo procedimento ao nó do bucket correspondente. Na raiz, aplica-se o hash a todos os itens de t.

Na raiz, aplicando o hash a cada item em t, garante-se que só serão ignorados os itemsets que iniciam com um item não presente em t. Como os itens são classificados alcançando um nó por hash, do item i, só é necessário considerar os itens em t que ocorrem depois de i.

Como exemplo, considere-se uma hashtree com buckets de tamanho 2, na qual já tenham sido incluídos os itemsets {BCF, BFP, CFP, CPW, FPW}. Seu aspecto seria o da figura 2.2.

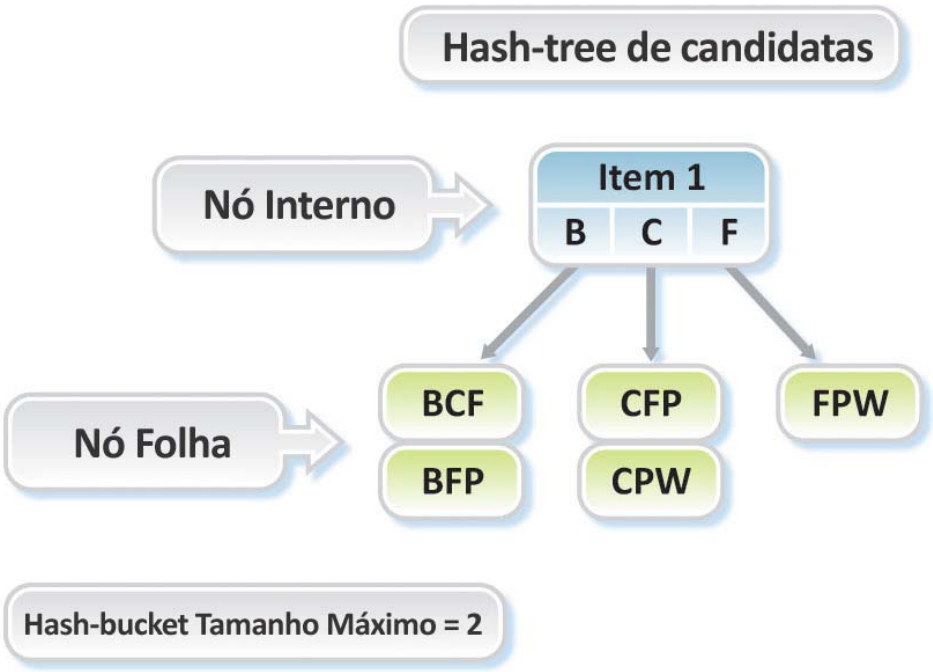


Figura 2.2 Hash tree com cinco itemsets

Caso se desejasse incluir o itemset BPW, o bucket iniciado por B estaria lotado, e o nó seria transformado em nó interno, como se pode ver na figura 2.3.

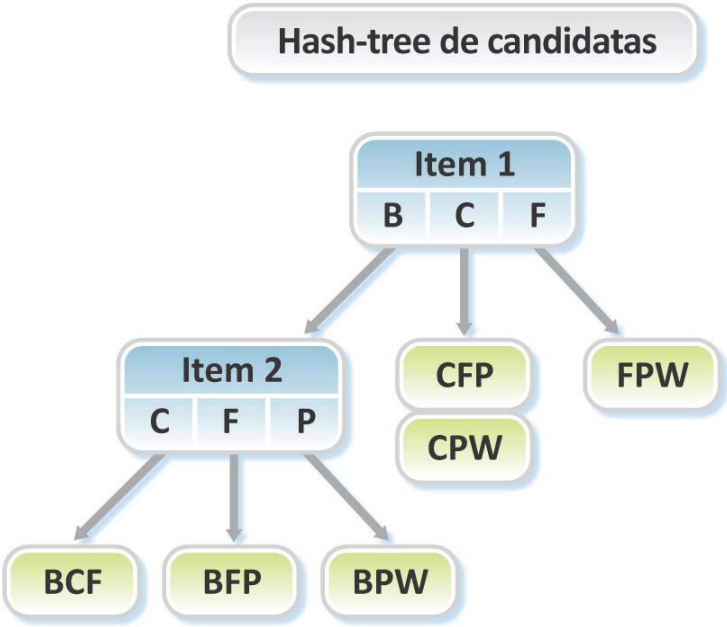


Figura 2.3 Hash tree com seis itemsets

Para a contagem, suponha-se que a hash tree tenha o aspecto da figura 2.4, e que se deseje fazer a contagem dos itemsets presentes na transação BCFW.

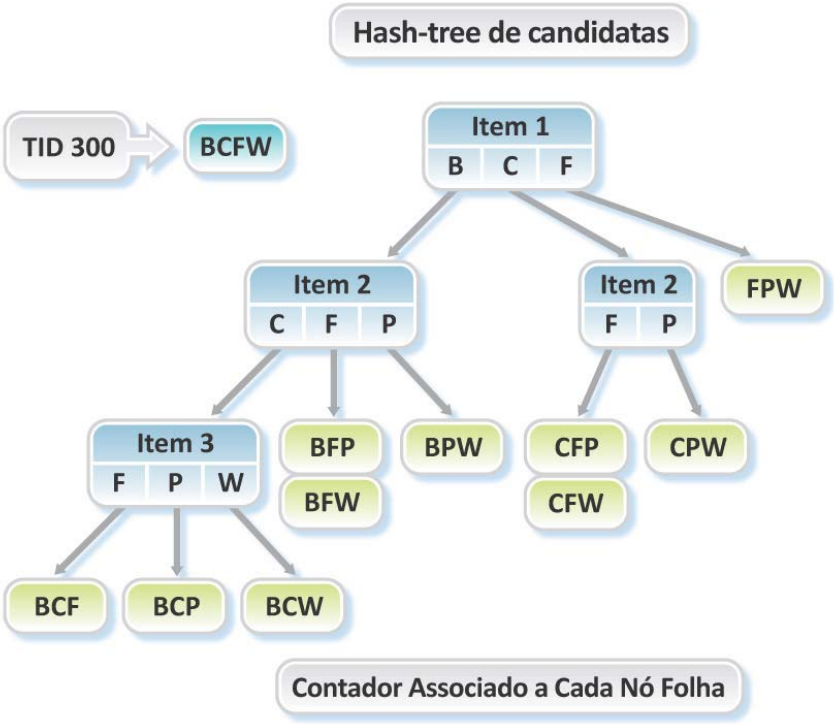


Figura 2.4 Hash tree com dez itemsets

Os itemsets BCF, BCW, BFW e CFW terão seus contadores de suporte incrementados e serão alcançados via hash tree sem necessidade de percurso seqüencial sobre todos os itemsets candidatos.

2.2.2 Outros algoritmos de mineração de regras de associação

Procurando os itemsets, freqüentes percorre-se o reticulado espaço de busca para encontrar a fronteira entre estes e o não freqüentes. Para tanto, há duas estratégias: a busca em profundidade e a busca em largura.

A busca em largura atravessa o reticulado nível a nível, usando padrões freqüentes no nível k para gerar candidatos ao nível k + 1. São podados os itemsets não freqüentes e mantidos os freqüentes, para emprego no nível k + 2 e assim por diante. Esta abordagem emprega muitas varreduras da base de dados e não é a mais adequada na mineração de bases com longos padrões freqüentes. Inicialmente,

os 1-itemsets freqüentes são gerados e usados para gerar candidatos maiores, a serem testados para os tamanhos 2 e maiores.

Na busca em profundidade, tenta-se detectar os padrões longos no início, e apenas se retorna para gerar os padrões freqüentes a partir dos longos considerados freqüentes. Para padrões longos, a busca em profundidade supera a busca em largura. No caso de bases de dados esparsas, onde candidatos longos não ocorrem com freqüência, a busca em profundidade mostra resultados não satisfatórios.

A determinação do suporte dos itemsets pode ser feita por contagem ou por interseção de conjuntos. Na contagem, para cada itemset estudado, um contador é, inicialmente, zerado. Todas as transações são varridas e sempre que um candidato é reconhecido em uma transação, seu contador é incrementado. A investigação de candidatos e a geração de subconjuntos é integrada (p.e. via hash tree). Não são gerados todos os subconjuntos de cada transação, mas apenas aqueles contidos nos candidatos, ou que tenham um prefixo comum a um dos candidatos [AGR94]. Na interseção de conjuntos, para cada item cria-se uma lista de identificadores de transações (tidlist) que contenham este item. A lista para o item A é A.tidlist, e para o item B é a B.tidlist. Caso um candidato seja $C = A \cup B$, então C.tidlist = A.tidlist $\cap$ B.tidlist e o suporte do itemset candidato é dado por |C.tidlist|. Aumenta-se a eficiência do processo mantendo as tidlists classificadas. A Tabela 3.1 mostra um esquema de classificação dos algoritmos de mineração de regras de associação mais utilizados.

Tabela 3.1 Classificação dos algoritmos de mineração de regras de associação

Tipo de busca	Determinação do suporte dos itemsets	Algoritmo
Em profundidade	Contagem	Apriori, AprioriTID, DIC
	Interseção de conjuntos	Partition
Em largura	Contagem	FP-growth
	Interseção de conjuntos	Eclat

AprioriTID [AGR94] é uma extensão da abordagem Apriori, e a varredura da base de dados é substituída pela varredura de uma representação interna de cada transação pelos candidatos correntes que ela contém.

O conjunto de candidatos C_k é substituído por um outro conjunto $\overline{C_K}$, da forma <{TID,{X_k}>, sendo que cada X_k é um k-itemset potencialmente freqüente, na transação com identificador TID. Para k=1, $\overline{C_1}$ corresponde à base de dados de transações. Considerando uma transação t, o membro correspondente de $\overline{C_K}$ é <t.TID,{c $\in$ C_k| c esteja contido em t}>. O processo de obtenção dos itemsets freqüentes L_k prossegue, mas a contagem de suporte é feita exclusivamente em memória. Obtém-se C_k a partir de L_{k-1} e $\overline{C_K}$ a partir de C_k e de $\overline{C_{K-1}}$. Toma-se cada entrada de $\overline{C_{K-1}}$ e determinam-se os candidatos de C_k contidos nessa transação. Assim, obtém-se o suporte de C_k e $\overline{C_K}$. Os itemsets de C_k com suporte adequado são usados para a criação de L_k.

AprioriHybrid mescla as características de Apriori e AprioriTID. Inicia com a técnica Apriori e quando diminui a necessidade de memória, passa a utilizar a técnica de listas de AprioriTID. Na heurística, usada para passar do Apriori para o AprioriTID, estima-se o tamanho de $\overline{C_K}$ a partir do tamanho de C_k, por meio da expressão $tamanho(\overline{C_k}) = \sum_{candidatos\ c \in C_k} \sup orte(c) + número\ de\ transações$. Caso $\overline{C_K}$ caiba na memória e o número de candidatos seja decrescente, passa-se do Apiori para o AprioriTID.

DIC ou Dynamic Itemset Counting [BRI97] é um algoritmo que, sempre que um candidato atinge o patamar especificado de suporte mínimo, inicia a geração de candidatos adicionais baseados nele, mesmo que não tenham sido percorridas todas as transações. Para isto, utiliza-se uma árvore de prefixo, na qual cada nó é atribuído exclusivamente a um itemset freqüente candidato. Desta forma, ao contrário das hash trees, ao alcançar um nó, pode-se ter certeza que o itemset correspondente está contido na transação. A combinação de determinação de suporte e geração de candidatos reduz o número de varreduras.

No processo denominado partição (Partition) [SAV95], pretende-se determinar o suporte de todos os (k-1) candidatos, antes da contagem dos k–candidatos. Para fazer isto com as tidlists, há necessidade de memória muito grande, então, o método divide a base de dados em diversas partições, que são tratadas

independentemente. Depois de encontrados os itemsets freqüentes em cada partição, é necessário executar uma varredura final, para verificar se os itemsets freqüentes das partições também o são no nível da base de dados.

O algoritmo denominado FP-Growth [HAN00], em uma etapa de pré-processamento, gera uma árvore FP-tree, que é uma representação condensada dos dados das transações. A geração da árvore é feita por busca em largura e contagem de ocorrência de itemsets indo, diretamente, a uma parte dos itemsets do espaço de busca. Em um segundo passo, FP-Growth usa a FP-tree para derivar os valores de suporte de todos os itemsets freqüentes.

Uma FP-tree consiste em uma raiz rotulada como "null", um conjunto de sub-árvores de prefixo, como filhas da raiz, e uma freqüent-item-header-table. Cada nó das sub-árvores de prefixo tem três campos: o item-name, o contador, e o nó-link. O item-name registra o item representado pelo nó, o contador armazena o número das transações representadas pela parcela do caminho que alcança este nó, e o nó-link aponta o nó seguinte, na FP-tree, que armazena o mesmo item-name, ou null, se não houver nenhum outro nó com o mesmo item-name.

Cada entrada na freqüent-item-header-table tem dois campos: item-name e cabeça de nó-link (um ponteiro para o primeiro nó na FP-tree, que armazena o mesmo item-name).

Existe uma FP-tree condicional para cada entrada na freqüent-item-header-table. Para obter uma FP-tree condicional de um item-name, entra-se na freqüent-item-header-table com o item-name, e na FP-tree, elimina-se tudo o que fica abaixo dos nós apontados pelos nó-links. Depois, sobe-se em direção à raiz, ajustando os contadores, de modo que nenhum contador seja maior do que o contador do nó com o item-name da folha correspondente.

Acumulam-se, desta forma, todos os prefixos de caminho de um itemset. Se a FP-tree contiver apenas um caminho, o conjunto dos padrões freqüentes pode ser gerado pela enumeração das combinações dos subcaminhos.

O algoritmo Eclat [ZAK97] combina a busca em largura com o uso de interseção de tidlists. A busca em largura permite guardar na memória apenas as tidlists, no caminho da raiz até a classe correntemente estudada, sem necessidade de partições. É feita uma otimização por interseções rápidas. Ao fazer a interseção

de duas tidlists, só nos interessa o resultado, se sua cardinalidade atingir o suporte mínimo especificado, significando que pode-se interromper cada interseção sempre que houver certeza de que ela não atingirá o patamar especificado. A versão original apenas minerava itemsets freqüentes de tamanho igual ou maior que 3.

Direct Hashing and Pruning, ou DHP [PAR95] é um método de mineração de regras de associação baseado no Apriori. Da mesma forma, apresenta os conjuntos de candidatos (C_k) e de itemsets freqüentes (L_k). A diferença consiste em só considerar como candidatos os itemsets que pareçam poder vir a ser itemsets freqüentes.

Os itemsets freqüentes têm algumas características que podem ser aproveitadas:

- todo subconjunto não vazio, de um itemset freqüente, é um itemset freqüente;
- uma transação é relevante para a descoberta de (k+1)-itemsets freqüentes, se possui, pelo menos, k+1 k-itemset freqüentes;
- na transição de C_k para L_k, se o número de k-itemsets for menor que k+1, a transação será omitida na geração de C_{k+1};
- se uma transação contiver um ou mais (k+1)-itemsets freqüentes, para cada item desses itemsets aparecem, pelo menos, k candidatos de C_k.

  Usa-se uma função de espalhamento (hash), para otimizar o desempenho.

Inicialmente, gera-se o conjunto C_1, criando-se, a seguir, uma hash tree, com todos os itens da base de dados, varrendo-se os itens. Se o item corrente estiver na árvore (H_1), seu contador de ocorrências é incrementado, caso contrário o item é inserido na árvore com contador igual a 1. Todos os possíveis subconjuntos de dois itens são gerados e inseridos em H_2. O endereço de um subconjunto é calculado em relação à posição de seus dois elementos em C_1, usando uma hash function. Ao inserir um item na hash tree, o peso de seu endereço é incrementado de 1. C_2 é gerado a partir de L_1, tal como no Apriori, porém apenas os elementos mapeados nos endereços, cujo peso alcance o suporte mínimo especificado, são levados em consideração. A hash table H_k é usada na geração do conjunto de candidatos Ck. Cada endereço de Hk contém um número de subconjuntos de cardinalidade k. Seu peso indica o número de elementos. Os conjuntos de itemsets freqüentes Lk são

calculados junto com os candidatos Ck. Quando não houver mais candidatos, o processamento se encerra, e usam-se os Lk para derivar as regras de associação.

Uma comparação feita entre os algoritmos mais freqüentes de mineração de todas as regras de associação, tais como Apriori, DIC, Partition e ECLAT [HIP00] mostrou que, muito embora em relação às estratégias empregadas, sejam identificadas diferenças fundamentais, em relação ao custo computacional os resultados não revelam diferenças significativas.

3 REDUÇÃO DA CARDINALIDADE DAS REGRAS DE ASSOCIAÇÃO

A redução da cardinalidade das regras de associação, geradas em um processo de mineração, é um imperativo, pois este número cresce, exponencialmente, com o aumento do número de atributos em consideração. Os seres humanos têm notável capacidade de raciocinar em ambientes e processos de síntese, mas seu "insight" perde foco, quando tem de enfrentar grande número de dados. Nosso cérebro e nosso sistema visual são programados para a concentração, e não para a dispersão. A esmagadora maioria das regras geradas tem pouca ou nenhuma serventia e, contudo, age de forma a dispersar a atenção do pesquisador e/ou usuário. O problema é bem conhecido e tem sido exaustivamente estudado. A maioria dos estudos que buscam a redução do número de regras de associação a considerar, baseiam-se em analises sintáticas e em medidores de interesse das regras.

Propostas sintáticas pecam por podar o que não deveria ser podado e deixar de podar o que deveria ser podado.

O enxugamento do número de regras de associação pode ser conduzido por pós-processamento dos resultados produzidos pelo modelo Suporte/Confiança ou, diretamente, no algoritmo de mineração das regras de associação.

A descoberta de regras de associação tem elevada complexidade computacional e gera um número muito grande de regras. Para reduzir o custo computacional pode-se reduzir:

- o número de itemsets candidatos, por exemplo, utilizando o princípio Apriori;
- o número de transações estudado, por exemplo, usando partição, DHP ou Eclat;
- O número de comparações, por exemplo, usando hash tree.

Uma vez gerado o conjunto das regras de associação torna-se necessário reduzir sua cardinalidade, o que é feito pelo mecanismo de poda. O problema corrente passa a ser quais as regras que devem ser podadas. Uma primeira abordagem utiliza a poda estatística, por meio de medidas objetivas de avaliação da interessabilidade das regras, ou seja, medidas que destaquem a utilidade e a inesperabilidade das regras [SIL95]. A outra abordagem é feita por meio de medidas subjetivas. Esta última depende do conhecimento do domínio, expresso por meio da ação do usuário. Os resultados são, significativamente, melhores, mas tornam inviável a automatização do processo. Esta tese busca métodos semânticos que, a partir de parâmetros, de maneira automática, façam a filtragem para enxugamento do conjunto de regras de associação mineradas.

Chama-se de regra forte a regra com regularidade e alta confiança, para uma grande quantidade de instâncias, enquanto uma regra fraca também apresenta regularidade e alta confiança, porém para uma pequena quantidade de instâncias [MEL04]. A avaliação objetiva busca determinar a qualidade dos padrões gerados em relação a sua compreensibilidade e grau de interesse.

As medidas de avaliação de regras fornecem uma indicação da força da associação hipotética entre o antecedente e o conseqüente de uma regra. Diz-se que uma instância está coberta por uma regra, se o antecedente da regra for verdadeiro para a instância.

3.1 PÓS-PROCESSAMENTO POR REGRAS SINTÁTICAS

A descrição das medidas de avaliação objetivas de regras de associação utiliza, como base padrão, as tabelas de contingência, que são generalizações de matrizes de confusão.

Uma tabela de contingência, para uma regra de associação LHS $\cup$ RHS, é aquela na qual LHS representa que o antecedente é verdadeiro, e $\overline{\text{LHS}}$ denota seu complemento (conjunto de instâncias para o qual LHS é falso). O mesmo vale para RHS. LHS RHS denota LHS $\cup$ RHS, $\overline{\text{LHS}}$ RHS denota $\overline{\text{LHS}}$ $\cup$ RHS, e assim por diante. As chamadas medidas relativas são aquelas que comparam o suporte de uma regra com o suporte esperado, sob a hipótese de independência estatística. A notação n(LR) significa o número de vezes em que aparece, na base de dados de transações, a ocorrência simultânea do antecedente (L) e o conseqüente da regra (R). As tabelas de contingência possuem quatro linhas e quatro colunas. A primeira coluna representa a identificação de ocorrência ou não ocorrência do antecedente (L). A segunda representa o número de vezes da ocorrência do conseqüente da regra (R). A terceira coluna representa o número de vezes da não ocorrência do conseqüente da regra (R). A última coluna representa a soma da segunda e terceira colunas.

Algumas das muitas medidas objetivas encontradas na literatura [LAV99], e que foram implementadas no Sistema ADDMiner, utilizado neste trabalho, são descritas a seguir. Serão mostradas duas notações para a descrição das medidas e da matriz de contingência. Para a definição das medidas, a notação adotada foi mais longa (facilita o entendimento, pois $\overline{\text{LHS}}$ é mais intuitivo do que l, por exemplo), como se vê na figura 3.2, e para a exibição das expressões foi adotada a mais curta (facilita a leitura), como se vê na figura 3.1.

Notação para a exibição das expressões
L $\equiv$ ocorrência do antecedente
l $\equiv$ não ocorrência do antecedente

R ≡ ocorrência do conseqüente

r ≡ não ocorrência do conseqüente

LR ≡ ocorrência simultânea do antecedente e do conseqüente

lr ≡ não ocorrência do antecedente nem do conseqüente

Lr ≡ ocorrência do antecedente e não ocorrência do conseqüente

lR≡ não ocorrência do antecedente e ocorrência do conseqüente

	R	r	
L	n(LR)	n(Lr)	n(L)
l	n(lR)	n(lr)	n(l)
	n(R)	n(r)	N

Figura 3.1 Contadores de ocorrências em notação curta

Notação para a exibição das descrições

LHS ≡ ocorrência do antecedente

$\overline{\text{LHS}}$ ≡ não ocorrência do antecedente

RHS ≡ ocorrência do conseqüente

$\overline{\text{RHS}}$ ≡ não ocorrência do conseqüente

LHS RHS ≡ ocorrência simultânea do antecedente e do conseqüente

$\overline{\text{LHS}}$ $\overline{\text{RHS}}$ ≡ não ocorrência do antecedente nem do conseqüente

LHS $\overline{\text{RHS}}$ ≡ ocorrência do antecedente e não ocorrência do conseqüente

$\overline{\text{LHS}}$ RHS ≡ não ocorrência do antecedente e ocorrência do conseqüente

	RHS	$\overline{\text{RHS}}$	
LHS	n(LHS RHS)	n(LHS $\overline{\text{RHS}}$)	n(LHS)
$\overline{\text{LHS}}$	n($\overline{\text{LHS}}$ RHS)	n($\overline{\text{LHS}}$ $\overline{\text{RHS}}$)	n($\overline{\text{LHS}}$
	n(RHS)	n($\overline{\text{RHS}}$)	N

Figura 3.2 Contadores de ocorrências em notação longa

A simples existência da multiplicidade de medidas indica que não existe consenso sobre a melhor medida [LAV99]. Dentre as medidas mais "recomendadas" podem ser citadas: lift, cobertura, especificidade, novidade, convicção [MEL04] [BUR06].

Lift ou interest

Esta é uma das medidas mais valorizadas para uma regra de associação $A \Rightarrow B$ [BRI97], também conhecida como interest, e indica o quanto mais freqüente torna-se B, quando A ocorre. Esta medida é calculada como o quociente entre a probabilidade conjunta observada e a probabilidade conjunta sob a hipótese de independência. Sua expressão é:

$Lift(A \Rightarrow B) = Prob(A \cup B)/(Prob(A) \times Prob(B)) = Suporte(A \cup B)/(Suporte(A) \times Suporte(B))$

A probabilidade conjunta (de A e B) representa o suporte real do conjunto de itens $(A \cup B)$, enquanto a probabilidade conjunta sob a hipótese de independência representa o suporte esperado do mesmo conjunto.

Esta medida varia entre 0 e ∞. Se $Lift(A \Rightarrow B) = 1$, então, A e B são independentes. Se $Lift(A \Rightarrow B) > 1$, então A e B são positivamente dependentes, ou seja, a presença de A aumenta a probabilidade da ocorrência de B.

Se $Lift(A \Rightarrow B) < 1$, então A e B são negativamente dependentes, o que significa que a presença de A diminui a probabilidade da ocorrência de B. O conceito de elevação vem do fato de o antecedente ter levantado o suporte do conseqüente.

Cobertura

Cobertura de uma regra de associação $A \Rightarrow B$ é a fração das instâncias cobertas pelo antecedente [LAV99]. É calculada pelo suporte do antecedente da regra:

$Cobertura(A \Rightarrow B) = Suporte(A)$

Seu domínio é entre 0 e 1. A cobertura pode ser interpretada como medida de generalidade da regra.

Especificidade

Especificidade de uma regra de associação A $\Rightarrow$ B é a freqüência condicional de que A seja falso dado, de que B é falso [LAV99], o que pode ser expresso da forma:

$$Especificidade(A \Rightarrow B) = f(\overline{A} \mid \overline{B}) = \frac{f(\overline{A} \cup \overline{B})}{f(\overline{B})}$$

Nesta expressão f($\overline{A} \cup \overline{B}$) e f($\overline{B}$), denotam-se as freqüências relativas associadas ao conjunto ($\overline{A} \cup \overline{B}$) e $\overline{B}$, respectivamente. A freqüência relativa de um conjunto $\overline{X}$ é calculada como n($\overline{X}$)/N, em que n($\overline{X}$) é o número de instâncias nas quais X é falso, e N o número total de instâncias. Seu domínio é entre 0 e 1.

Novidade

Introduzida em [PIA91], esta medida tem vários nomes: *Rule Interest* (RI), PS (letras iniciais do nome do autor), e *leverage*. Ela tem por finalidade identificar quão inovadora, interessante ou não usual é uma dada regra, calculada como a diferença entre o suporte real e o suporte esperado da regra de associação, de acordo com a expressão:

Novidade(A $\Rightarrow$ B) = Prob(A $\cup$ B) – (Prob(A) × Prob(B)) =

= Suporte(A $\cup$ B) – (Suporte(A) × Suporte(B))

O primeiro termo da expressão é o suporte da regra, indicando que valores elevados de suporte apenas são interessantes quando os valores de suportes do antecedente e/ou do conseqüente são relativamente pequenos. Seu domínio fica entre -0.25 e 0.25. Se Novidade(A $\Rightarrow$ B) = 0, então A e B são independentes. Se Novidade(A)B) > 0, então A e B são positivamente dependentes. Se Novidade(A $\Rightarrow$ B) < 0, então A e B são negativamente dependentes. Quanto maior o valor da medida novidade, mais interessante é a regra.

Convicção

A convicção [BRI97] mede o poder associativo entre o antecedente e o conseqüente de uma regra de associação. Ao contrário do lift e da novidade, a convicção não é simétrica, pois convicção(A $\Rightarrow$ B) $\neq$ convicção(B $\Rightarrow$ A). A convicção

permite avaliar uma regra de associação, considerando-se o sentido da implicação. O valor da convicção para A $\Rightarrow$ B é calculado por:

$$Convicção(A \Rightarrow B) = \frac{\sup orte(A) \times \sup orte(\overline{B})}{\sup orte(A \bigcup \overline{B})}$$

O valor da convicção pode variar entre 0 e ∞, apresentando o valor 1 quando os conjuntos A e B são independentes, e 1 quando o valor da confiança for igual a 100%.

A simples existência da multiplicidade de medidas indica que não existe consenso sobre a melhor medida [LAV99]. Dentre as medidas mais "recomendadas", podem ser citadas lift, cobertura, especificidade, novidade, convicção [MEL04] [BUR06].

Nos experimentos realizados, foram utilizadas as medidas citadas, exceto a cobertura, porque é a medida tradicional de aferição da generalidade de uma regra, mas não é boa medida balizadora de corte, porque, sendo a fração das instâncias coberta pelo antecedente, desconsidera o conseqüente da regra.

Na Tabela 3.2, são exibidas as definições e expressões das medidas objetivas, de interesse das regras de associação, selecionadas para a parte experimental. Para comparação com as expressões conhecidas, são exibidas nessa tabela, também as medidas suporte e confiança, que permitem a mineração das próprias regras de associação. Estas últimas são usadas na mineração e não no pós-processamento das regras.

Tabela 3.2 Definições de algumas medidas de interesse

Medida	Definição	Expressão
Suporte	Incidência de um itemset ou regra no conjunto, ou seja, freqüência de X ou LHS U RHS, no conjunto de dados.	m[1] = LR
Confiança	Freqüência com que LHS e RHS ocorrem juntos em relação ao número de transações onde ocorre LHS.	m[2] = LR / L;
Lift	Razão entre confiança e confiança esperada, ou inesperabilidade da regra.	m[4] = m[2] / R;
Especificidade	Freqüência condicional de LHS ser falso, dado que RHS seja falso.	m[10] = lr / r
Cobertura	Fração das instâncias cobertas pelo antecedente da regra (LHS) ou generalidade da regra.	m[11] = L;
Novidade	Identificação da inovação, interesse ou singularidade de uma regra, ou comparação entre o valor observado de LHS e RHS e o valor esperado, se LHS e RHS forem independentes.	m[12] = LR - R * L
Convicção	Representação do poder associativo entre o antecedente e o conseqüente.	m[22] = (1 - R) / (1 - LR/L)

3.2 PÓS-PROCESSAMENTO POR REGRAS SEMÂNTICAS

Pela Encyclopaedia Britannica, semântica é o estudo do significado, e tem a mesma raiz grega de semiótica. Semântica é a doutrina do significado, particularmente do significado lingüístico.

O intercâmbio automático de informações, dentro de uma linguagem única e com um modelo comum de mundo, é possível, utilizando-se conhecimentos semânticos. Contudo, diferentes fontes de informação ou de linguagem trazem diferentes semânticas. Semânticas baseadas em ontologias permitem este intercâmbio [CIO99].

O uso da semântica, no pós-processamento do conjunto das regras de associação e dos padrões seqüenciais permite uma filtragem, pelo conteúdo e pelo

significado, do resultado da mineração. A intrínseca dificuldade da natureza subjetiva do significado tem levado os pesquisadores a dar preferência às abordagens sintáticas do pós-processamento, e poda dos conjuntos de regras e padrões minerados.

Uma abordagem semântica pode ser encontrada na mineração de conexões entre textos, em bases de dados biomédicos, usando enriquecimento semântico e poda, agindo nas regras individualmente pelo contexto [XIA05]. Em outra abordagem semântica, de [CHE08], as regras de associação são divididas em cinco categorias: triviais, conhecidas e corretas, desconhecidas e corretas, conhecidas e incorretas e desconhecidas e incorretas. O enfoque é bastante interessante, porém altamente dependente de julgamento humano.

A proposta de [XIA05] analisa o conteúdo semântico de títulos, procurando relacionamento de sinônimos. Nosso enfoque será em relacionamentos do tipo "é-um" e "parte-de". O trabalho de [CHE08] faz rotulação humana nas regras mineradas, ao contrário de nossa proposta algorítmica. [HSU03] e [CHA07] citam, de maneira explícita, o enriquecimento da informação minerada. [BAR08] destaca a avaliação da redundância de uma regra, em função da condição de mais geral ou mais específica que as demais.

Este trabalho colocou-se nesse nicho de pesquisa, trabalhando com algoritmos de pós-processamento e generalização, que executam, simultaneamente, o enriquecimento do conjunto de regras de associação e poda semântica.

4 ONTOLOGIAS

Esta tese baseia-se na exploração de conhecimento de domínio presente nas ontologias, ainda que de maneira subjacente. A clássica definição de ontologia é: uma especificação explícita de uma conceituação" [GRU93].

Conceitos são, recursivamente, definidos sobre conceitos. Não existe um nível primitivo de conceitos significativos, em termos dos quais os demais possam ser definidos [HEY90]. Uma maneira de defini-lo é listar o conjunto de conceitos que o

encadeiam (ou disparam), mais o conjunto de conceitos por ele encadeados (ou disparados). O conhecimento sobre um domínio é expresso em função de conceitos e relações entre eles.

Uma ontologia é um formalismo declarativo de uma rede semântica. Nesse formalismo, o conjunto de objetos do domínio é chamado de universo de discurso. O conjunto de objetos, assim como os relacionamentos entre eles, que possam ser descritos, se refletem na representação vocabular, ou dicionário, que representa conhecimento armazenado em uma base de conhecimento.

Uma rede semântica, utilizada como representação de conhecimento, é um grafo dirigido, que consiste de vértices que representam conceitos e arcos que representam relações semânticas entre os conceitos. Redes semânticas são um tipo comum de dicionário, legível por máquina, que envolvem associações semânticas imprecisas, todavia úteis para o ser humano. Seu poder de expressão é igual ou superior ao da lógica de predicados de primeira ordem, confiáveis para dedução lógica automática.

Pode-se dizer que uma ontologia é a declaração de uma teoria lógica e especificada em uma lógica descritiva, que se baseia em conceitos (classes) e papéis. Conceitos (classes) são interpretados como conjuntos de objetos, e papéis, como relações binárias entre objetos. A ontologia de uma rede semântica pode ser descrita definindo um conjunto de termos representativos. Definições associam nomes de entidades do universo do discurso (classes, relações, funções, etc.), à descrição do significado dos termos, em textos legíveis, e a axiomas formais, que restringem a interpretação e caracterizam o uso "well-formed" dos termos. Em essência, pode-se dizer que as ontologias trazem a semântica do domínio para o "centro do palco".

No estudo e emprego das ontologias, distinguem-se três níveis de entidades [SMI06]:

- no primeiro nível, encontram-se os objetos, processos, características, estados, etc;
- no segundo, as representações cognitivas da realidade, por parte de pesquisadores e outros profissionais;

- no terceiro, a concretização das representações cognitivas (produtos conhecidos representando o conhecimento).

Em particular, o que se pretende, é utilizar relações ontológicas, do tipo "é-um" (generalização e especialização) e "parte-de" (composição), como balizadoras semânticas, que permitam uma redução do número de regras de associação, geradas pela mineração de dados, de maneira mais eficaz do que as tentativas meramente sintáticas, que permitam, ao mesmo tempo, um enriquecimento semântico do conjunto de regras mineradas.

Os problemas concernentes são como gerar uma ontologia [GRU93].

Como exemplos de representações gráficas de ontologias, são exibidas as figuras 4.1 e 4.2. A figura 4.1, criada em Altova XMLSpy, um ambiente de desenvolvimento e editor XML, vem de uma publicação do grupo de pesquisa em que se insere o autor. Exibe um exemplo de ontologia desenvolvida para caracterizar a possibilidade da Web de viabilizar um modelo de e-cidadão, evidenciando um governo eletrônico como uma composição de informações, serviços e participação dos cidadãos. A figura 4.2 foi criada no software produto ONTO ADDLabs, e mostra parte de uma ontologia de cesta de compras, criada para ilustrar novo modelo de descoberta de conhecimento, pela introdução dos conceitos de coesão de conhecimento (KC) e distância semântica entre conceitos (termos).

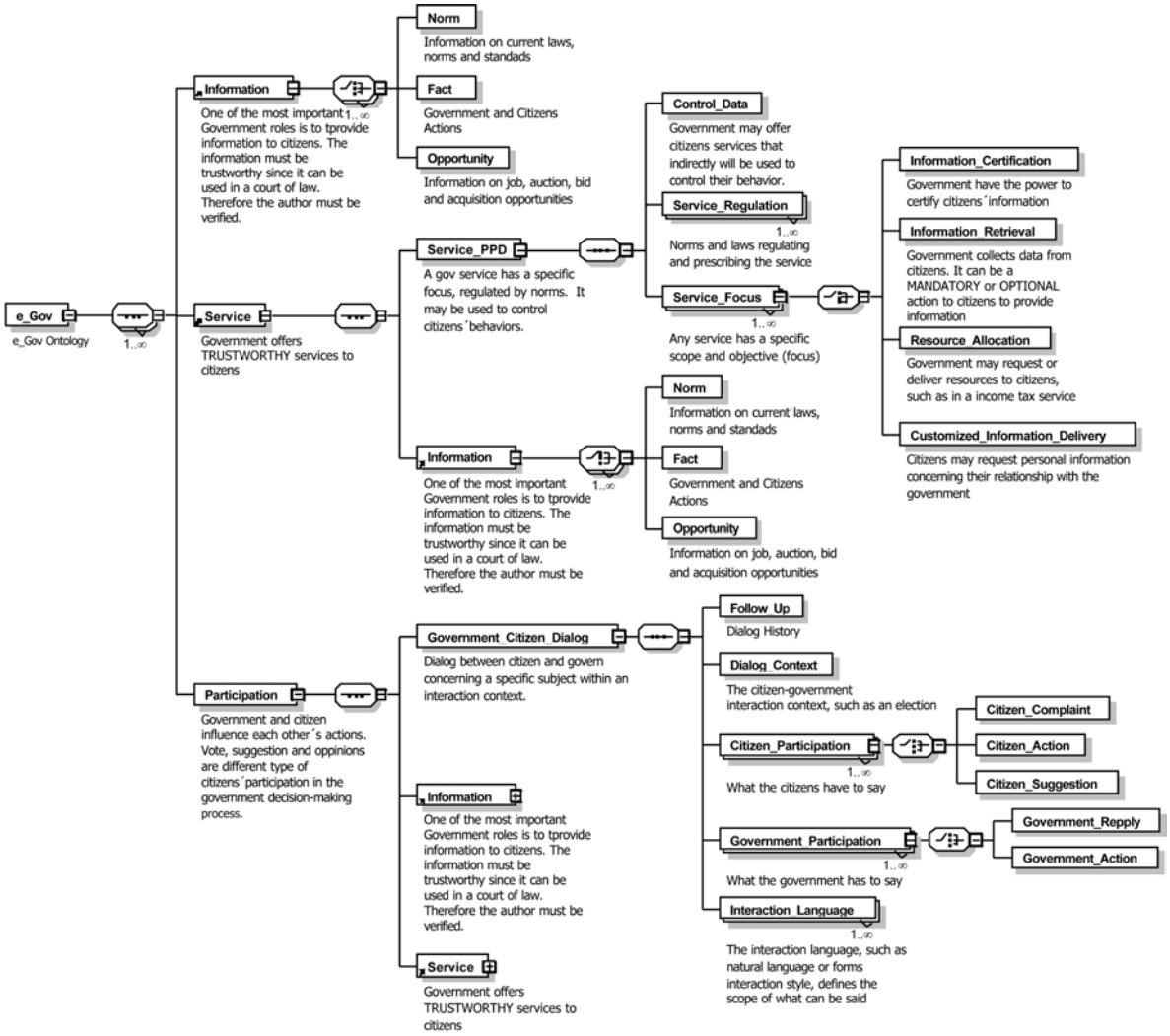


Figura 4.1 Exemplo de ontologia de e-gov extraída de [GAR06]

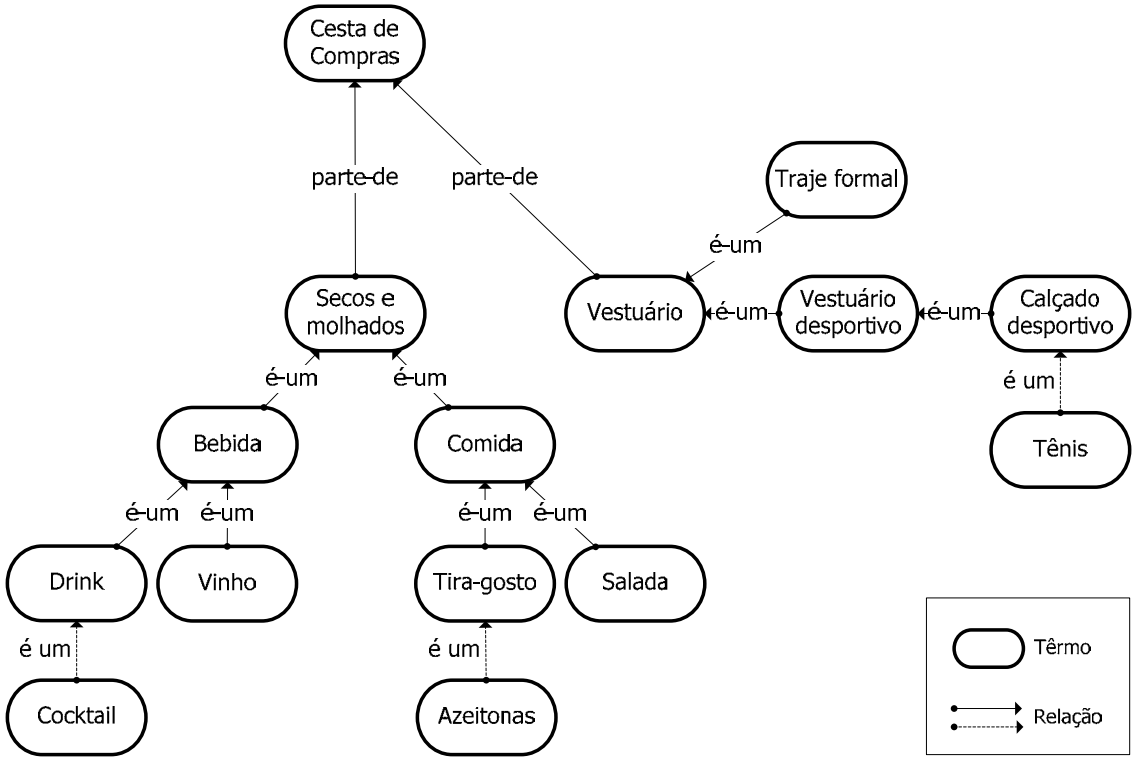


Figura 4.2 Exemplo de ontologia de cesta de compras extraída de [GAR09]

4.1 INFORMAÇÃO DESPERCEBIDA

O conhecimento de mundo é uma fonte de enriquecimento da mineração de dados, pois o que se deseja é extrair conhecimento, e não somente dados, do repositório de informações. Para ilustrar o poder enriquecedor do conhecimento de mundo, na extração do conhecimento de um repositório de dados, será exibido um exemplo. O item 5.2.1 vai definir um atributo BMI ("Body Mass Index" ou Índice de Massa Corporal), calculado a partir dos atributos Peso e Altura. A relação de dependência, entre o atributo BMI e os atributos Peso e Altura, é definida por f: Domínio({Peso, Altura}) $\Rightarrow$ (Domínio({BMI}), tal que f (BMI) = Peso/(Altura)^2 .
Diz-se que um adulto está "acima do peso", se BMI ≥25. Pode-se supor que tenham sido mineradas as seguintes regras:

R1: (Peso=71 kg) ^ (Altura=1,68 m) ^ (Sexo="F") ⇒ (Pressão-sanguínea="alta")
R2: (Peso=80 kg) ^ (Altura=1,75 m) ^ (Sexo="F") ⇒ (Pressão-sanguínea="alta")
R3: (Peso=86 kg) ^ (Altura=1,80 m) ^ (Sexo="F") ⇒ (Pressão-sanguínea="alta")
R4: (Peso=88 kg) ^ (Altura=1,85 m) ^ (Sexo="F") ⇒ (Pressão-sanguínea="alta")
R5: (Peso=93 kg) ^ (Altura=1,90 m) ^ (Sexo="F") ⇒ (Pressão-sanguínea="alta")

Conhecendo-se o conceito de BMI, pode-se gerar uma regra que traz um "insight" muito maior ao ser humano, da forma

(BMI > 25) ^ (Sexo="F") ⇒ (Pressão-sanguínea="alta").

Todavia, o enriquecedor conhecimento de mundo terá de ser representado junto aos dados minerados, provocando um aumento da massa de dados a ser interpretada. Aparentemente, o conhecimento de mundo vem complicar o problema do número gigantesco de regras mineradas, em vez de contribuir para a sua solução. Porém, o resultado obtido é muito melhor, por permitir que os processos de redução da cardinalidade da massa, por poda de padrões e associações, utilizem critérios semânticos. Nas situações em que o conhecimento de mundo possa ser explorado na etapa de pós-processamento, deixa de ocorrer a expansão do conjunto de regras de associação mineradas. Processos apenas sintáticos reduzem a cardinalidade, mas também podam o que não deveria ser podado, e deixam de podar o que deveria ter sido podado. Isto não ocorre quando se usa o conhecimento de mundo como mecanismo de poda.

Sem o uso do conhecimento de mundo, das ontologias, esse conhecimento passaria despercebido ou precisaria ser garimpado pela intuição do analista.

4.1.1 Agrupamentos de valores de atributos

A maneira mais simples de utilização de conhecimento subjacente é fazer o agrupamento das instâncias de atributos, em faixas de valores. Agrupando-se itens a adquirir, em faixas de preço (barato, preço normal, caro), os indivíduos por altura (baixo, médio, alto), ou por produtividade (pequena, média, alta, excelente), pode-se

encaminhar a descoberta muito mais eficazmente, do que tentar o mesmo com as instâncias isoladas.

4.1.2 Redundância

O conhecimento de mundo, agregado aos dados minerados, vai gerar uma redundância, pois coexistirão atributos abrangentes e atributos específicos. Atributos abrangentes, tais como as faixas de valores de itens, generalizadores, ou representantes de agrupamentos, serão aqueles que, sob o ponto de vista do armazenamento de informações são chamados dependentes, uma vez que conhecido um atributo específico (ou determinante), o atributo abrangente (ou dependente) é conhecido. Cada tipo de relacionamento possui seus mecanismos de representação e resolução de dependências.

4.2 ONTOLOGIA NO ESTABELECIMENTO DE DEPENDÊNCIAS

Uma ontologia pode ser representada por um grafo, no qual cada vértice representa um conceito. Estes vértices são conectados por diversos tipos de arcos, mostrando relacionamentos entre conceitos [GEL08]. Na literatura podem ser encontradas diversas classificações de relacionamentos semânticos [GUA00] [WAN99] [STO93], que podem ser: antônimos, sinônimos, inclusão em classe, parte-todo e caso [LAN87], ou fazer parte de uma lista, com 31 relacionamentos [CHA88], ou ainda, sintetizados em "é-um", "parte-de", inclusão ou atributo, etc. [BRA83].

As relações de dependência são relacionamentos entre dois conjuntos de atributos, com respeito à determinada base de dados. A busca dessas relações de dependência nas ontologias, e do relacionamento entre atributos, é uma conseqüência natural da busca de melhor exploração de um domínio do conhecimento. No caso em tela, pretende-se utilizar relações ontológicas do tipo "é-

um" (generalização e especialização) e "parte-de" (composição), como balizadoras semânticas, que permitam redução do número de regras de associação, geradas pela mineração de dados, de maneira mais eficaz do que as tentativas meramente sintáticas. A seleção dos tipos de relacionamento escolhidos para este trabalho ocorreu, em virtude de serem bastante freqüentes nas bases de dados legadas e existentes nas aplicações de mineração de dados. Outro tipo de relacionamento, encontrado com freqüência nas bases de dados correntes, é o relacionamento seqüencial. A exploração de resultados da mineração de padrões seqüenciais é de natureza muito diferente daquela feita nos relacionamentos "é-um" e "parte-de". Os resultados desejados, na busca de padrões, enfocam o atendimento de restrições temporais, janelas de tempo de eventos e de observações, e o tratamento de generalização/especialização. Assim sendo, o foco do trabalho ficou nos relacionamentos "é-um" e "parte-de".

4.2.1 Relacionamento "é-um"

Na representação do conhecimento e na programação orientada a objeto de design, "é-um" é um relacionamento no qual uma classe A é uma subclasse de outra classe B (e, portanto B é uma superclasse de A). "A é-um B" normalmente significa que o conceito A é uma especialização do conceito B, e o conceito B, uma generalização do conceito A [MIT03].

O relacionamento de generalização é um relacionamento taxonômico, entre uma descrição mais geral e uma mais específica, que tem como base o mais geral, expandindo-o. A descrição mais específica é inteiramente consistente com a mais geral e pode conter informações adicionais.

A descrição mais geral é chamada de pai, a mais específica, de filha, um elemento no fecho transitivo é um ancestral e é um descendente.

A generalização define uma hierarquia de abstrações, na qual uma subclasse herda de uma ou mais superclasses, e o relacionamento é do tipo "é-um".

Uma subclasse herda os atributos, operações e relacionamentos de seus pais, e pode ter mais atributos, operações e relacionamentos, além dos herdados.

Uma subclasse pode ser utilizada em qualquer lugar em que também possa ser utilizada a superclasse, mas não vice-versa. Generalização é o nome do relacionamento, e herança é o mecanismo que a relação de generalização representa.

Um conjunto de relacionamentos de generalização representa relacionamentos do tipo "é-um". Nestes, uma classe é caracterizada como subclasse de outra classe. A classe de objetos super-conjunto é chamada classe Generalização, e a de objetos subconjunto, de Especialização.

Um objeto da classe generalização pode conter mais de uma classe especialização. Neste tipo de relacionamento são válidas as seguintes restrições[PAD07]:

- união - cada membro de uma generalização deve ser também de uma das especializações;
- exclusão mútua - nenhum membro de uma classe de objetos especialização pode ser de outra classe de objetos especialização. Um elemento de grupo de especializações de uma generalização são disjuntos dois a dois;
- partição - um grupo de especializações divide uma generalização.

Como exemplo de relação de dependência do tipo "é-um", considere-se o atributo BMI (Índice de Massa Corporal), calculado a partir dos atributos Peso e Altura. Esta relação é definida por f: Domínio({Peso, Altura}) (Domínio({BMI}), tal que f (BMI) = Peso/(Altura)2. A dependência de BMI por Peso e Altura é representada por: {Peso, Altura} $\Rightarrow$ {BMI}, onde BMI é o atributo dependente, e Peso e Altura os atributos determinantes. O atributo IMC também é conhecido como BMI (Body Mass Index).

A relação de dependência X $\Rightarrow$ Y é de grau n, quando existirem n funções que mapeiam os elementos do domínio de X para os elementos do domínio de Y, passando por n - 1 domínios diferentes. A relação de dependência de grau n de Y por X será representada por: X $\overset{n}{\Rightarrow}$ Y

Como exemplo, considere-se o atributo status, calculado a partir do atributo BMI, tal que o primeiro seja uma discretização do segundo. A relação de dependência entre os dois é de grau 1 (n=1), porque existe uma função g : Domínio({BMI}) $\Rightarrow$ Domínio({status}), que mapeia os elementos de {BMI} para os elementos de {status}, passando por 0 (n -1) domínios diferentes. A função g pode ser vista a seguir.

g (Status) = {"abaixo do peso", Se BMI ≤ 19;
"no peso normal", Se 20 ≤ BMI ≤ 24;
"acima do peso", Se BMI ≥25}

A relação de dependência entre status e os atributos peso e altura é de grau 2 (n=2), porque existem duas funções, g e f (f : domínio({peso, altura} $\Rightarrow$ (domínio({BMI})), que mapeiam os elementos do domínio de {peso, altura} para os elementos do domínio de {status}, passando por 1 (n-1) domínio diferente (domínio(BMI)). A relação de dependência entre o atributo status e os atributos peso e altura é representada por: {peso, altura} $\overset{2}{\Rightarrow}$ {status}. Neste trabalho, estudamos o uso do conhecimento prévio do domínio, representado pelas relações de dependência entre os atributos, para eliminar regras redundantes ou de menor impacto para o usuário.

Uma consideração necessária é a da cardinalidade do relacionamento entre atributos determinantes e dependentes. Esta cardinalidade pode ser do tipo m:n ou 1:n (não há sentido no estudo do relacionamento 1:1). Por exemplo, o primeiro caso (relacionamento m:n), que chamaremos de tipo A, ocorre muito freqüentemente quando se utilizam faixas de agregação.. Faz mais sentido para um analista de problemas saber que um tipo de aquisição foi feito por alguém acima do peso, do que por alguém de 83 quilogramas. Normalmente, os atributos determinantes (que darão origem aos atributos agregadores) são em pequeno número. Este tipo de tratamento é bastante usado, no caso de faixas de valores de atributos e de atributos agregadores derivados. Usualmente, uma atividade de pré-processamento cria novas regras com os atributos dependentes estabelecidos em um arquivo de dependências. Esta pesquisa transfere o enriquecimento semântico da mineração de regras para a etapa de pós-processamento, em conjunto com o tratamento de poda.

As cardinalidades do tipo 1:n, recebendo a denominação de tipo B, representam hierarquias e seu exemplo mais comum é o relacionamento do tipo taxonomia. Usualmente, todos os itens da amostra enquadram-se na taxonomia e o número de regras adicionais é muito maior do que no caso anterior. Se um consumidor adquiriu uma garrafa de conhaque, o conhecimento de mundo acrescenta o fato de ter adquirido uma garrafa de bebida alcoólica. A característica fundamental a considerar é que, com muita freqüência, o suporte dos itens, que são folhas da hierarquia taxonômica, é muito pequeno, não permitindo sua captura no processo de mineração fazendo com que este conhecimento possa se perder.

Um problema clássico da mineração de dados é o da escolha da precisão ou da "abertura da peneira". Com especificações que exigem pequenos valores de suporte e confiança ("peneira com malha estreita"), o usuário pode ser sobrecarregado com um grande número de padrões e associações muito parecidos entre si. Caso se diminua a abertura, ou seja, especificando valores maiores de suporte e confiança, alguns padrões e associações interessantes podem não ser identificados. Pode-se tentar contornar este dilema pela pesquisa de representações compactas dos conjuntos freqüentes e regras "fortes", cuja cardinalidade é, substancialmente, menor que a massa minerada.

Pela consideração dos ancestrais na taxonomia, cujo suporte quase sempre é a soma dos suportes dos descendentes, consegue-se capturar comportamentos valiosos.

A incorporação do conhecimento de mundo pode ser feita por pré-processamento mais processamento [HAN95], ou embutindo a criação de regras dependentes na mineração, contemplando os ancestrais dos itens na taxonomia [SRI]. Um arquivo de dependências serve para definir o grafo direcionado acíclico ou "Directed Acyclic Graph" (DAG) da ontologia do domínio (taxonomia). Em havendo múltiplas taxonomias sobre a mesma base de dados (por categorias e por preços de itens, por exemplo), o usuário deve selecionar qual o contexto desejado, e o pré-processamento carrega esse contexto sob a forma de hierarquia preferencial a ser utilizada.

4.2.2 Relacionamento "parte-de"

As relações parte-todo podem ser meronímicas ou mereológicas. Meronímia é uma relação semântica de pertinência. Mereologia é a teoria ou estudo lógico-matemático das relações entre as partes e o todo, e das relações entre as partes no interior de um todo [ZAL08].

As relações parte/todo podem ter a classificação que se segue:

- Relação parte-de
    - Relação meronímica
        - Membro-de
        - Constituído-de
        - Sub-quantidade-de
        - Participa-em
    - Parte-de mereológica
        - Parte-estrutural-de
            - Parte-funcional-de
        - Parte-espacial-de
            - Contido-em
            - Localizado-em
        - Envolvido-em

O nível mais complexo da mereologia é expresso pela "General Extensional Mereology", ou GEM.

A forma mais simples da teoria mereológica configura uma relação reflexiva, anti-simétrica e transitiva, sob a forma de três predicados:

a) tudo é parte de si mesmo
part_of(x, x)
b) duas coisas distintas não podem ser parte uma da outra, ou se x é parte de y, e y é parte de x, então x e y são a mesma coisa
(part_of x, y) ^ part_of(y, x)) $\Rightarrow$ x = y
c) se x é parte de y e y é parte de z então x é parte de z
(part_of (x, y) ^ part_of(y, z)) $\Rightarrow$ part_of(x, z)

Destes predicados e da definição de parte própria, que diz que x é parte própria de y se x é parte de y e y não é parte de x, podem ser inferidas propriedades, nas quais o símbolo $\triangleq$ significa "igual por definição":

d) se x é parte de y e y não é parte de x (definição de parte própria)
proper_part_of(x, y) $\triangleq$ part_of(x, y) ^ ¬part_of(y, x)
e) quando x e y partilham um pedaço de z, isto é reflexivo e simétrico
overlap(x, y) $\triangleq$ $\exists$z(part_of(z, x) ^ part_of(z, y))
f) quando x e y são ambos um pedaço de algum z, isto é reflexivo e simétrico
underlap(x, y) $\triangleq$ $\exists$z(part_of(x, z) ^ part_of(y, z))
g) quando x e y "overlap" mas x não é parte de y
overcross(x, y) $\triangleq$ overlap(x, y) ^ ¬part_of(x, y)
h) quando x e y "underlap" mas y não é parte de x
undercross(x, y) $\triangleq$ underlap(x, y) ^ ¬part_of(y, x)
i) combinação de "parte-de" própria e "overlap"
proper_overlap(x, y) $\triangleq$ overcross(x, y) ^ overcross(y, x)
j) combinação de "parte-de" própria e "underlap"
proper_underlap(x, y) $\triangleq$ undercross(x, y) ^ undercross(y, x)

Estas propriedades comportam suplementações "fraca" e "forte".

k) suplementação "fraca": toda parte própria deve ser suplementada por outra parte própria, disjunta da primeira

proper_part_of(x, y) $\Rightarrow \exists$z(part_of(z, y) ^ ¬overlap(z, x))

l) suplementação "forte": se um objeto falha na inclusão de outro entre suas partes, então deve haver uma parte "restante" do outro (não incluída)

¬part_of(y, x) $\Rightarrow \exists$z(part_of(z, y) ^ ¬overlap(z, x))

As relações de composição são determinadas pela combinação de três propriedades básicas:

- Configuração, que indica se as partes têm particulares relacionamentos funcionais ou estruturais umas com as outras, ou com o objeto que constituem.
- Homeomericidade, que indica se as partes são ou não da mesma espécie que o todo.
- Invariância, que indica se as partes podem ser separadas do todo.

As relações de composição podem ser de vários tipos, baseadas em combinações das propriedades básicas:

- Componente-objeto integral (configuração de partes em um todo); por exemplo; eletricidade é parte da física, cerdas são parte das escovas.

Desta forma, um objeto integral é dividido em partes componentes que, por sua vez, são também objetos.

- Material-objeto (configuração invariante de partes em um todo); por exemplo: móveis são parcialmente feitos de madeira, Dry Martini é parcialmente feito de gin. Especifica de que os objetos são constituídos (fisicamente).
- Porção-objeto (configuração homeomérica de partes dentro de um todo); por exemplo: metros são parte de um quilômetro, farelo de pão é parte de uma fatia de pão. Neste tipo de composição, a característica principal é a quantidade.
- Local-área (configuração homeomérica, e invariante de partes em um todo); por exemplo: Região dos Lagos é parte do Estado do Rio de Janeiro, cume é

parte de uma montanha. Neste tipo de composição, as partes não podem ser removidas.

- Grupo-membro (coleção de partes como um todo); por exemplo: um avião é parte de uma esquadrilha, uma árvore é parte de uma floresta. As partes têm, entre si, ou com o objeto que as engloba, um relacionamento estrutural ou funcional.
- Pertinência-membro (coleção invariante de partes de um todo); por exemplo: Stan Laurel é parte da dupla O Gordo e o Magro, Leandro é parte da dupla Leandro e Leonardo. Caso um membro da composição seja removido, a pertinência é destruída.

Cruzando as propriedades básicas das relações de composição com os tipos de composição pode-se visualizar a Tabela 4.1.

Tabela 4.1 Relações de composição

Tipo de composição	Configuração (relacionamentos funcionais)	Homeomericidade (mesma espécie)	Invariância (não separabilidade)
componente-objeto integral	sim	não	não
material-objeto	sim	não	sim
porção-objeto	sim	sim	não
área-local	sim	sim	sim
grupo-membro	não	não	não
pertinência-membro	não	não	sim

O relacionamento do tipo "parte-de" tem, inversamente, o relacionamento do tipo "tem-partes" e não existem mecanismos conclusivos para certificar a transitividade destes tipos, [HAH03]. Há várias propostas para tratar este tipo de incógnita, como as triplas SEP ("structure, entity, part"). Contudo, mesmo as propostas mais elaboradas reconhecem que a parcela necessária de trabalho humano é grande, para a definição "não primitivamente natural" das estruturas auxiliares (triplas ou equivalente), e muito grande, para o trabalho de conciliação do pré-processamento, o que torna temerário tentar avançar nessa direção, caso se pretenda automatizar a exploração da ontologia na redução da redundância.

Na representação de ontologias por grafos, as funções, os processos e os componentes podem ser representados por um grafo acíclico direcionado, DAG, ou redes. A diferença fundamental entre um DAG e uma hierarquia é que nesta, cada nó só pode ter um ancestral direto, enquanto que em um DAG, um nó pode ter mais de um ancestral direto. Um termo filho pode ser uma instância do ancestral direto (relacionamento "é-um"), ou um componente deste mesmo termo (relacionamento "parte-de"). Outra diferença deve ser feita nas relações de composição, que podem ser representadas por hierarquias (taxonomias), quando a dependência for do tipo em que cada valor específico do conjunto determinante é mapeado, exclusivamente, em uma instância do conjunto dependente; em caso contrário, também podem ser representadas por DAG. Se, em um determinado ambiente, as rodas forem, exclusivamente, de carros, tem-se uma hierarquia; em outro ambiente, rodas de carros e de motos, um DAG. No primeiro caso, o tratamento de filtragem utiliza a hipótese de distribuição uniforme dos suportes das regras, com atributos dependentes de pai comum. No outro caso, este tratamento vai utilizar o suporte e a confiança das instâncias, tal como no caso do relacionamento do tipo "é-um".

A escolha entre generalização (preferencial) e especialização é feita pela hipótese de distribuição uniforme dos suportes das regras, contendo atributos dependentes, com ancestral comum. Se a razão entre o desvio padrão do suporte das seqüências, e a média aritmética da distribuição desses suportes estiver abaixo de um patamar especificado, o comportamento das regras específicas é regular, e estas regras podem ser substituídas pela regra geral e podadas. Caso contrário, a regra geral é que não traz nenhum valor agregado, e pode ser podada.

Aqui também é necessário considerar a cardinalidade do relacionamento entre atributos determinantes e dependentes. Esta cardinalidade pode ser do tipo m:n ou 1:n (não há sentido no estudo do relacionamento 1:1). O primeiro caso (relacionamento m:n), de tipo C, ocorre quando se deseja incluir na mineração de dados o conhecimento da ontologia do domínio, modelada por um grafo genérico, permitindo, por exemplo, a ocorrência de rodas. As rodas podem ser de automóveis, de motocicletas, de tratores, etc., parte de qualquer destes veículos. O tratamento do problema necessita de uma etapa de pré-processamento, que receba um arquivo de dependências com o grafo da ontologia do domínio. Quando houver caso de mais

de um ancestral (carro, moto, trator para roda, por exemplo), o usuário deve selecionar o contexto desejado, e o pré-processamento carrega esse contexto, sob a forma de hierarquia.

4.3 CONHECIMENTO DE MUNDO E AS REGRAS DE ASSOCIAÇÃO

4.3.1 Redundância entre regras de associação

As regras de associação {A,B,C} $\Rightarrow$ {D} e {A,B} $\Rightarrow$ {D} são redundantes, se apresentarem os valores para o suporte de ambas iguais aos valores de confiança. Pode-se dizer que uma regra de associação R' é redundante, se pode ser deduzida a partir de outra, R, presente no conjunto de regras e, por algum motivo, R' é menos relevante do que R [BUR06].

É consenso que regras mais gerais são mais relevantes do que as mais específicas [ZAK00]. Uma regra R1 é mais geral do que uma R2, se R2 puder ser gerada adicionando-se itens ao antecedente ou ao conseqüente de R1. Considera-se a regra R1: X1$\Rightarrow$Y1 redundante, se existir uma regra R2: X2$\Rightarrow$Y2, com sup(R1) = sup(R2) e conf(R1) = conf(R2), tal que X2 $\subseteq$ X1 e Y1 $\subseteq$ Y2. Zaki [ZAK00] apresenta a mineração de regras de associação não redundantes, baseada no conceito de conjuntos freqüentes fechados (closed itemset freqüentes). Um itemset é fechado, se nenhum de seus super-conjuntos imediatos possuir o mesmo suporte. Um itemset freqüente fechado é um subconjunto do conjunto freqüente correspondente. Os super-conjuntos de um itemset freqüente fechado podem ser itemsets freqüentes, mas com suporte menor do que o fechado. Este subconjunto é necessário e suficiente para capturar todas as informações sobre o conjunto freqüente. Os conjuntos freqüentes fechados são os menores subconjuntos representativos dos conjuntos freqüentes, sem que perda de informação. Um itemset é maximal freqüente, se nenhum de seus super-conjuntos imediatos for freqüente. Com a

notação F, para os itemset freqüentes, C para os "closed" itemsets freqüentes e M para os "maximal" itemsets freqüentes, então $M \subseteq C \subseteq F$.

O algoritmo [ZAK00] considera apenas regras entre os conjuntos freqüentes fechados adjacentes. A partir do conjunto de regras gerado, pode-se inferir todas as regras de associação possíveis, através de operações, como transitividade e incremento. O resultado é um subconjunto das regras de associação que permite ao usuário inferências e ampliações.

Encontra-se em [BAS00], um algoritmo para extrair somente as regras com o antecedente mínimo e o conseqüente máximo, chamadas de minimal non-redundant association rules, consideradas as mais úteis e relevantes. Utiliza o conceito de bases e de fechamento de conexão de Galois.

4.4 USO DE TAXONOMIA NA REDUÇÃO DA CARDINALIDADE DO CONJUNTO DE REGRAS

Segundo o dicionário Merriam-Webster, taxonomia é uma classificação ordenada das plantas e dos animais, de acordo com seus supostos relacionamentos naturais. Um exemplo claro de taxonomia é a do reino animal. O enquadramento de um lobo cinzento comum seria: reino "animais", classe "mamíferos", ordem "carnívoros", gênero "canis", espécie "lupus canis". Outros membros do gênero "canis" são cachorros e chacais, em uma taxonomia baseada na suposta relação "é um tipo de". Uma taxonomia pode ser melhor descrita como uma hierarquia, criada de acordo com o conteúdo de seus itens, que indicam como podem ser hierarquicamente classificados [ADA01].

Na figura 4.3, um pequeno exemplo de uma taxonomia [SRI95, em que se pode verificar que: paletó é um sobretudo, calça de esqui é sobretudo, sobretudo é um tipo de roupa, camisa é um tipo de roupa, sapato é um tipo de calçado e bota de alpinismo é um tipo de calçado.

A informação da taxonomia pode ser utilizada para se encontrar associações entre itens, presentes em diferentes níveis da hierarquia, e não apenas entre os

contidos no nível inferior da taxonomia, como ocorre em regras de associação que não se utilizam de taxonomias [SRI95]. Por exemplo, a partir da taxonomia da figura 4.3, pode-se inferir uma regra do tipo: "pessoas que compram calçados tendem a comprar paletós", sem necessidade de utilizar as regras: "pessoas que compram sapatos também compram paletós" e "pessoas que compram botas de esqui também compram paletós".

A incorporação da taxonomia ao conjunto de regras de associação mineradas tem diversas abordagens, destacando-se [SRI95] e [HAN95]. Na abordagem de Srikant e Agrawal, a entrada de dados consiste no conjunto de transações da base de dados, na taxonomia e nos parâmetros únicos, suporte mínimo (minsup) e confiança mínima (minconf). Não existe pré-processamento, e o processamento incorpora a taxonomia, através das transações estendidas na geração de itemsets candidatos a freqüentes, utilizando o método Apriori.

Na abordagem de Han e Fu, a entrada de dados consiste no conjunto de transações da base de dados (sales_transaction table), na taxonomia (sales_item relation) e nos parâmetros suporte mínimo (minsup), para cada nível da taxonomia. Em uma etapa de pré-processamento, se faz a geração da descrição generalizada dos itens (generalizes item description table) e a geração das transações codificadas (encoded transaction table). A etapa de processamento utiliza algoritmo próprio, na determinação dos itemsets freqüentes (large itemsets).

O enfoque da presente pesquisa é o enriquecimento da mineração pela incorporação do conhecimento de mundo e a sua utilização para reduzir a cardinalidade do conjunto de regras mineradas. As duas abordagens fazem o enriquecimento das regras, e nosso trabalho de redução será feito na etapa de pós-processamento.

A utilização das informações contidas nas taxonomias é bastante interessante, pois podem ser descobertas regras interessantes, com classes de itens que, individualmente, não têm o suporte mínimo especificado. Além disso, a taxonomia ajuda a identificar regras redundantes. Em [SRI95], há dois algoritmos para encontrar associações entre itens presentes em diferentes níveis da hierarquia taxonômica. Esses algoritmos geram as regras gerais de associação, que possuem suporte e confiança maiores do que os valores mínimos especificados. Definem uma

regra de associação geral, como uma implicação da forma X $\Rightarrow$ Y, para X e Y conjuntos disjuntos de itens; X e Y contêm itens em qualquer nível da hierarquia, e nenhum item de Y é ancestral de qualquer item de X. Para eliminar as regras redundantes, utiliza-se uma medida de interesse baseada na taxonomia, considerando-se regra redundante aquela que não apresenta informação adicional em relação a sua forma mais geral (regra ancestral).

Por exemplo, considere-se a taxonomia apresentada na figura 4.3, na qual "Calçados" é pai de "Sapatos", e a seguinte regra foi minerada: Calçados $\Rightarrow$ Paletós (8% Sup, 70% Conf). Se cerca de um quarto das vendas de calçados é de sapatos, esperamos que a regra (Sapatos $\Rightarrow$ Paletós) tenha 2% de Suporte e 70% de Confiança. Se o Suporte e a Confiança são próximos dos valores esperados, a regra é considerada redundante, já que não possui um comportamento significativamente diferente de sua generalização. A partir dessa noção, pode-se dizer que uma regra é interessante, se a mesma possui valores de Suporte ou Confiança maiores ou iguais δ vezes o valor de um Suporte ou de uma Confiança esperada, para algum valor δ especificado pelo usuário.

Outra abordagem, que emprega as taxonomias para reduzir o volume de regras extraídas, encontra-se em [DOM04a].

Figura 4.3 Taxonomia de vestuário

Considere-se novamente a taxonomia apresentada na figura 4.3, na qual "Calçados" é pai de "Sapatos" e "Botas de esqui", e as seguintes regras de associação foram mineradas:

Sapatos $\Rightarrow$ Paletós

Botas de esqui $\Rightarrow$ Paletós

Uma vez que a associação com o item "Paletó" está presente em todos os filhos do item "Calçados", o método proposto por Domingues [DOM04a] substitui as regras referentes a "Sapatos" e "Botas de esqui" pela regra mais geral: Calçados $\Rightarrow$ Paletós .

Na mesma direção, o método ADDCUT [BUR06] utiliza a generalização e especialização das regras de associação para a poda de regras redundantes.

Srikant e Agrawal [SRI95] escolhem o valor do interesse mínimo, de maneira subjetiva (de acordo com a experiência do usuário), o que não garante uma grande redução do volume de regras. Além disso, o problema da redundância semântica das regras não é enfocado, pois, nem sempre se eliminam todas as regras mais específicas, e não se leva em consideração o processo de especialização (a regra mais geral eliminada em favor das mais específicas). O método proposto por Domingues [DOM04a] só generaliza, deterministicamente, e não, probabilisticamente. Para efetuar a generalização, todas as especializações devem estar presentes no conjunto de regras mineradas, sem considerar os casos intermediários, mais comuns no dia a dia.

5 MODELO SEMPRUNE

O modelo proposto, SemPrune, é a contribuição desta tese, incorporando áreas de conhecimento usualmente menos envolvidas na tarefa de mineração de regras de associação.

A metodologia clássica utilizada compreendeu desenvolvimento teórico e um experimento controlado.

Inicialmente, verificou-se a necessidade de tratar a redução do resultado da mineração de dados em duas vertentes distintas. A descoberta de relacionamentos, entre elementos do domínio, permite explorar a redundância decorrente.

O modelo SemPrune compreende a utilização do conhecimento de mundo, das ontologias, para a extração de regras não mineradas pelos métodos tradicionais, bem como o uso da redundância, para a criação de métodos de redução do número de regras de associação geradas pela mineração de dados. O conceito de redundância foi formalmente definido nos itens 4.1.2 e 4.3.1. Como conseqüência da caracterização da redundância, determinam-se as dependências entre atributos e/ou eventos. Uma vez definida a acepção de redundância, em cada caso, o modelo comportou a criação de métricas de redundância. O estudo dessas métricas forneceu patamares que definem o nível de corte de cada regra específica. Finalmente, o modelo SemPrune comporta um algoritmo de filtragem das regras, segundo os conceitos de redundância definidos e as métricas calculadas, e comporta duas versões. Na primeira, o enriquecimento das regras de associação ocorre no pós-processamento, e na segunda, no pré-processamento. A figura 5.1 mostra um esquema do modelo, com enriquecimento no pós-processamento, que apresenta três módulos:

- Gerador de regras mais abrangentes
- Analisador de relevância de regras
- Podador de regras menos relevantes

O usuário especifica a ontologia do domínio a adotar e os parâmetros de filtragem. Quando for necessária a introdução do conhecimento de mundo na etapa de pré-processamento as regras mais gerais já devem existir, e o gerador de regras

mais abrangentes as encontrará pela função gera-regras-gerais. Quando houver possibilidade de introduzir o conhecimento de mundo na etapa de pós-processamento, as regras mais gerais ainda não existirão e as mais abrangentes serão geradas pela função enriquecimento.

O analisador de relevância de regras calcula as métricas que determinarão a opção entre generalização e especialização de regras.

O podador de regras menos relevantes, à luz dos parâmetros de filtragem e dos valores das métricas, fornecidas pelo analisador de relevância de regras, cria o conjunto de regras filtradas, eliminado as podadas.

A figura 5.2 mostra um esquema do modelo, com enriquecimento no pré-processamento. A diferença em relação ao caso anterior é que são necessários dois módulos adicionais: o primeiro deles executa o enriquecimento da base de dados de transações, inserindo um atributo dependente em cada transação em que apareçam seus atributos determinantes; o segundo executa a mineração das regras de associação, na base de transações enriquecida. Depois deste ponto as duas versões coincidem.

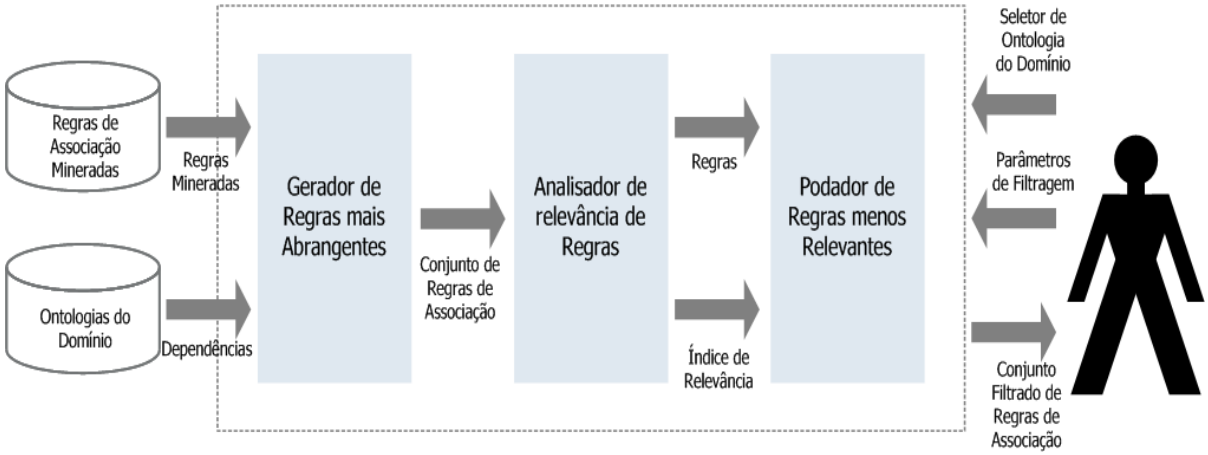


Figura 5.1 Modelo SemPrune com enriquecimento no pós-processamento

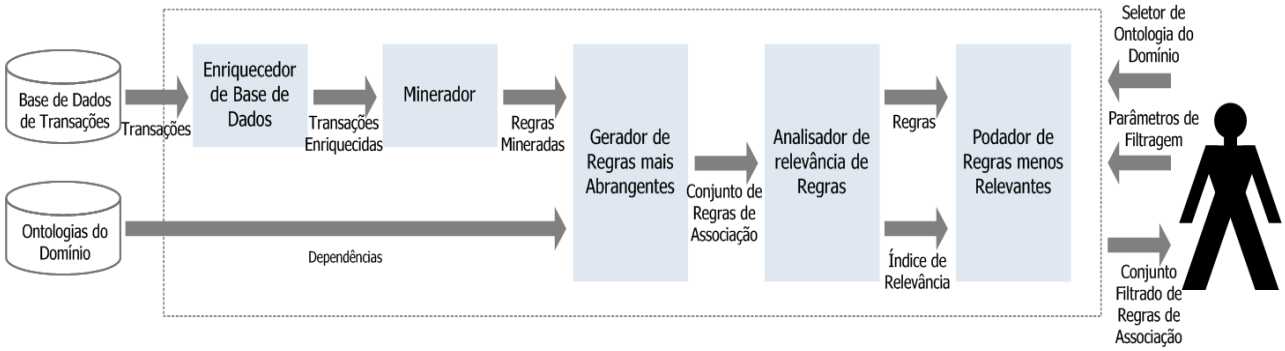


Figura 5.2 Modelo SemPrune com enriquecimento no pré-processamento

5.1 VISÃO GERAL

Uma relação de dependência é definida como o relacionamento entre dois conjuntos de atributos, no que diz respeito à determinada base de dados. Considere-se D uma base de dados, e X e Y subconjuntos dos atributos de D, sendo Y um conjunto unitário (de um único elemento). Dizemos que Y é dependente de X (ou que X determina Y), se, e somente se, existe uma função f: Domínio(X) $\Rightarrow$ Domínio(Y), válida para todo valor possível de D. A relação de dependência de Y por X será representada por: X $\Rightarrow$ Y. Chamaremos o atributo do subconjunto Y de atributo dependente, e os atributos do subconjunto X de atributos determinantes.

5.1.1 Enriquecimento dos resultados da mineração e filtragem semântica

O enriquecimento dos resultados da mineração através do conhecimento de mundo, tem características distintas, para cada tipo de relacionamento entre os dados.

O relacionamento com enriquecimento das regras é feito por pré-processamento, e nesta fase são incluídas agregações, faixas e classificações instintivas, como atributos novos da base de transações. A seguir, ocorre a

mineração de regras, gerando um número maior delas. A filtragem semântica é feita com o uso das métricas semânticas.

Quando for possível a mineração, sem agregação do conhecimento de mundo, e sem perda de regras relevantes - embora de pouco suporte - , ocorrerão, na etapa de pós-processamento, a incorporação do conhecimento de mundo e a filtragem semântica.

Para o relacionamento do tipo C, ou relacionamento "parte-de" m:n, na etapa de pré-processamento, o enriquecimento das regras exige a seleção das múltiplas dependências de ancestrais, a considerar em cada contexto, e a partir daí, o problema recai no tipo B.

Para o relacionamento do tipo D, ou relacionamento "parte-de" 1:n, o tratamento não necessita ampliar a etapa de pré-processamento, sendo igual ao tratamento do tipo B.

O enriquecimento no pós-processamento é, evidentemente, mais vantajoso, uma vez que o conjunto de regras de associação mineradas enriquece e sofre redução da cardinalidade, em uma só operação. Todavia, quando o processo de agregação é 1:n o enriquecimento poderia se perder, caso fosse deixado para o pós-processamento, pois poderiam não ser minerados baixos valores de suporte dos filhos de um nó agregador, e o agregador e seus filhos não apareceriam na mineração.

Outra característica é que o enriquecimento, no pré-processamento, cria redundâncias internas a regras e isto vai obrigar que estas redundâncias sejam tratadas. No item 5.2.1.1, serão identificados quatro tipos de redundâncias, sendo que as três primeiras só vão ocorrer quando o enriquecimento for executado no pré-processamento.

5.2 PÓS-PROCESSAMENTO DE REGRAS DE ASSOCIAÇÃO

O conjunto de regras de associação e de padrões seqüenciais, gerados no processo de mineração, costuma ser muito grande, e o trabalho de busca de

informação relevante, nova e interessante, sofre apenas uma mudança de escala, extraindo milhares de regras ou padrões de milhões de registros. Para tornar este conjunto mais adequado para os especialistas, que deverão fazer a interpretação dos resultados minerados, é fundamental efetuar operações de tratamento, filtragem e poda do resultado inicial da mineração. Estas transformações recebem o nome de pós-processamento. Esta pesquisa direcionou o pós-processamento, quando possível e necessário, para o enriquecimento semântico do conjunto de regras de associação mineradas, e sempre, para a redução da cardinalidade dos conjuntos de informações obtidos na mineração de dados. A filtragem semântica será feita com o uso das métricas definidas a seguir. Podem ser usadas métricas baseadas nos índices objetivos, cobertura e confiança, e as que utilizam a hipótese de distribuição uniforme dos suportes das regras, contendo atributos dependentes, que tenham pai comum, como descrito a seguir.

5.2.1 Pós-processamento com enriquecimento no pós-processamento

O método proposto a seguir tem como objetivo enriquecer as regras de associação mineradas e, ao mesmo tempo, eliminar as de associação redundantes, ou de menor impacto, com base nas relações de dependência entre os atributos.
A poda de regras só ocorrerá quando houver um conjunto de regras específicas que possa ser generalizado. A poda das regras generalizadas nunca vai afetar a cardinalidade do conjunto interseção de informações relevantes e recuperadas. Esta cardinalidade, não afetada, é o numerador da expressão da precisão da recuperação da informação. Só o denominador, cardinalidade do conjunto de regras recuperadas, é que vai ser reduzido.

O item 5.2.1.1 apresenta um formalismo baseado na lógica de predicados, para motivar e apresentar o método proposto. O mesmo item aborda a generalização e especialização das regras de associação, além de apresentar uma declaração formal do problema. O item 5.2.1.2 é dedicado à apresentação do algoritmo de poda

das regras redundantes. No item 5.2.1.3, são apresentadas as considerações finais a respeito do método.

Convém salientar a confusão idiomática proveniente do uso simultâneo do jargão das monografias e o da descoberta de conhecimento em bases de dados. Item é aqui referenciado como um trecho da tese, tal como "item 5.2.1.1 Formalização", mas também é um elemento de uma transação de KDD, definida no "item 2.1 regras de associação". Apenas o contexto vai indicar a acepção correta.

5.2.1.1 Formalização

Seja D uma relação pertencente a uma base de dados multidimensional. Considere-se x e y dois atributos de D. Seja a dependência de x e y dada por f: Domínio(x) $\Rightarrow$ Domínio(y). Considere-se y um elemento do domínio de Y, e x um elemento do domínio de X, tal que f($\overline{\mathbf{x}}$) = y. Considere-se X e Y dois conjuntos de condições definidas sobre os atributos x e y de D, respectivamente. Considere-se P(h, A) o predicado: a instância h satisfaz as condições do conjunto A, que é um conjunto de condições definidas sobre atributos de D. Uma regra de associação $A \Rightarrow B$ pode ser representada como uma implicação de probabilidade p: $P(h, A) \overset{p}{\Rightarrow} P(h, B)$, para A e B, conjuntos disjuntos de condições definidas sobre atributos de D. Para cada instância que satisfaça as condições de A, espera-se que ela também satisfaça as condições de B, com uma probabilidade *p*. Havendo relação de dependência entre os atributos x e y, ou, entre X e Y, se uma instância satisfaz a condição do conjunto X , esta instância também satisfaz a condição do conjunto Y, ou: $P(h, X) \Rightarrow P(h, Y)$.

As redundâncias entre regras foram pesquisadas a partir da premissa de que sempre existem dependências entre atributos e regras, porque algumas regras podem ter, no antecedente ou no conseqüente, atributos determinantes e atributos determinados. A redundância entre elas pode ser de quatro tipos:

- o antecedente da regra possui atributos determinantes e o conseqüente, atributo dependente (redundante com a premissa);
- uma regra possui, no antecedente, atributos determinantes, e outra, com o mesmo conseqüente, possui, no antecedente, atributos determinantes mais o atributo dependente (segunda regra redundante com a primeira);
- uma regra possui, no antecedente, atributos determinantes, e outra, no conseqüente, o atributo dependente mais o conseqüente da primeira regra (segunda regra redundante com a primeira);
- uma regra possui, como parte do antecedente, um atributo dependente, e outra tem, como única diferença, a substituição do atributo dependente pelos seus atributos determinantes (segunda regra redundante com a primeira).

Definição 5.2.1.1.1 (Redundância das regras na forma 1):

Se existe uma regra da forma X ^ C $\Rightarrow$ Y, esta é redundante, e C é um conjunto de condições definidas sobre atributos de D. A justificativa vem da lógica de primeira ordem, onde: uma regra de associação que apresenta a forma da expressão X ^ C $\Rightarrow$ Y, (5.2.1.1) expressa, logicamente, por P(h, X) ^ P(h, C) $\Rightarrow$ P(h, Y), pode ser deduzida a partir da relação de dependência existente entre os conjuntos X e Y, por meio da monotonicidade do encadeamento de Gentzen [SZA69], da forma que se segue:

i.	X $\Rightarrow$Y	Premissa
ii.	X ^ C $\Rightarrow$Y	Monotonicidade do encadeamento

Exemplo 5.2.1.1.1:

Considere-se a relação de dependência entre os atributos **BMI** e **Status**, {BMI} $\overset{1}{\Rightarrow}$ {Status}, definida pela função f: Domínio(BMI) $\Rightarrow$ Domínio(Status), tal que f(24) = "normal". Uma regra de associação do tipo (BMI=24) ^ (Idade=50) ^ (Sexo="F") $\Rightarrow$ (Status="normal"), que apresenta a forma da expressão 5.2.1.1, é redundante, porque pode ser deduzida da relação de dependência {BMI} $\overset{1}{\Rightarrow}$

{Status}, associando X a (BMI=24), C a (idade=50) ^ (Sexo="F") e Y a (status="normal") na definição 5.2.1.1.1.

Desta forma, podemos concluir que as regras que apresentam a forma da expressão 5.2.1.1 podem ser eliminadas, uma vez que representam uma relação previamente conhecida dos dados - as relações de dependência entre os atributos.

Definição 5.2.1.1.2 (Redundância das regras na forma 2):

Se existem duas regras das formas R_1: X ^ A $\Rightarrow$ B e R_2: X ^ Y ^ A $\Rightarrow$ B, então a regra R_2 é redundante, sendo A e B conjuntos de condições definidas sobre atributos de D. A justificativa vem da lógica de primeira ordem, onde: uma regra de associação que apresenta a forma da expressão:

X ^ Y ^ A $\Rightarrow$ B, (5.2.1.2)

pode ser deduzida, por meio da monotonicidade do encadeamento de Gentzen [SZA69], e da regra da mão esquerda de Girard [GIR87], como se segue:

i. X ^ A $\Rightarrow$ B — Premissa

ii. X ^ A ^ Y $\Rightarrow$ B — Monotonicidade do encadeamento

iii. X ^ Y ^ A $\Rightarrow$ B — Regra da mão esquerda de Girard

Exemplo 5.2.1.1.2:

Considere-se a relação de dependência entre os atributos **BMI** e **Status,** definida no exemplo 5.2.1.1.1 e as seguintes regras mineradas:

R_1 : (BMI=24) ^ (Idade=50) $\Rightarrow$ (Seguro-saúde="sim") e

R_2 : (BMI=24) ^ (Status="normal") ^ (Idade=50) $\Rightarrow$ (Seguro-saúde="sim")

A regra R_2, que apresenta a forma da expressão 5.2.1.2, e que pode ser expressa logicamente pela implicação P(h, {BMI=24}) ^ P(h, {Status="normal"}) ^ P(h, {Idade=50}) $\Rightarrow$ P(h, {Seguro-saúde="sim"}), é redundante, porque pode ser deduzida a partir da regra R_1 e da relação de dependência {BMI} $\overset{1}{\Rightarrow}$ {Status}, associando X a (BMI=24), A a (Idade=50), B a (Seguro-saúde="sim") e Y a (Status="normal") na definição 5.2.1.1.2:

A regra R_2 pode, então, ser descartada.

Definição 5.2.1.1.3 (Redundância das regras na forma 3):

Se existem duas regras das formas R_1: X ^ A $\Rightarrow$ B e R2: X ^ A $\Rightarrow$ Y ^ B, então a regra R2 é redundante. A justificativa vem da lógica de primeira ordem, onde uma expressão da forma X ^ A $\Rightarrow$ Y ^ B, (5.2.1.3) pode ser deduzida a partir de do enfraquecimento (sinônimo de monotonicidade), e pela regra da mão direita [GIR87] para o "and":

i.	X ^ A $\Rightarrow$ B	Premissa
ii.	X $\Rightarrow$ Y	Premissa
iii.	X ^ A $\Rightarrow$ Y	Monotonicidade do encadeamento
iv.	X ^ A $\Rightarrow$ Y ^ B	i, ii e regra da mão direita

A regra R_2 pode, então, ser descartada.

Exemplo 5.2.1.1.3:

Considere-se a relação de dependência entre os atributos **BMI** e **Status,** definida no exemplo 5.2.1.1.1 e as seguintes regras mineradas:

R_1 : (BMI=24) ^ (Idade=50) $\Rightarrow$ (Seguro-saúde="sim") e

R_2 : (BMI=24) ^ (Idade=50) $\Rightarrow$ (Status="normal") ^ (Seguro-saúde="sim")

A regra R_2, que apresenta a forma da expressão 5.2.1.3, e que pode ser expressa, logicamente, pela implicação P(h, {BMI=24}) ^ P(h, {Idade=50}) $\Rightarrow$ P(h, {(Status="normal")}) ^ P(h, {Seguro-saúde="sim"}), é redundante ,porque pode ser deduzida a partir da regra R_1 e da relação de dependência {BMI} $\overset{1}{\Rightarrow}$ {Status}, associando X a (BMI=24), A a (Idade=50), B a (Seguro-saúde="sim") e Y a (Status="normal") na definição 5.2.1.1.3:

A regra R_2 pode, então, ser descartada.

Definição 5.2.1.1.4 (Redundância de regras da forma 4):

Se existem duas regras das formas R_1: Y ^ A $\Rightarrow$ B e R_2: X ^ A $\Rightarrow$ B, a regra R_2 é redundante. A justificativa vem da lógica de primeira ordem. Considere-se a relação de dependência existente entre os atributos x e y, uma regra de associação da forma da expressão: Y ^ A $\Rightarrow$ B, (5.2.1.4).

Pode-se provar que esta expressão é equivalente à expressão X ^ A $\Rightarrow$ B (5.2.1.5) usando as regras de Gentzen como cálculo matemático, da forma se ($p \Rightarrow q$),($q \Rightarrow r$) então ($p \Rightarrow r$) [SZA69]

i.	X $\Rightarrow$Y	Premissa
ii.	Y ^ A $\Rightarrow$ B	Premissa
iii.	X ^ A $\Rightarrow$ B	Regra de Gentzen como cálculo

Exemplo 5.2.1.1.4:

Considere-se a relação de dependência entre os atributos **BMI** e **Status,** definida no exemplo 5.2.1.1.1 e as seguintes regras mineradas:

R_1 : (Status="normal") ^ (Idade=50) $\Rightarrow$ (Seguro-saúde="sim") e

R_2 : (BMI=24) ^ (Idade=50) ((Seguro-saúde="sim")

A regra R2, que apresenta a forma da expressão 5.2.1.5, e que pode ser expressa logicamente pela implicação P(h, {BMI=24}) ^ P(h, {Idade=50}) (P(h, {Seguro-saúde="sim"}), é redundante, porque pode ser deduzida a partir da regra R1 e da relação de dependência {BMI} $\overset{1}{\Rightarrow}$ {Status}, associando Y a (Status="normal"), A a (Idade=50), B a (Seguro-saúde="sim") e X a (BMI=24), na definição 5.2.1.1.4:

A regra R2 pode, então, ser descartada.

5.2.1.1.1 Generalização e especialização das regras

Comparando as expressões 5.2.1.4 e 5.2.1.5, observa-se uma redundância semântica, a primeira mais geral e a segunda mais específica. Pode-se reduzir esta redundância por processos de generalização e especialização das regras. No processo de generalização, a regra mais geral é considerada mais relevante, de maior impacto que as regras mais específicas, que são eliminadas, restando a regra mais geral. Quando as regras mais específicas apresentam maior relevância do que a regra mais geral, que é descartada, ocorre especialização Na generalização,

valorizamos a capacidade de síntese das relações descobertas, ao passo que na especialização, valorizamos a capacidade discriminatória das regras.

A escolha entre generalização (preferencial) e especialização é feita utilizando-se, inicialmente, a medida de interesse cobertura, medida tradicional de aferição da generalidade de uma regra. A métrica escolhida é a razão entre o somatório das coberturas dos antecedentes das regras específicas e a cobertura do antecedente da regra geral. Este valor não é unitário, pois, embora haja função determinística entre atributo determinante (das regras específicas) e atributo dependente (da regra geral), a co-ocorrência só existe entre atributos e conseqüente da regra. Como as singularidades são características de atributos independentes, e não de sua agregação, a cobertura do antecedente da regra geral é sempre maior e a métrica, menor que 1,0. Para verificar se regras gerais podem substituir diversas específicas, observar se as regras gerais proporcionam cobertura suficiente sobre todas as específicas. Neste caso, as regras específicas podem ser podadas Caso contrário, existem regras específicas singulares, que não se enquadram na regra geral e não podem ser podadas. É preciso, então, saber se a regra geral não pode ser representativa das específicas não singulares. Se a regra geral cobrir suficientemente as regras específicas não singulares, estas últimas serão podadas. Caso contrário, a regra geral é a que não traz nenhum valor agregado, e pode ser podada.

A verificação do valor relativo entre a regra geral e as regras específicas não singulares é feita com o uso da confiança da regra geral, na ausência das regras específicas. Alcançando nível mínimo estipulado de confiança, a regra geral ainda pode substituir as regras específicas não singulares. Caso contrário, a regra geral é a que não traz nenhum valor agregado e pode ser podada.

De maneira geral, quando o comportamento descrito pela regra mais geral independe dos atributos determinantes, ou melhor, quando descreve um comportamento uniforme da população, consideramos a regra mais geral de maior relevância. No caso de uma população desbalanceada, na qual o comportamento descrito pela regra mais geral não é uniforme para os valores possíveis dos atributos determinantes, as regras mais específicas são consideradas as mais relevantes. Segue-se um cenário de escolha entre os processos de generalização e especialização das regras.

Exemplo 5.2.1.1.5:

Dado um banco de dados com informações sobre pacientes de uma clínica, considere-se a relação de dependência entre os atributos BMI (Índice de Massa Corporal) e Status: {BMI} $\Rightarrow$ {Status}, definida por f : Domínio({BMI}) $\Rightarrow$ Domínio({Status}), tal que: f(38)="obeso".

Considere-se as seguintes regras mineradas:

R1: (Status="obeso") $\Rightarrow$ (Pressão-sangüínea="alta")

R2: (BMI=37) $\Rightarrow$ (Pressão-sangüínea="alta")

R3: (BMI=38) $\Rightarrow$ (Pressão-sangüínea="alta")

Neste exemplo, a regra mais geral, representada pela R1, descreve uma associação entre os indivíduos de status obeso e a pressão sangüínea alta. As regras R2 e R3 descrevem a mesma associação da regra R1 para os indivíduos de BMI igual a 37 e 38, respectivamente. Se o comportamento descrito pela regra R1 for uniforme, em relação ao BMI, ou seja, os indivíduos obesos têm a pressão alta, não importando o valor do BMI, então aplicamos o processo de generalização. No caso de uma população desbalanceada, na qual, dos indivíduos obesos, apenas aqueles de BMI 37 e 38 têm a pressão alta, então aplicamos o processo de especialização.

A noção do quanto um comportamento é considerado uniforme será dada pelas medidas CRg e CRm [BUR06], descritas a seguir.

O indicador CRg mede a generalidade da regra mais geral, com relação aos atributos que determinam o atributo dependente. Esta medida é baseada na medida de interesse cobertura [LAV99], que representa a fração das instâncias cobertas pelo antecedente da regra, e pode ser considerada como medida de generalidade da regra. O valor da cobertura de uma regra é dado pelo suporte do antecedente dessa regra.

Definição 5.2.1.1.5 (CRg):

Seja D uma base de dados multidimensional. Seja R_i uma regra de associação da forma da expressão 5.2.1.4 (Y ^ A $\Rightarrow$ B), e $S_i = \{r_{ij} \mid j = 1..n\}$ o conjunto de regras correspondentes na forma da expressão 5.2.1.5 (X ^ A $\Rightarrow$ B), obtidas a partir de D.

O valor da medida CRg para R_i e S_i é dado por:

$$CRg(R_i, S_i) = \frac{\sum_{j=1..n} \text{suporte}(antecedente(r_{ij}))}{\text{suporte}(antecedente(R_i)} \qquad (5.2.1.6)$$

Quanto maior for a medida CRg, maior a representatividade das instâncias cobertas pelas regras mais específicas com relação às instâncias cobertas pela regra mais geral, o que significa um comportamento uniforme da população. Esta medida varia entre 0 e 1. A medida CRg pode ser interpretada como a probabilidade condicional de uma instância satisfazer o antecedente de uma das regras mais específicas, dado que a instância satisfaz o antecedente da regra mais geral. Utilizando o exemplo 5.2.1.1.5, apresentado no início deste item, se o valor calculado para a medida CRg for alto (maior que um valor mínimo especificado pelo usuário), significa que a maior parte dos indivíduos obesos tem o BMI igual a 37 ou a 38. Desta forma, o comportamento descrito pela regra mais geral é considerado uniforme em relação ao BMI, uma vez que os valores de BMI 37 e 38 também estão associados à pressão sangüínea alta. O valor da medida CRg pode ser calculado a partir dos valores de suporte e confiança da regra mais geral e das regras mais específicas. Para tanto, utilizamos a fórmula da medida confiança para calcular o suporte do antecedente de uma regra R, como:

$$\text{sup}\, orte(antecedente(R)) = \frac{\text{suporte}(R)}{confiança(R)} \qquad (5.2.1.7)$$

Substituindo o suporte do antecedente das regras na fórmula 5.2.1.6, de acordo com a expressão 5.2.1.7, a medida CRg é dada por:

$$CRg(R_i, S) = \frac{\left(\sum_{j=1..n} \frac{\text{suporte}(r_{ij})}{confiança(r_{ij})} \right)}{\frac{\text{suporte}(R_i)}{confiança(R_i)}} \qquad (5.2.1.8)$$

Contudo, a existência de algumas regras específicas singulares não invalida totalmente a regra mais geral. É preciso verificar se a regra mais geral continua

válida, considerando apenas as instâncias não cobertas pelas regras mais específicas, o que é feito pela medida CRm. Esta medida é utilizada quando o valor da medida CRg estiver abaixo do valor mínimo especificado pelo usuário. Esta medida é baseada na medida confiança, e a mesma fórmula é utilizada, com a inclusão da restrição das instâncias consideradas no cálculo da confiança da regra mais geral, considerando apenas aquelas não cobertas pelas regras mais específicas singulares, de alto suporte.

Definição 5.2.1.1.6 (CRm):

Seja D uma base de dados multidimensional. Seja R_i uma regra de associação da forma da expressão 5.2.1.4 (Y ^ A $\Rightarrow$ B), e $S_i = \{r_{ij} \mid j = 1..n\}$ o conjunto de regras correspondentes na forma da expressão 5.2.1.5 (X ^ A $\Rightarrow$ B), obtidas a partir de D.

O valor da confiança para R_i e S_i é dado por:

$$CRm(R_i, S_i) = \frac{\text{suporte}\left(A \cap Y_i \cap B \cap \left(\overline{\cup X_{ij} \cap A}\right)\right)}{\text{suporte}\left(A \cap Y_i \cap \left(\overline{\cup X_{ij} \cap A}\right)\right)} \qquad (5.2.1.9)$$

onde a condição $\left(\overline{\cup X_{ij} \cap A}\right)$ representa a restrição da não consideração das instâncias cobertas pelas regras mais específicas singulares. Se a regra mais geral continuar válida com esta restrição, ou seja, se o valor da medida CRm for acima do valor mínimo especificado pelo usuário, significa que o comportamento da população é uniforme, uma vez que o comportamento descrito pela regra mais geral é válido para quaisquer valores dos atributos que determinam o atributo dependente. Nestas condições, a regra mais geral ainda pode substituir as mais específicas. Se nem CRg nem CRm atingirem o patamar especificado, CRgMin, então a regra mais geral não tem serventia. A medida CRm varia entre 0 e 1. Em sua expressão, podemos substituir a condição $\left(\overline{\cup X_{ij} \cap A}\right)$ pela condição equivalente: $\left(\overline{\cup X_{ij}} \cup \overline{A}\right)$. A conjunção dessa condição com o conjunto A pode ser calculada como:

$$A \cap \left(\overline{\cup X_{ij}} \cup \overline{A}\right) = \left(A \cap \overline{\cup X_{ij}}\right) \cup \left(A \cap \overline{A}\right) = A \cap \overline{\cup X_{ij}}$$

Substituindo a fórmula acima, na expressão 5.2.1.9, temos que:

$$CRm(R_i, S_i) = \frac{\text{suporte}\left(A \cap Y_i \cap B \cap \overline{\cup X_{ij}}\right)}{\text{suporte}\left(A \cap Y_i \cap \overline{\cup X_{ij}}\right)}$$

Os suportes da expressão acima podem ser calculados da seguinte maneira:

$\sup orte\left(A \cap Y_i \cap B \cap \overline{\cup X_{ij}}\right) = \sup orte\left(A \cap Y_i \cap B\right) - \sup orte\left(A \cap Y_i \cap B \cap \cup X_{ij}\right)$ e

$\sup orte\left(A \cap Y_i \cap \overline{\cup X_{ij}}\right) = \sup orte\left(A \cap Y_i\right) - \sup orte\left(A \cap Y_i \cap \cup X_{ij}\right)$

Nas expressões acima, podemos substituir o termo $Y_i \cap \cup X_{ij}$ pela união dos conjuntos $X_{ij} (\cup X_{ij})$, uma vez que: $\cup X_{ij} \subseteq Y_i$, e logo, $Y_i \cap \cup X_{ij} = \cup X_{ij}$. Desta forma, podemos calcular os suportes por:

$\sup orte\left(A \cap Y_i \cap B \cap \cup X_{ij}\right) = \sup orte\left(A \cap B \cap \cup X_{ij}\right)$ e

$\sup orte\left(A \cap Y_i \cap \cup X_{ij}\right) = \sup orte\left(A \cap \cup X_{ij}\right)$

Substituindo o cálculo dos suportes acima, na fórmula da CRm, temos que:

$$CRm(R_i, S_i) = \frac{\text{suporte}\left(A \cap Y_i \cap B\right) - \text{suporte}\left(A \cap B \cap X_{ij}\right)}{\text{suporte}\left(A \cap Y_i\right) - \text{suporte}\left(A \cap \cup X_{ij}\right)}$$

Das definições das medidas suporte e confiança, podemos deduzir que:

$\sup orte\left(A \cap Y_i \cap B\right) = \text{suporte}\left(R_i\right)$,

$\sup orte\left(A \cap B \cap \cup X_{ij}\right) = \sum_{j=1..n} \text{suporte}\left(r_{ij}\right)$,

$\sup orte\left(A \cap Y_i\right) = \text{suporte}\left(antecedente(R_i)\right)$ e

$\sup orte\left(A \cap \cup X_{ij}\right) = \sum_{j=1..n} \text{suporte}\left(antecedente(r_{ij})\right)$

Substituindo os termos acima na expressão da CRm, temos que:

$$CRm(R_i, S_i) = \frac{\text{suporte}\left(R_i\right) - \sum_{j=1..n} \text{suporte}\left(r_{ij}\right)}{\text{suporte}\left(antecedente(R_i)\right) - \sum_{j=1..n} \text{suporte}\left(antecedente(r_{ij})\right)}$$

Assim como a medida CRg, o valor da CRm também pode ser calculado a partir dos valores de suporte e confiança da regra mais geral e das regras mais específicas. Para tanto, substituímos o suporte dos antecedentes na expressão da medida CRm, de acordo com a fórmula 5.2.1.7. Desta forma, a medida CRm é calculada como:

$$CRm(R_i, S_i) = \frac{\text{suporte}(R_i) - \sum_{j=1..n} \text{suporte}(r_{ij})}{\frac{\text{suporte}(R_i)}{confiança(R_i)} - \sum_{j=1..n} \frac{\text{suporte}(r_{ij})}{confiança(r_{ij})}} \qquad (5.2.1.10)$$

5.2.1.1.2 Declaração formal do problema

A seguir é apresentada uma declaração formal do problema do pós-processamento de regras de associação, com base nas relações de dependência entre os atributos.

Definição 5.2.1.1.7 (Problema do pós-processamento de regras de associação):

Seja R um conjunto de regras de associação, obtidas a partir de uma base de dados multidimensional D. Seja RD o conjunto das relações de dependência entre os atributos da forma $X \overset{n}{\Longrightarrow} Y$, para X e Y conjuntos de atributos de D (sendo Y um conjunto unitário). O problema do pós-processamento de regras de associação, com base nas relações de dependência entre os atributos consiste em:

1. Eliminar do conjunto R todas as regras de associação redundantes que apresentam as formas das expressões 5.2.1.1, 5.2.1.2 e 5.2.1.3.
2. Para cada regra R_i, da forma da expressão 5.2.1.4, e o conjunto S_i, correspondente de regras da forma da expressão 5.2.1.5, redundantes com R_i, avaliar a relevância das regras com a utilização das medidas CRg e CRm, e eliminar do conjunto R aquelas (ou aquela) de menor relevância, ou de menor impacto para o usuário.

5.2.1.2 O algoritmo

Neste item descrevemos o algoritmo, responsável por resolver o problema do pós-processamento de regras de associação, descrito no item anterior. O algoritmo 1, mostra o processo e recebe como parâmetros de entrada: um conjunto *R,* de regras de associação, um conjunto *RD,* das relações de dependência entre os atributos, o valor *GrauMax,* que indica o grau máximo de dependência entre os atributos, a ser considerado, e os valores mínimos para as medidas CRg e CRm, *CRgMin* e *CRmMin*, respectivamente. O algoritmo retorna um conjunto filtrado, por poda, das regras de associação.

Algoritmo 1

Require: <R, RD, n, CRgMin, CRmMin>

1: R' := poda-regras-1(R, RD)

2: R' := poda-regras-2(R', RD, *esquerdo*)

3: R' := poda-regras-2(R', RD, *direito*)

4: R' := poda-regras-3(R', RD)

5: $\hat{S}$:= gera-subconjuntos(R', *direito*)

6: **for all** subconjunto S tal que S $\in$ $\hat{S}$ **do**

7: G := gera-regras-gerais(S, RD)

8: **for all** r tal que r $\in$ G **do**

9: **for** n = 1 até *GrauMax* **do**

10: E := gera-regras-específicas(S, RD, r, n)

11: crg := calcula-CRg(r, E)

12: **if** crg ≥ CRgMin then

13: R' := elimina-regras(E, R') {Processo de generalização}

14: else

15: CRm := calcula-CRm(r, E)

16: if CRm ≥ CRmMin then

17: R' := elimina-regras(E, R') {Processo de generalização}

18: else
19: R’ := elimina-regras(r, R’) {Processo de especialização}
20: n := GrauMax {“Pula” para a próxima regra mais geral presente em G}
21: end if
22: end if
23: **end for**
24: **end for**
25: **end for**
26: **return** R’

Inicialmente, o conjunto R’, que irá armazenar as regras de associação filtradas, recebe o resultado da função poda-regras-1, que elimina as regras que apresentam a forma da expressão 5.2.1.1. A função poda-regras-1 é descrita pelo algoritmo 2. Em seguida, as regras que apresentam a forma da expressão 5.2.1.2, para o antecedente (lado esquerdo) e o conseqüente (lado direito) da regra, são eliminadas do conjunto R’, pela função poda-regras-2, descrita pelo algoritmo 3. No próximo passo do algoritmo, as regras que apresentam a forma da expressão 5.2.1.3 são eliminadas do conjunto R’ pela função poda-regras-3, descrita pelo algoritmo 4. Convém salientar que estas funções só podarão redundâncias herdadas do enriquecimento do conjunto de regras no pré-processamento. Quando o enriquecimento ocorrer no pós-processamento, as redundâncias das formas das expressões 5.2.1.1, 5.2.1.2 e 5.2.1.3 não serão introduzidas e a poda será desnecessária.

Em continuação, são gerados os subconjuntos presentes em R’, de regras que apresentam o mesmo conseqüente, pela função gera-subconjuntos, descrita em [DOM04b]. Para cada subconjunto S gerado, a função gera-regras-gerais, descrita no algoritmo 5, gera o conjunto G das regras mais gerais (da forma da expressão 5.2.1.4), presentes em S. Quando se tratar de problemas do tipo A (“é-um” m:n), as regras mais gerais já existem e são encontradas pela função gera-regras-gerais. Quando se tratar de problemas do tipo B (“é-um” 1:n), as regras mais gerais ainda não existem, e o conjunto G será gerado pela função enriquecimento descrita no

algoritmo 6. Para cada regra mais geral r $\in$ G, e para o grau de dependência, variando de 1 até *GrauMax*, é gerado o conjunto E das regras mais específicas redundantes com r (da forma da expressão 5.2.1.3), pela função gera-regras-específicas, descrita pelo algoritmo 7.

A partir da regra mais geral, e do conjunto correspondente das regras mais específicas, a função calcula-CRg calcula o valor da medida CRg, de acordo com a fórmula 5.2.1.8. Se o valor da medida CRg for maior ou igual ao valor mínimo especificado pelo usuário (*CRgMin*), as regras mais específicas são eliminadas do conjunto R'(processo de generalização). O processo de generalização é realizado através da função elimina-regras, que elimina do conjunto R' as regras presentes em E e retorna o resultado. Caso contrário, o valor para a medida CRm é calculado pela função calcula-CRm. A função calcula-CRm utiliza a fórmula 5.2.1.10, para calcular a medida CRm a partir da regra mais geral, e do conjunto correspondente das regras mais específicas. Se este valor for maior, ou igual ao valor mínimo especificado pelo usuário, mais uma vez aplicamos o processo de generalização das regras. Caso o valor da medida CRm seja menor que o valor mínimo especificado pelo usuário, o processo executado é o de especialização das regras. Desta vez, o conjunto R' recebe o resultado da função elimina-regras, que elimina do conjunto R' a regra r. Por fim, na linha 26, o algoritmo retorna o conjunto R'.

Os algoritmos 2, 3, 4, 5, 6 e 7 exibem funções chamadas pelo algoritmo 1, cujos nomes já são indicadores de suas funcionalidades. Julgamos assim mais conveniente exibi-los nos Apêndices.

5.2.1.3 Considerações finais

No desenvolvimento do trabalho, levou-se em consideração que a base de dados utilizada na mineração de regras de associação é representativa do domínio. O que se pretende é automatizar a eliminação semântica da redundância das regras de associação mineradas, reduzindo a quantidade e, portanto, a complexidade do

conjunto de regras de associação. O caminho utilizado baseou-se no conhecimento de mundo representado pelas relações de dependência entre os atributos.

O trabalho envolveu uma identificação dos tipos de redundância e, por meio de ações de generalização, procurou-se eliminar regras que estão cobertas por outras mais gerais. Verificou-se que existem situações, nas quais as regras mais gerais não descrevem o domínio de maneira adequada, situação na qual estas regras de associação mais gerais terão de conviver com regras especializadas.

Foram definidas duas métricas, CRg e CRm, que indicam quando vale a pena utilizar a poda de regras de associação.

O algoritmo desenvolvido permite resultados bastante satisfatórios (entre 21% e 34% de poda nos experimentos realizados), como será relatado mais adiante.

5.2.2 Pós-processamento com enriquecimento no pré-processamento

O método que se propõe, a seguir, tem como objetivo eliminar as regras de associação redundantes, ou de menor impacto, com base nas relações de dependência entre os atributos, supondo-se que o enriquecimento de informações já tenha sido realizado no pré-processamento da mineração das regras de associação.

O item 5.2.2.1 apresenta uma caracterização do problema em tela; o 5.2.2.2 é dedicado à apresentação do algoritmo de poda das regras redundantes, e no item 5.2.2.3 são apresentadas as considerações finais a respeito do método.

Em muitos casos, tal como nas taxonomias, uma entidade só pode ser colocada em um único lugar, ou seja, só pode ter um único código 'pai'. Este tipo de relacionamento é modelado em uma floresta de árvores. Como cada atributo isolado é determinante, os processos de generalização e especialização de regras se tornam de menor complexidade.

5.2.2.1 Caracterização do problema

O método adotado para a solução do problema é bastante semelhante ao utilizado na filtragem, com inclusão do conhecimento de mundo no pós-processamento. A maior diferença consiste na definição de dependência entre atributos. No caso de relacionamento do tipo B, considera-se que o descendente na taxonomia é determinante de seu ancestral, o que implica maior simplicidade no cálculo de valores de atributos dependentes para agrupamento, separação, comparação, etc. Em compensação, aumenta o número de dependências, pois dentro da hierarquia, todo atributo tem pai (logo tem dependência).

5.2.2.1.1 Generalização e especialização das regras

A singularidade de regras específicas, analisada pelas métricas CRg e CRm, utiliza regras mais gerais, inicialmente introduzidas no pré-processamento. Julgou-se adequado pesquisar métricas baseadas, exclusivamente, nas regras mais específicas, que retratassem eventual presença de regras singulares, pelo valor preponderante de seu suporte.

A escolha, entre generalização (preferencial) e especialização, pode ser feita utilizando-se a hipótese de distribuição uniforme dos suportes das regras com atributos dependentes, que possuam pai comum. A solução natural seria usar regras estatísticas, para buscar os valores fora da faixa "média" ("outliers"). Ocorre que a busca padrão de valores fora da faixa é para aqueles que estão ou muito acima ou muito abaixo do esperado. No caso em tela, as regras de associação específicas, com suporte baixo, têm mesmo de ser podadas. Usualmente, a caracterização dos "outliers" é feita por desvio maior que 1,5 ou 2,0, desvios padrões da distribuição. Experimentalmente, determinamos um desvio padrão mínimo, para considerar a inexpressividade dos "outliers".

Com os valores obtidos, calcula-se o desvio padrão da distribuição de suporte das regras específicas. Se a razão entre o desvio padrão e a média aritmética da distribuição estiver abaixo de um patamar especificado, o comportamento das regras específicas é regular, e elas podem ser substituídas pela regra geral e podadas. Caso contrário, existem regras específicas singulares, que não se enquadram na regra geral e não podem ser podadas. É preciso, então, saber se a regra geral pode ou não ser representativa das regras específicas não singulares. Se a regra geral cobrir, suficientemente, as regras específicas não singulares, estas últimas serão podadas. Caso contrário, é a regra geral que não traz nenhum valor agregado, e pode ser podada.

A noção do quanto um comportamento é considerado uniforme será dada pelas medidas TRg e TRm, criadas no presente trabalho e descritas a seguir.

Definição 5.2.2.1.1 (TRg):

Seja D uma base de dados multidimensional. Seja R_i uma regra de associação da forma da expressão 5.2.1.4, e $S_i = \{r_{ij} \mid j = 1..n\}$, o conjunto de regras correspondentes na forma da expressão 5.2.1.5, obtidas a partir de D. Seja x_k o valor do suporte de uma regra r_{ij} e μ o valor da média aritmética do suporte dessas regras. Seja σ o valor do desvio padrão do suporte dessa população de regras. O valor da medida TRg para R_i e S_i é dado pelo inverso do coeficiente de variação, ou seja:

$$\mathrm{TRg}(\mathrm{R_i}, \mathrm{S_i}) = \frac{\mu}{\sigma} \qquad (5.2.2.1)$$

onde

$$\mu = \frac{\sum_{k=1}^{n} x_k}{n} \quad \text{e} \quad \sigma = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(x_k - \mu\right)^2}$$

Quanto maior a medida TRg, menor a média dos desvios de suporte das regras específicas em relação à sua média, significando a importância relativa de regras específicas com suportes elevados (regras singulares). Quando os

afastamentos do suporte de regras forem maiores que o TrgMin, ou patamar de desvio do que se considera população regularmente uniforme, serão consideradas regras singulares aquelas que o desvio de seu suporte, em relação à média aritmética dos suportes da distribuição, dividido pelo desvio padrão da população seja maior do que α * TRgMin (sendo α um coeficiente empírico característico do domínio), ou ainda, aquelas regras para as quais vale a desigualdade

$$\frac{x_k - \mu}{\sigma} > \alpha * TrgMin \qquad (5.2.2.2)$$

A verificação do valor relativo entre a regra geral e as regras específicas não singulares é feita pelo mesmo processo de cálculo do desvio padrão dos suportes das regras específicas, na ausência das regras específicas singulares, que não podem ser podadas. Caso a razão, entre o desvio padrão e a média da distribuição, esteja abaixo do patamar especificado anteriormente, a regra geral ainda pode substituir as específicas não singulares. Caso contrário, é a regra geral que não traz nenhum valor agregado e pode ser podada.

Definição 5.2.2.1.2 (TRm):

Seja D uma base de dados multidimensional. Seja R_i uma regra de associação da forma da expressão 5.2.1.4, e $S_i = \{r_{ij} \mid j = 1..t\}$, o conjunto de regras específicas correspondentes na forma da expressão 5.2.1.5, obtidas a partir de D, para as quais não se verifica a desigualdade da expressão (5.2.2.2). Seja x_k o valor do suporte de uma regra específica não singular r_{ij}, S_i e μ_{ns}, o valor da média do suporte dessas regras não específicas. Seja σ_{ns} o valor do desvio padrão dessa população de regras. O valor da medida TRm para R_i e S_i é dado por:

$$\mathrm{TRm}(\mathrm{R_i}, \mathrm{S_i}) = \frac{\mu_{ns}}{\sigma_{ns}} \qquad (5.2.2.3)$$

onde

$$\mu_{ns} = \frac{\sum_{k=1}^{t} x_k}{t} \quad \text{e} \quad \sigma_{ns} = \sqrt{\frac{1}{t}\sum_{k=1}^{t}\left(x_k - \mu_{ns}\right)^2}$$

Sempre que TRm for menor que TRgMin, a regra geral será descartada, restando apenas as regras específicas; caso contrário, será mantida a regra geral, podadas as regras específicas.

De maneira geral, quando o comportamento descrito pela regra mais geral independe dos atributos determinantes, ou melhor, quando a regra mais geral descreve um comportamento regular da população, consideramos a regra mais geral de maior relevância. No caso de uma população desbalanceada, na qual o comportamento descrito pela regra mais geral não é uniforme para os valores possíveis dos atributos determinantes, as regras mais específicas são consideradas as mais relevantes.

5.2.2.2 O algoritmo

Neste item descrevemos o algoritmo, responsável por resolver o problema do pós-processamento de regras de associação, descrito no item anterior. O algoritmo 8 mostra o processo e recebe como parâmetros de entrada: um conjunto *R,* de regras de associação, um conjunto *RD,* das relações de dependência entre os atributos, o valor *GrauMax,* que indica o grau máximo de dependência entre os atributos, a ser considerado, e os valores mínimos para as medidas TRg e TRm, *TRgMin*, respectivamente. O algoritmo retorna um conjunto filtrado, por poda, das regras de associação.

Algoritmo 8

Require: <R, RD, n, TRgMin>

1: R' := poda-regras-1(R, RD)

2: R' := poda-regras-2(R', RD, *esquerdo*)

3: R' := poda-regras-2(R', RD, *direito*)

4: R' := poda-regras-3(R', RD)

5: $\hat{S}$:= gera-subconjuntos(R', *direito*)

6: **for all** subconjunto S tal que S $\in \hat{S}$ **do**

7: G := gera-regras-gerais-2(S, RD)

8: **for all** r tal que r $\in$ G **do**

9: **for** n = 1 até *GrauMax* **do**

10: E := gera-regras-específicas-2(S, RD, r, n)

11: trg := calcula-TRg(r, E)

12: **if** trg ≥ TRgMin then

13: R' := elimina-regras(E, R') {Processo de generalização}

14: else

15: trm := calcula-TRm(r, E)

16: if trm ≥ TRgMin then

17: R' := elimina-regras(E, R') {Processo de generalização}

18: else

19: R' := elimina-regras(r, R') {Processo de especialização}

20: n := GrauMax {"Pula" para a próxima regra mais geral presente em G}

21: end if

22: end if

23: **end for**

24: **end for**

25: **end for**

26: **return** R'

Inicialmente, o conjunto R', que irá armazenar as regras de associação filtradas recebe o resultado da função poda-regras-1, que elimina as que apresentam a forma da expressão 5.2.1.1. A função poda-regras-1 é descrita pelo mesmo algoritmo 2, utilizado para o relacionamento tipo A. Em seguida, as regras que apresentam a forma da expressão 5.2.1.2, para o antecedente (lado esquerdo) e para o conseqüente (lado direito) da regra, são eliminadas do conjunto R' pela função poda-regras-2, descrita pelo mesmo algoritmo 3. No próximo passo do algoritmo, as regras que apresentam a forma da expressão 5.2.1.3 são eliminadas do conjunto R', pela função poda-regras-3, descrita pelo mesmo algoritmo 4.

A seguir são gerados os subconjuntos, presentes em R', de regras que apresentam o mesmo conseqüente, pela função gera-subconjuntos, descrita anteriormente. Para cada subconjunto S gerado, a função gera-regras-gerais, descrita no algoritmo 5, gera o conjunto G das regras mais gerais (da forma da expressão 5.2.1.4), presentes em S. Para cada regra mais geral r $\in$ G, e para o grau de dependência, variando de 1 até *GrauMax*, é gerado o conjunto E das regras mais específicas redundantes com r (da forma da expressão 5.2.1.3), pela função gera-regras-específicas, descrita pelo algoritmo 7.

A partir da regra mais geral, e do conjunto correspondente das regras mais específicas, a função calcula-TRg calcula o valor da medida TRg, de acordo com a fórmula 5.2.2.1. Se o valor da medida TRg for maior ou igual ao valor mínimo especificado pelo usuário (*TRgMin*), as regras mais específicas são eliminadas do conjunto R'(processo de generalização). O processo de generalização é realizado através da função elimina-regras, que elimina do conjunto R' as regras presentes em E, e retorna o resultado. Caso contrário, o valor para a medida TRm é calculado pela função calcula-TRm, que utiliza a fórmula 5.2.2.3. Se este valor for menor ou igual ao valor mínimo especificado pelo usuário, mais uma vez aplicamos o processo de generalização das regras. Caso o valor da medida TRm seja maior que o valor mínimo especificado pelo usuário, o processo executado é o de especialização das regras. Desta vez, o conjunto R' recebe o resultado da função elimina-regras, que elimina do conjunto R' a regra r. Por fim, na linha 26, o algoritmo retorna o conjunto R'.

Os algoritmos 2, 3, 4, 5, 6 e 7 exibem funções chamadas pelo algoritmo 8, cujos nomes já são indicadores de suas funcionalidades. Julgamos assim mais conveniente exibi-los nos Apêndices.

5.2.2.3 Considerações Finais

O trabalho desenvolvido leva em consideração que a base de dados utilizada na mineração de regras de associação é representativa do domínio. Automatizou-se

a eliminação semântica da redundância das regras de associação mineradas, reduzindo a quantidade e, portanto, a complexidade, do conjunto de regras de associação. O caminho utilizado baseou-se no conhecimento de mundo representado pelas relações de dependência entre os atributos, via relacionamento ontológico.

Este trabalho, por meio de ações de generalização, procurou eliminar regras cobertas por outras mais gerais. Verificou-se que existem situações nas quais as regras mais gerais não descrevem o domínio de maneira adequada, situação na qual estas regras de associação mais gerais terão de conviver com regras especializadas. Foram definidas duas métricas, TRg e TRm, que indicam quando vale a pena utilizar a poda de regras de associação.

O algoritmo desenvolvido permite resultados bastante satisfatórios (serão exibidos resultados com poda de 50% do número de regras mineradas).

6 EXPERIMENTOS REALIZADOS

Os experimentos realizados foram aplicações dos algoritmos desenvolvidos para validar o modelo SemPrune. Para o plano amostral, foram utilizadas bases de dados amplamente conhecidas e referenciadas, quando possível. Nos demais casos, houve necessidade de geração das massas de teste das hipóteses formuladas.

6.1 MODELO DE BASE DE DADOS

A parte experimental foi realizada sobre bases de dados com transações ou seqüências de eventos. Os dados foram lidos de arquivos no formato de saída dos aplicativos Weka [WIT05] e ADDMiner.

O ambiente Weka é uma coleção de algoritmos de aprendizagem de máquina para aplicações de mineração de dados, desenvolvido na Universidade de Waikato (www.waikato.ac.nz), na Nova Zelândia, dentro da filosofia GNU (General Public License), de código fonte aberto, e possui ferramentas para pré-processamento de dados, classificação, regressão, agrupamento, regras de associação e visualização.

O ADDMiner, desenvolvido pelo laboratório ADDLabs (www.addlabs.uff.br), é um aplicativo que disponibiliza ferramentas para a mineração de dados utilizando a técnica de regras de associação. Dentre as suas principais características, este aplicativo, além das medidas suporte e confiança, gera as principais medidas objetivas de interesse encontradas na literatura, e permite controlar os atributos que podem aparecer no antecedente e no conseqüente das regras mineradas.

Para os relacionamentos do tipo "é-um", com cardinalidade m:n, as relações de dependência entre os atributos serão lidas a partir de um arquivo ASCII, cuja extensão deve ser *.dep, formado por uma série de comandos "@atributo" que especifiquem o nome e o tipo do atributo, seguido de um comando "@dependencia" que especifique os atributos que o determinam, e a regra de derivação. Os atributos

que determinam o atributo dependente serão escritos entre parênteses e separados por vírgulas, e a regra de derivação é delimitada por chaves.

A regra de derivação especifica como o valor do atributo dependente será calculado: [<condições:v:{s|n}>]; v:{s|n}.

Ou seja, a regra de derivação conterá zero ou mais declarações do tipo: condições:v:{s|n}, onde: condições especifica as condições para que o atributo assuma o valor v, e {s|n} indica se v deve ser avaliado como uma expressão tal que s = sim, n = não. O último termo, separado por ponto e vírgula (v:{s|n}), deve indicar o valor do atributo dependente, caso nenhuma das condições seja satisfeita. Mais uma vez, o parâmetro {s|n} indicará se v deve ser avaliado como uma expressão. De maneira mais simplificada, para os relacionamentos do tipo "é-um" 1:n (ou hierarquias), os arquivos *.dep se iniciam com o comando "@dependencia" e são seguidos de pares <item, ancestral direto>.

6.2 PLANO AMOSTRAL

Os experimentos foram realizados sobre diversas bases de dados, uma vez que nossa proposta contempla variadas situações. É bastante fácil obter bases de dados para mineração de dados clássica ou original, sem a existência de restrições e dependências entre os itens minerados. Contudo, esta pesquisa dirige-se especificamente a ambientes com restrições, uma vez que é a semântica das mesmas que permite a redução do número de regras e padrões minerados.

Foi necessária geração de bases de dados adequadas, que não fossem totalmente artificiais, criadas para a presente pesquisa.

Para o estudo dos relacionamentos do tipo "é-um", a geração foi fácil, utilizando-se bases de dados dos repositórios usuais da internet e introduzindo nelas atributos dependentes, do tipo faixa de domínio o agregador, usual na mineração de dados. Para os relacionamentos do tipo "parte-de", os repositórios de bases de dados são pródigos no estudo de DNA, mas a intuição necessária ao filtro humano na mineração de dados é escasso. A opção foi utilizar-se um banco de dados da

empresa Microsoft (Northwind Traders), distribuído com os produtos SQL Server e Access.

6.2.1 Bases de dados para mineração de regras de associação com inclusão do conhecimento de mundo no pós-processamento

6.2.1.1 Base de Dados do Censo Americano (Adult)

A Base de Dados Adult originou-se do Banco de Dados do Census Bureau norte- americano, resultados do censo de 1994, que foram extraídos por Barry Becker e disponibilizados no UCI repository [MER98]. O objetivo desta base de dados é a determinar que a renda de uma pessoa é maior que USD 50,000 por ano, menor ou igual àquele valor, através do atributo alvo *income*.

Os 14 atributos da Base são:

- Idade
- Tipo de trabalho
- Peso final (características socioeconômicas da população)
- Grau de educação
- Estado civil
- Tipo de ocupação
- Relação familiar
- Raça
- Sexo
- Ganho de capital
- Perda de capital
- Horas trabalhadas por semana
- Naturalidade
- Renda anual

6.2.1.2 Base de Dados da Arteriosclerose (STULONG)

A base de dados STULONG foi obtida de ECML/PKDD 2004 Discovery Challenge [ECM04], e contém informações sobre exames inicias em 1417 pacientes, além de dados pessoais (idade, estado civil, etc.), dados gerais (fumante ou não fumante, sedentarismo, etc.) e resultados de exames físicos e bioquímicos (altura, peso, pressão arterial, colesterol, etc.). Sua finalidade é estudar a arteriosclerose. A arteriosclerose é uma doença marcada pela formação de placas de gordura que impedem a passagem do sangue nas artérias. Quando ocorre nas artérias cerebrais e coronarianas caracteriza uma doença vascular, uma das mais estudadas, que é a principal causa de morte no mundo desenvolvido.

No presente estudo foram aproveitados dados com 22 atributos dos 244 coletados nos exames iniciais.

A mineração simulou seu uso por analistas de uma companhia de seguros, incluídos os atributos IMC (Body Mass Index ou *BMI*), status e a faixa etária (*age-range*). A faixa etária, atributo categórico, é comumente usada para enquadramento de clientes em planos de seguros e foi extraída do atributo numérico age. BMI significa Índice de Massa Corporal, utilizado nos estudos de obesidade, correlacionado com risco de diabetes e de moléstias cardiovasculares. Para calculá-lo, divide-se a massa da pessoa, medida em quilogramas, pelo quadrado da altura, medida em metros. O atributo status é uma discretização dos valores do IMC (*BMI*).

Os 18 atributos da Base são:

- Estado civil
- Grau de educação
- Relação de trabalho
- Atividade física no trabalho
- Atividade física depois do trabalho
- Transporte usual
- Duração do transporte para o trabalho
- Fumante
- Fumante por quanto tempo

- Álcool
- Altura
- Peso
- Pressão arterial
- Colesterol
- Triglicérides
- Cerveja
- Vinho
- Destilados

6.2.1.3 Base de dados sobre contratos de trabalho (Labor)

A base de dados Labor resultou da análise de todos os acordos coletivos de trabalho na indústria do Canadá, entre 1987 e o primeiro quadrimestre de 1988. Foram analisados contratos com, pelo menos, 500 empregados (professores, enfermeiros, pessoal universitário, policiais, etc.). O título desta base é "Acordos finais nas negociações trabalhistas na indústria do Canadá." Os dados foram usados para caracterizar contratos aceitáveis e não aceitáveis para os empregados. Compreende apenas 57 instâncias (contratos) com 16 atributos:

- Duração do acordo
- Aumento salarial no primeiro ano do contrato
- Aumento salarial no segundo ano do contrato
- Aumento salarial no terceiro ano do contrato
- Correção pelo custo de vida
- Horas trabalhadas por semana
- Contribuição patronal para fundo de pensão
- Pagamento por horas disponíveis (standby pay)
- Pagamento extra por trabalho em turno
- Auxílio educacional
- Número de feriados oficiais reconhecidos

- Dias de férias
- Ajuda em invalidez duradoura
- Contribuição patronal para plano dental
- Auxílio funeral (bereavement)
- Contribuição patronal para plano de saúde

6.2.2 Pré-processamento das bases de dados para mineração de regras de associação com inclusão do conhecimento de mundo no pós-processamento

A especificação dos arquivos de dependência, definidos no item 6.1, pode ser um exemplo para o caso da criação de um atributo dependente na base de dados Stulong, BMI expresso em função dos atributos weight e heigth, existentes na base, da forma {weight, height} $\overset{1}{\Rightarrow}$ {BMI}, como pode ser visto na Figura 6.1.

A linha inicial da especificação indica que o nome do atributo dependente é BMI e que se trata de um do tipo "real". A segunda linha mostra weight e height como atributos determinantes de BMI. Pela regra de derivação, BMI recebe o valor "vazio" caso um dos atributos weight e heigt seja também "vazio". Se isto não acontecer, o último termo da regra de derivação indica que à BMI é atribuído o valor resultado da expressão

```
weight/(( heigth /100)*( heigth /100))
```

```
@atributo BMI real
@dependencia (Peso, Altura) {Peso=vazio|Altura=vazio :vazio:n;
@atributo BMI int
@dependencia (weight,heigth) {heigth = <vazio> | weight = <vazio>
:<vazio>:n;weight/(( heigth /100)*( heigth /100)):s}
```

Figura 6.1 Especificação de dependência para um atributo criado

Base de dados Labor

Para a base de dados Labor, foram incluídos os atributos wage-inc e sweat-hours. O primeiro mede o acréscimo salarial dos dois primeiros anos no emprego, calculado pela adição dos atributos wage-increase-first-year e wage-increase-second-year, que indicam os acréscimos obtidos em cada um dos dois primeiros anos de emprego. O atributo sweat-hours é obtido da carga horária semanal (working-hours) e dos feriados oficiais (statutory-holidays), cujo número varia de local para local. Formalizando, chega-se a { wage-increase-first-year, wage-increase-second-year } $\overset{1}{\Rightarrow}$ { wage-inc } e { working-hours, statutory-holidays } $\overset{1}{\Rightarrow}$ { sweat-hours }. wage-inc, calculado analiticamente pela soma de seus atributos determinantes, enquanto sweat-hours será considerado "low", quando ((working-hours <= 40) and (statutory-holidays >= 10)) e "high", em caso contrário. Essas dependências podem ser vistas na figura 6.2.

```
@atributo wage-inc real
@dependencia (wage-increase-first-year,wage-increase-second-year) {; wage-increase-first-year + wage-increase-second-year :s}
@atributo sweat-hours string
@dependencia (working-hours,statutory-holidays) { working-hours <= 40 and statutory-holidays >= 10 :low:n;high:n}
```

Figura 6.2 Arquivo de dependências para a Base de Dados Labor

Base de dados Stulong

Os atributos incluídos foram BMI, Status e faixa etária. A faixa etária, atributo categórico, comumente usada para enquadramento de clientes em planos de seguros, foi extraída do atributo numérico age. BMI significa Índice de Massa Corporal, utilizado nos estudos de obesidade, e está correlacionado com risco de diabetes e de moléstias cardiovasculares. Para calculá-lo divide-se o peso, medido em metros, pelo quadrado da altura da pessoa, medida em metros. O atributo status é uma discretização dos valores do BMI. Formalizando, chega-se a {weight, height}

$\overset{1}{\Rightarrow}$ {BMI}., {weight, height} $\overset{2}{\Rightarrow}$ {Status}, {BMI } $\overset{1}{\Rightarrow}$ {Status} e {age } $\overset{1}{\Rightarrow}$ {age-range}. BMI será calculado analiticamente e status e age-range, obtidos por discretização.

A definição de status se econtra na Tabela 6.1 e a de age-range, na Tabela 6.2, de acordo com critérios de faixas.

Tabela 6.1 Definição do atributo Status

BMI	Status
BMI <= 18,5	Underweight
18,5 <BMI<= 24,9	Normal
24,9 <BMI<= 29,9	Overweight
29,9 <BMI<= 34,9	Obese
34,9 <BMI<= 39,9	Extremely Obese
BMI > 39,9	Morbidly Obese

Tabela 6.2 Definição do atributo age-range

Age	age-range
<= 18	0-18
18 <age<= 23	19-23
23 <age<= 28	24-28
28 <age<= 33	29-33
33 <age<= 38	34-38
38 <age<= 43	39-43
43 <age<= 48	44-48
48 <age<= 53	49-53
53 <age<= 58	54-58
58 <age	>=59

A figura 6.3 mostra as dependências para os atributos introduzidos na base de dados Stulong.

```
@atributo BMI int
@dependencia (weight,heigth) {heigth = <vazio> | weight = <vazio>
:<vazio>:n;weight/(( heigth /100)*( heigth /100)):s}
@atributo status string
@dependencia (BMI) {BMI <= 18.5 :Underweight:n,BMI > 18.5 and BMI <= 24.9
:Normal:n,BMI > 24.9 and BMI <= 29.9 :Overweight:n,BMI > 29.9 and BMI <=
34.9 :Obese:n,BMI > 34.0 and BMI <= 39.9 : Extremely Obese:n,BMI > 39.9
:Morbidlyly Obese :n;<vazio>:s}
@atributo age-range string
@dependencia (age) {age <= 18 :0-18:n,age > 18 and age <= 23 :19-23:n,age >
23 and age <= 28 :24-28:n,age > 28 and age <= 33 :29-33:n,age > 33 and age
<= 38 :34-38:n,age > 38 and age <= 43 :39-43:n,age > 43 and age <= 48 :44-
48:n,age > 48 and age <= 53 :49-53:n,age > 53 and age <= 58 :54-58:n,age >
58 :>=59:n;<vazio>:n}
```

Figura 6.3 Arquivo de dependências para a Base de Dados Stulong

Base de dados Adult

Na base de dados Adult, foram adicionados os atributos classe social e faixa etária: classe social, obtida dos atributos originais e do Hollingshead Index of Social Position (ISP) [HOL58], é bastante usada pelos gerentes de marketing, na formulação de estratégias; faixa etária, categórico foi extraída do atributo idade, numérico.

Formalizando, chega-se a {occupation-score, education-score } $\overset{1}{\Rightarrow}$ {ISP}, {ISP} $\overset{1}{\Rightarrow}$ {social-class} e {age } $\overset{1}{\Rightarrow}$ {age-group}. ISP será calculado analiticamente e social-class e age-group, obtidos por discretização.

ISP = (occupation-score * 7) + (education-score * 4)

A definição de social-class se encontra na Tabela 6.3 e a de age-group, na Tabela 6.4, de acordo com critérios de faixas considerando grupos de pessoas que caracterizam um ambiente social, político, histórico, e econômico, bem como uma estratégia de marketing[HAW04].

Tabela 6.3 Definição do atributo social-class

ISP	social-class
11-17	Upper
18-31	Upper-middle
32-47	Middle
48-63	Lower-middle
64-77	Lower

Tabela 6.4 Definição do atributo age-group

age	age-group
< 18	< 18
18-24	18-24
25-34	25-34
35-44	35-44
45-54	45-54
55-64	55-64
>=65	>=65

A figura 6.4 mostra o arquivo de dependências para a base de dados Adult.

```
@atributo ISP int
@dependencia (education-score,occupation-score) {;( occupation-score * 7) +
( education-score * 4):s}
@atributo social-class string
@dependencia (ISP) {ISP >= 11 and ISP <= 17:Upper:n,ISP >= 18 and ISP <= 31
:Upper-middle:n,ISP >= 32 and ISP <= 47:Middle:n,ISP >= 48 and ISP <=
63:Lower-middle:n,ISP >= 64 and ISP <= 77:Lower:n;<vazio>:s}
@atributo age-group string
@dependencia (age) {age < 18:<18:n,age >= 18 and age <= 24 :18-24:n,age >=
25 and age <= 34:25-34:n,age >= 35 and age <= 44:35-44:n,age >= 45 and age
<= 54:45-54:n,age >= 55 and age <= 64:55-64:n,age >= 65:>=65:n;<vazio>:s}
```

Figura 6.4 Arquivo de dependências para a Base de Dados Adult

6.2.3 Base de dados para mineração de regras de associação com inclusão do conhecimento de mundo no pré-processamento

Northwind Traders é um banco de dados de uma fictícia empresa comercial, fornecido como exemplo, com os produtos Microsoft Access e SQL Server, cuja finalidade é servir de suporte a aplicações e a MBP ("Modeling Business Processes"). Sua estrutura básica pode ser vista na Figura 6.5.
Principais tabelas do referido banco de dados:

- Orders, com 830 registros
- Products, com 77 registros
- Order details, com 2155 registros
- Employees, com 9 registros
- Categories, com 8 registros
- Customers, com 91 registros
- Shippers, com 3 registros
- Suppliers, com 29 registros

O banco de dados compõe-se de mais 7 tabelas auxiliares, é fornecido com 16 visões e 7 "stored procedures".

Para transformá-lo em uma base de dados de transações, que permita a mineração de dados do comportamento dos usuários, foi preciso criar uma visão adequada. Para tanto, pode-se observar, na Figura 6.5, que os dados relativos a hábitos de compra dos usuários estão registrados nas tabelas Order Details, Orders e Customers. Considerando que para efeito de mineração de dados, a identificação do usuário pode ser feita apenas pelo código CustomerID, basta fazer a junção das tabelas Order Details e Orders para criar a base de dados desejada. O conhecimento de mundo, base do relacionamento taxonômico, pode ser obtido pela junção das tabelas Categories e Products.

Assim, os atributos que farão parte da visão da base de dados de transações são CustomerID, Order Date e ProductID, e os que farão parte do conhecimento de mundo são ProductID e CategoryID.

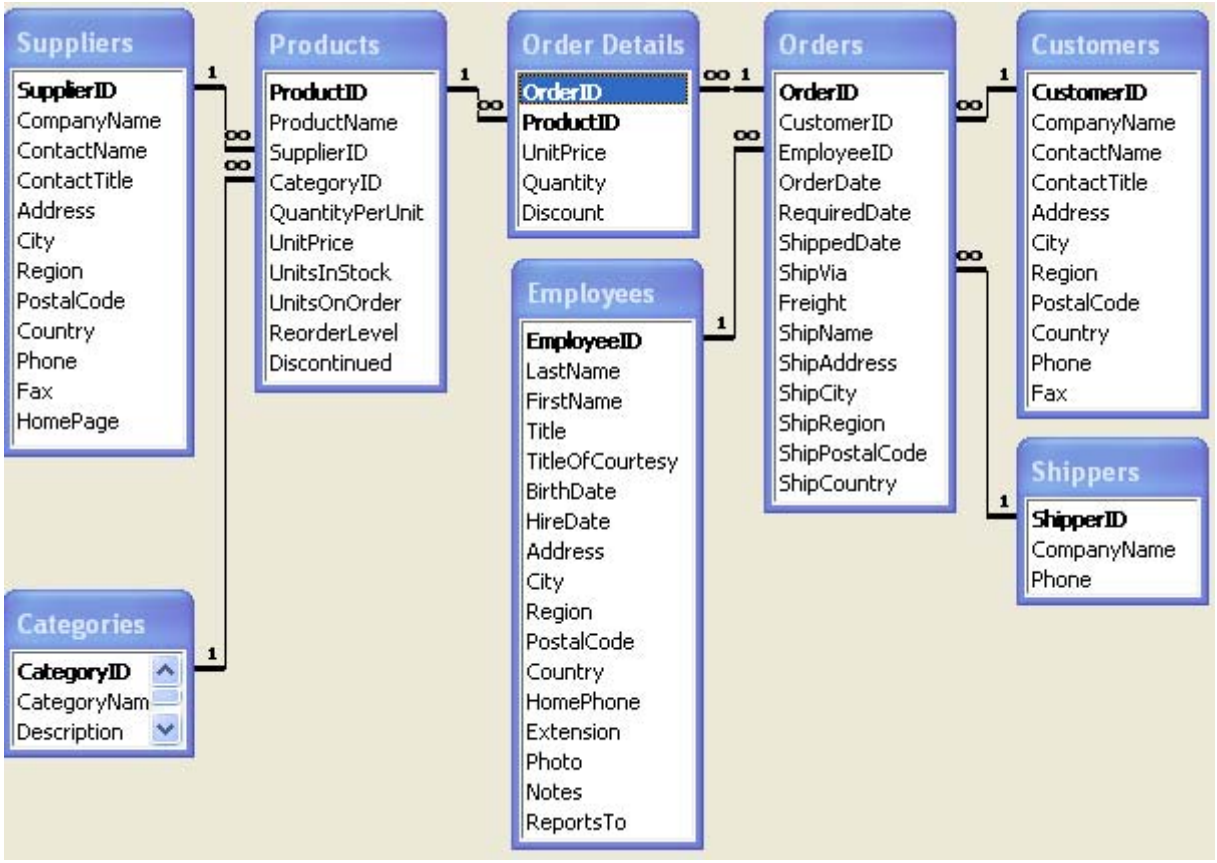


Figura 6.5 Estrutura do banco de dados Northwind Traders

Para transformar um banco de dados em base de dados de transações foi necessário agregar a tabela OrderDetails, fazendo com que uma transação fosse um itemset de produtos, adquiridos pelo mesmo CostumerID. Considerou-se uma transação um conjunto de itens adquiridos por um dado CustomerID, mas, para efeito de mineração de regras de associação, não é necessário o atributo CustomerID. Na geração da base de dados de transações foram formadas 72 transações.

6.3 PONTOS DE COMPARAÇÃO

Os pontos de comparação entre as hipóteses foram a quantidade de regras podadas e a validade das podas efetuadas: a primeira é quantitativa, verificando-se a quantidade de regras podadas, pelo modelo e pelo volume de controle; a segunda é a verificação da validade das podas efetuadas. Do ponto de vista semântico, o modelo SemPrune funciona como um observador humano que compara todas as regras mais especializadas com a mais geral correspondente, à luz da ontologia.

Os experimentos realizados tiveram como objetivo mostrar:

- a redução quantitativa de regras de associação, por meio de poda semântica;
- a incapacidade de as técnicas sintáticas de poda perceberem a existência de regras mais gerais.

Esta incapacidade foi analisada escolhendo-se métodos não semânticos, que se utilizam de medidas de interesse das regras. Como padrão de comparação, no pós-processamento, foram escolhidas as medidas de interesse objetivas: convicção, especificidade, lift e novidade.

Esta incapacidade foi analisada por dois conjuntos de resultados:

- verificação quantitativa das regras de associação podadas não por medidas objetivas de interesse, mas por processo de generalização do modelo SemPrune;
- yabelas de desempenho, mostrando que as diferenças de comportamento entre os resultados obtidos das diversas medidas objetivas de interesse, no tocante às dicotomias "devia ter sido podada/não devia ter sido podada" e "foi podada/não foi podada" são muito grandes e que, para uma mesma medida, pequenos valores do parâmetro de corte apresentam variações de desempenho muito distantes da linearidade e/ou previsibilidade.

6.4 AMBIENTE DE AVALIAÇÃO

O ambiente de avaliação compreendeu um conjunto de três bases de dados, com inclusão do conhecimento de mundo no pós-processamento e uma, transformada para experimentos, com inclusão do conhecimento de mundo no pré-processamento.

A Tabela 6.5 mostra um resumo das características das bases de dados utilizadas na mineração de regras de associação para relacionamentos tipo A. As bases Adult e Labor foram projetadas especificamenteb para a busca de um alvo determinado. No caso da Base Adult, o objetivo era verificar a renda anual de cada pessoa, e no da Base Labor, classificar um contrato de trabalho na indústria canadense como bom ou ruim. A Base Stulong é mais ambiciosa, acompanhando o estado de saúde ou de risco de pacientes ao longo de um largo tempo. Em conseqüência, os atributos, potencialmente, alvos são diversos.

Tabela 6.5 Bases de dados utilizadas na mineração de regras de associação com inclusão do conhecimento de mundo no pós-processamento

Base de dados	**# Atributos**	**# Registros**	**Atributos criados**	**Atributo alvo**
Adult	15	32.561	Classe-social Faixa-etária	income
STULONG	22	1.417	BMI Status Faixa-etária	blood_pressure
Labor	18	57	Faixa-etária Sweat-hours	class

A Tabela 6.6 mostra as configurações utilizadas no pós-processamento das regras de associação mineradas. Em todos os casos, a confiança utilizada foi de 0,7. Na Tabela 6.6, encontram-se o número máximo de itens que uma regra pode ter e o número de regras mineradas.

Tabela 6.6 Configurações usadas na mineração de regras de associação com inclusão do conhecimento de mundo no pós-processamento

Base de dados	# máximo de itens	Suporte	Regras mineradas
Adult	5	0,04	2.924
STULONG	3	0,04	2.912
Labor	5	0,04	181.229

Para os relacionamentos com inclusão do conhecimento de mundo no pré-processamento, as regras geradas tiveram 5 como número máximo de itens, 0,04, como suporte mínimo, e 9941 regras mineradas para a base de dados Northwind Traders.

6.5 PROCESSO DE AVALIAÇÃO

O processo de avaliação do modelo SemPrune, para os relacionamentos com inclusão do conhecimento de mundo no pós-processamento, seguiu o roteiro:

1 Preparação da entrada para a mineração de dados
- 1.1 Bases de dados
- 1.2 Medidas de interesse objetivas a utilizar
- 1.3 Atribuição de valores aos parâmetros (tamanho de regra e patamares de corte)

2 Mineração para obtenção de regras de associação utilizando o algoritmo Apriori com um "software" que permita embutir nas regras os valores das medidas de interesse especificadas pelo usuário

3 Preparação da entrada para o processamento semântico
- 3.1 Arquivo de regras de associação mineradas
- 3.2 Descrição de atributos
- 3.3 Relacionamentos entre os atributos
- 3.4 Atribuição de valores aos parâmetros de corte

4 Processamento semântico

5 Análise dos resultados

5.1 Verificação da consistência dos processos de generalização

5.2 Extração dos resultados quantitativos

5.3 Extração dos resultados de qualidade de corte

5.4 Comparação dos resultados obtidos pelo modelo SemPrune e pelos métodos sintáticos

O processo de avaliação do modelo SemPrune, para os relacionamentos com inclusão do conhecimento de mundo no pré-processamento, seguiu o roteiro:

1 Geração da base de dados de transações via pré-processamento

2 Enriquecimento da base de dados

2.1 Base de dados

2.2 Relacionamentos entre os atributos

2.3 Processamento de inclusão de atributos dependentes

3 Preparação da entrada para a mineração de dados

3.1 Base de dados

3.2 Medidas de interesse objetivas a utilizar

3.3 Atribuição de valores aos parâmetros (tamanho de regra e patamares de corte)

4 Mineração para obtenção de regras de associação utilizando o algoritmo Apriori com um "software" que permita embutir nas regras os valores das medidas de interesse especificadas pelo usuário

5 Preparação da entrada para o processamento semântico

5.1 Arquivo de regras de associação mineradas

5.2 Descrição de atributos

5.3 Relacionamentos entre os atributos

5.4 Atribuição de valores aos parâmetros de corte

6 Processamento semântico

7 Análise dos resultados

7.1 Verificação da consistência da eliminação de regras com redundâncias internas

7.2 Verificação da consistência dos processos de generalização

7.3 Verificação da consistência dos processos de especialização

7.4 Extração dos resultados quantitativos

7.5 Extração dos resultados de qualidade de corte

7.6 Comparação dos resultados obtidos pelo modelo SemPrune e pelos métodos sintáticos

6.6 CONSIDERAÇÕES GERAIS

O experimento foi realizado para comprovar os efeitos desejados pela aplicação do modelo SemPrune, um enriquecedor do conjunto de regras de associação mineradas, acoplado a uma poda semântica. Para esta aplicação, os experimentos foram realizados sobre bases de dados com relacionamentos de tipos A e B, que englobam a inclusão do conhecimento de mundo tanto no pós-processamento, quanto no pré-processamento, e são suficientes para testar as propriedades do modelo.

A poda semântica compreende a eliminação de regras com as seguintes características:

- mais específicas, substituídas por mais gerais em processos de generalização;
- com redundância interna introduzida pelo enriquecimento, na fase de pré-processamento;
- mais gerais, introduzidas pelo enriquecimento na fase de pré-processamento, que não conseguiram substituir adequadamente as regras mais específicas correspondentes.

A aplicação do modelo SemPrune a bases de dados com relacionamento de tipo A só efetua o primeiro tipo de poda, e às de tipo B efetua todos os três tipos de poda.

7 ANÁLISE DOS RESULTADOS

A partir daqui serão apresentados os resultados obtidos na tentativa de redução do número de regras de associação mineradas. Inicialmente, a análise vai mostrar o que se obteve, quando em presença de um relacionamento do tipo A entre os atributos da base de dados. Em seguida, os resultados encontrados quando a base de dados apresenta relacionamentos entre atributos do tipo B. A refutação da hipótese nula será mostrada nos itens 7.1.1.4 e 7.2.1.1.

7.1 MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO COM INCLUSÃO DO CONHECIMENTO DE MUNDO NO PÓS-PROCESSAMENTO

Descreve-se, a seguir, o que foi obtido aplicando-se o modelo sobre as bases de dados descritas nos itens 6.1.1.1 a 6.1.1.3.

A eliminação sumária de regras eliminadas, por apresentarem as formas das expressões 5.2.1.1, 5.2.1.2 e 5.2.1.3, não ocorrerá, pois quando o enriquecimento é feito no pós-processamento, a filtragem acontece simultaneamente. Assim sendo, iniciar-se-á pelo tratamento dos processos de generalização e especialização das regras.

A apresentação dos resultados será feita em forma tabular, tal como mostrado na tabela 7.1 e seguintes. Inicia-se, identificando o tipo do processo: poda de regras da forma das expressões 5.2.1.1, 5.2.1.2 ou 5.2.1.3, processo de generalização ou processo de especialização. O corpo da tabela apresenta cinco colunas: o identificador da regra (**Id**), o antecedente (**Antecedente**), o conseqüente (**Conseqüente**) e os valores para as medidas suporte (**Sup**) e confiança (**Conf**).

A filtragem do modelo utiliza como métricas CRg e CRM, com valor de 0,70, ou seja, o mesmo utilizado para a confiança na fase de mineração. O grau máximo de dependência, na amostra, foi 2.

7.1.1.1 Base de dados da Arteriosclerose (Stulong)

Foram eliminadas 8163 regras, mineradas da base de dados da Arteriosclerose, número que representa 23,71% do total. Todas elas foram eliminadas nos 1063 processos de generalização.

Serão descritos, a seguir, alguns exemplos de generalização.

A tabela 7.1 mostra um exemplo baseado na relação de dependência {age} $\overset{1}{\Rightarrow}$ { age-range }. A regra mais geral, referente ao atributo **Age-range**, possui duas regras mais específicas com o atributo **age**, cujos valores são: 47 e 44. O valor de CRg calculado foi de 84%, sendo essa a probabilidade do paciente ser casado, dado que está na faixa de 49 a 53 anos, e sua atividade física após o trabalho é moderada. Esta medida é superior ao valor especificado para o ponto de corte (70%), significando que a regra com o atributo Age-range pode substituir as regras com o atributo age.

Tabela 7.1: Processo de generalização baseado na relação de dependência {age} $\overset{1}{\Rightarrow}$ { age-range } da base STULONG.

Processo de generalização com CRg = 0,84 e CRm = 0,44				
Regra ID	Antecedente	Conseqüente	Sup	Conf
Regras mais especializadas a serem podadas				
80	(age=47), (phys_act_after_job=moderate activity)	(marital_status=married)	0,07	0,96
107	(age=44), (phys_act_after_job=moderate activity)	(marital_status=married)	0,04	0,91
Regra mais geral que representa as regras especializadas podadas				
73	(age-range=44-48), (phys_act_after_job=moderate activity)	(marital_status=married)	0,12	0,86

Tabela 7.2: Processo de generalização baseado na relação de dependência {BMI} $\overset{1}{\Rightarrow}$ {Status} da base STULONG.

Processo de generalização com CRg = 0,90 e CRm = 0,39				
Regra ID	Antecedente	Conseqüente	Sup	Conf
Regras mais especializadas a serem podadas				
1971	(BMI=27), (job_transp_duration=around 1/2 hour)	(phys_act_after_job=moderate activity)	0,10	0,77
1975	(BMI=28), (job_transp_duration=around 1/2 hour)	(phys_act_after_job=moderate activity)	0,08	0,72
1979	(BMI=25), (job_transp_duration=around 1/2 hour)	(phys_act_after_job=moderate activity)	0,15	0,72
Regra mais geral que representa as regras especializadas podadas				
1967	(status=Overweight), (job_transp_duration=around 1/2 hour)	(phys_act_after_job=moderate activity)	0,35	0,70

A tabela 7.2 mostra três regras mais específicas, indicando que os valores 27, 28 e 25 do atributo **BMI**, correspondem ao valor "Overweight" do atributo **status** (25 a 30). A probabilidade desta ocorrência, medida por CRg, é de 90%. Como esta medida ultrapassou o valor de confiança mínimo do usuário, o processo de generalização das regras predomina.

7.1.1.2 Base de dados Labor

Foram eliminadas 32.874 regras da base de dados Labor,o que representa 18,14% do total de regras mineradas, eliminadas em 587 processos de generalização.

Seguem-se exemplos, ilustrando as relações de dependência entre os dados. A tabela 7.3 apresenta um exemplo de generalização baseada na relação de dependência {working-hours, statutory**-**holidays } $\overset{1}{\Rightarrow}$ {sweat-hours}. A tabela 7.11

apresenta um exemplo de generalização baseada na relação de dependência { wage-increase-first-year, wage-increase-second-year } $\overset{1}{\Rightarrow}$ {wage-inc}.

Tabela 7.3: Processo de generalização baseado na relação de dependência {working-hours, statutory-holidays } $\overset{1}{\Rightarrow}$ {sweat-hours} da base Labor.

Processo de generalização com CRg = 0,84 e CRm = 0,32				
Regra ID	Antecedente	Consequente	Sup	Conf
Regras mais especializadas a serem podadas				
128025	(standby-pay=0.000), (statutory-holidays=12), (working-hours=35.000)	(wage-increase-third-year=0.000)	0,04	1,00
128224	(standby-pay=0.000), (statutory-holidays=11), (working-hours=38.000)	(wage-increase-third-year=0.000)	0,04	1,00
128226	(standby-pay=0.000), (statutory-holidays=12), (working-hours=38.000)	(wage-increase-third-year=0.000)	0,04	1,00
128803	(standby-pay=0.000), (statutory-holidays=11), (working-hours=37.000)	(wage-increase-third-year=0.000)	0,04	1,00
Regra mais geral que representa as regras especializadas podadas				
126467	(standby-pay=0.000), (sweat-hours=low)	(wage-increase-third-year=0.000)	0,17	0,89

No processo de generalização da tabela 7.3, a medida CRg indica que há uma probabilidade de 84% de a empresa não oferecer nenhum aumento automático, no

terceiro ano. Desta forma, o processo de generalização foi aplicado, e as regras mais específicas, eliminadas em prol da regra mais geral, com o atributo **sweat-hours**.

Tabela 7.4: Processo de generalização baseado na relação de dependência {wage-increase-first-year,wage-increase-second-year} $\overset{1}{\Rightarrow}$ {wage-inc} da base Labor.

Processo de generalização com CRg = 0,44 e CRm = 1,00				
Regra ID	Antecedente	Consequente	Sup	Conf
Regras mais especializadas a serem podadas				
87388	(wage-increase-first-year=2.500), (wage-increase-second-year=2.500)	(pension=empl_contr), (wage-increase-third-year=0.000)	0,04	1,00
Regra mais geral que representa as regras especializadas podadas				
70893	(wage-inc=5.)	(pension=empl_contr), (wage-increase-third-year=0.000)	0,09	1,00

No exemplo da tabela 7.4, de acordo com a medida CRg, é baixa a probabilidade de a empresa contribuir para o fundo de pensão do empregado e lhe dar aumento no terceiro ano de trabalho dado que a empresa oferece um razoável aumento salarial nos dois primeiros anos de emprego, mas o valor de CRm é 100%, o que valida a regra mais geral. Assim, no processo de generalização, as regras relativas aos atributos **wage-increase-first-yea**r e **wage-increase-second-year** foram eliminadas em prol da regra mais geral com **wage-inc**.

Calculando CRg para as regras da tabela 7.5, observa-se que a probabilidade de o trabalho "moleza" (sweat-hours=low), ser considerado bom, sem pagar horas paradas é de 89%. Prevaleceu o processo de generalização, e as regras mais específicas, englobando **statutory-holydays** e **working-hours**, eliminadas em favor da regra mais geral com **sweat-hours**.

Tabela 7.5: Processo de generalização baseado na relação de dependência {working-hours, statutory-holidays } $\overset{1}{\Rightarrow}$ {sweat-hours} da base Labor

Processo de generalização com CRg = 0,89 e CRm = 1,00				
Regra ID	Antecedente	Consequente	Sup	Conf
Regras mais especializadas a serem podadas				
160391	(statutory-holidays=12.000), working-hours=35.000)	(class=good), (standby-pay=0.000)	0,04	1,00
161680	(statutory-holidays=11.000), working-hours=37.000)	(class=good), (standby-pay=0.000)	0,04	1,00
Regra mais geral que representa as regras especializadas podadas				
157053	(sweat-hours=low)	(class=good), (standby-pay=0.000)	0,09	1,00

7.1.1.3 Base de dados do censo americano (Adult)

Com a base de dados do censo americano foram eliminadas, no total, 515 regras, valor que representa 17,61% do total de regras mineradas e 25,91% do total das que apresentavam dependência (uma regra apresenta dependência quando possui um atributo dependente, ou que determina outro atributo da base).

Os 103 processos de generalização eliminaram ao todo 515 regras. Um exemplo de generalização baseada na relação de dependência {age } $\overset{1}{\Rightarrow}$ {age-group} é apresentado na tabela 7.6. A tabela 7.7 apresenta um exemplo de generalização baseada na relação de dependência {occupation-score, education-score } $\overset{2}{\Rightarrow}$ {social-class}.

De acordo com a tabela 7.6, existe uma regra mais específica para cada uma das idades que são mapeadas para o valor "25-34" do atributo faixa-etária, exceto para a idade 34. Neste caso, a medida CRg indica a probabilidade de o entrevistado ter entre 25 e 33 anos, dado que ele está na faixa dos 25 a 34, é do sexo masculino e nativo dos EUA. O valor calculado para a medida CRg foi 74%, um valor acima da confiança mínima do usuário. Desta forma, o processo de generalização foi aplicado,

e as regras relativas ao atributo **age,** eliminadas em prol da regra mais geral com **faixa-etária**.

Tabela 7.6: Processo de generalização baseado na relação de dependência {age } $\overset{1}{\Rightarrow}$ {age-group} da base Adult.

Processo de generalização com CRg = 0,74 e CRm = 0,76				
Regra ID	Antecedente	Consequente	Sup	Conf
Regras mais especializadas a serem podadas				
128025	(age=28), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,83
128224	(age=31), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,75
128226	(age=30), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,75
128803	(age=32), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,71
128025	(age=25), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,92
128224	(age=29), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,80
128226	(age=27), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,89
128803	(age=33), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,74
128803	(age=26), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,89
Regra mais geral que representa as regras especializadas podadas				
126467	(age-group=25-34), (sex=male), (native-country=United-States)	(income=<=50K))	0,12	0,79

No exemplo da tabela 7.7, de acordo com a medida CRg, a probabilidade de o entrevistado ter a escolaridade e a ocupação que aparecem nas regras mais específicas, é de 41%, um valor abaixo da confiança mínima do usuário, dado que ele nunca foi casado, pertence à classe social média, e à classe de trabalho privada. Entretanto, a medida CRm igual a 96% indica que a regra mais geral continua válida

para os demais casos de escolaridade e ocupação dos mesmos entrevistados. Assim, no processo de generalização, as regras relativas aos atributos **educationscore** e **occupationscore** foram eliminadas, em prol da regra mais geral com **classe-social**.

Tabela 7.7: Processo de generalização baseado na relação de dependência {occupation-score, education-score } $\overset{2}{\Rightarrow}$ {social-class} da base Adult.

Processo de generalização com CRg = 0,41 e CRm = 0,96				
Regra ID	Antecedente	Consequente	Sup	Conf
Regras mais especializadas a serem podadas				
128025	(education-score=3), (marital-status=Never-married), (occupation-score=3), (workclass=Private)	(income=<=50K)	0,01	0,99
128224	(education-score=3), (marital-status=Never-married), (occupation-score=4), (workclass=Private)	(income=<=50K)	0,01	0,99
128226	(education-score=4), (marital-status=Never-married), (occupation-score=3), (workclass=Private)	(income=<=50K)	0,01	0,99
Regra mais geral que representa as regras especializadas podadas				
126467	(marital-status=Never-married), (social-class=Middle), (workclass=Private)	(income=<=50K)	0,07	0,97

7.1.1.4 Resumo dos resultados

Um resumo dos resultados obtidos pelo método proposto sobre as três bases de dados selecionadas pode ser visto na tabela 7.8. A primeira coluna mostra as bases de dados; a segunda, o número de regras de associação mineradas; a terceira, o número de regras consideradas redundantes eliminadas pelo método semântico; a quarta, a razão entre o número de regras eliminadas e o número total de regras mineradas.

Tabela 7.8: Resumo dos resultados obtidos pelo método SemPrune sobre as três bases de dados selecionadas com inclusão do conhecimento de mundo no pós-processamento

Base de Dados	Regras mineradas	Regras com dependência	Regras eliminadas	Eliminadas/ Total
Adult	2924	1988	515	17,61%
Stulong	34422	12601	8163	23,71%
Labor	181229	134962	32874	18,14%

Observa-se que o filtro semântico atuou em todas as bases de dados da amostra. As taxas de redução foram razoáveis, chegando a mais de 23% para a base Stulong. A proposta de métodos semânticos sempre contribui para diminuir o número de regras filtradas por métodos semânticos, além de dar mais consistência aos resultados.

A Tabela 7.15 mostra os dados de refutação da hipótese nula, para o caso da inclusão do conhecimento de mundo no pós-processamento das regras de associação mineradas. e o ganho em precisão, obtido pela aplicação do modelo SemPrune. Em todas as bases de dados utilizadas não houve nenhuma poda desnecessária de regra, e a quantidade de informação recuperada diminuiu. Os ganhos variaram de 21% (Base de dados Adult) a 34% (Base de dados Stulong). O "recall" se manteve constante, pois não sofreram alterações nem a interseção dos conjuntos de informações relevantes e das informações recuperadas, nem o conjunto das informações relevantes.

7.1.1.5 Análise comparativa entre medidas de interesse objetivas

É evidente a redução da cardinalidade do conjunto obtido da mineração de dados com o uso de técnicas semânticas. Além disso, a verificação da melhora da qualidade do produto obtido pelo uso destas técnicas serve de comparação entre o presente trabalho e as técnicas sintáticas usuais.

Processos de generalização/especialização produzem regras mais relevantes, pois acompanham o processo intuitivo humano.

O teste desta hipótese pode ser feito comparando-se resultados gerados com a filtragem de pós-processamento, orientados por medidas de interesse objetivas, especificamente: convicção, especificidade, lift e novidade, todas descritas no item 3.1.

Inicialmente, a aplicação dessas quatro medidas objetivas será feita sobre qualquer ambiente de nossa pesquisa, descrito na Tabela 7.8. Serão analisadas as bases de dados estudadas, usando-se cinco valores distintos para cada medida objetiva de interesse, como ponto de corte. As Tabelas 7.9, 7.10 e 7.11 mostram os resultados obtidos. Em cada uma delas, na primeira coluna aparece o nome da medida de interesse utilizada e na segunda, o valor de corte considerado.

Depois disto, dois grupos de três colunas: o primeiro grupo gerado usando as métricas CRg e CRm e o segundo, usando as métricas TRg e TRm.

Na primeira coluna de cada grupo, a terceira e a sexta da tabela, aparece a fração das regras eliminadas pela medida de interesse para a métrica escolhida; na segunda, a quarta e a sétima da tabela, o número de regras deixadas (consideradas interessantes, segundo a medida); na terceira, a quinta e a oitava, a razão entre o número de regras eliminadas pelo filtro semântico e o número de regras não podadas pela medida de interesse.

Observando as Tabelas 7.12 a 7.14 pode-se notar que os filtros objetivos deixam passar regras redundantes, que podem ser detectadas por filtros semânticos. Além disso, os filtros objetivos têm um ajuste muito sensível, de forma que pequenas variações do ponto de corte apresentem resultados significativamente diferentes. Tal comportamento exige muito do usuário, o que, em geral, leva à baixa qualidade do resultado. Tomando como exemplo a base de dados Stulong, verifica-se que uma variação de apenas 0,02 no valor de corte de especificidade (0,95 para 0,97) fez com que o número de regras deixadas caísse de 2.583 para 1.307, com redução de 49%. Uma variação de apenas 0,10 no valor de corte de lift (1,00 para 1,10) fez com que o número de regras deixadas caísse de 2.950 para 921, com redução de 68,77%.

Tabela 7.9: Resultados obtidos pelas medidas de interesse com a base Stulong

Métricas		CRg e CRm			TRg e TRm		
Medida	Corte	% Regras eliminadas	#Regras deixadas	SemPrune % regras eliminadas	% Regras eliminadas	#Regras deixadas	SemPrune % regras eliminadas
Convicção	1,10	45,85%	1597	26,35%	45,85%	1597	15,14%
	1,20	53,33%	1377	16,26%	53,33%	1377	9,76%
	1,30	58,04%	1238	14,47%	58,04%	1238	6,51%
	1,40	61,45%	1137	12,60%	61,45%	1137	4,72%
	1,50	63,18%	1086	11,54%	63,18%	1086	4,12%
Especificidade	0,95	12,44%	2583	12,83%	12,44%	2583	7,28%
	0,97	26,33%	1307	8,91%	26,33%	1307	1,37%
	0,98	5,22%	2180	2,46%	5,22%	2180	0,82%
	0,99	53,43%	1374	0,65%	53,43%	1374	0,65%
	1,00	59,38%	1198	0,00%	59,38%	1198	0,75%
Lift	1,00	0,00%	2950	17,61%	0,00%	2950	10,93%
	1,10	20,86%	921	5,84%	20,86%	921	2,92%
	1,20	24,89%	2216	0,00%	24,89%	2216	1,21%
	1,30	25,30%	2204	0,00%	25,30%	2204	1,22%
	1,40	26,06%	2181	0,00%	26,06%	2181	1,23%
Novidade	0,00	0,00%	2950	17,61%	0,00%	2950	10,93%
	0,10	100,00%	0	0,00%	100,00%	0	0,00%
	0,20	100,00%	0	0,00%	100,00%	0	0,00%
	0,30	100,00%	0	0,00%	100,00%	0	0,00%
	0,40	100,00%	0	0,00%	100,00%	0	0,00%

As Tabela 7.9 a 7.11 mostram como as medidas objetivas de interesse ignoram totalmente a presença das regras semânticas. A primeira coluna identifica as medidas objetivas; a segunda mostra, para cada medida, cinco pontos de corte adotados para exemplo; a terceira, quarta e quinta colunas referem-se a resultados obtidos pela utilizacão das métricas CRg e CRm; as colunas de 6 a 8 mostram os resultados obtidos utilizando-se as métricas TRg e TRm. Na terceira coluna aparece a porcentagem do total de regras mineradas cortadas pelo critério sintático. A quantidade de regras não eliminadas aparece na quarta coluna. Na quinta coluna aparecem quantas regras não eliminadas sintaticamente, o são pelo modelo SemPrune.

Tabela 7.10: Resultados obtidos pelas medidas de interesse com a base Labor

Métricas		CRg e CRm			TRg e TRm		
Medida	Corte	% Regras eliminadas	#Regras deixadas	SemPrune % regras eliminadas	% Regras eliminadas	#Regras deixadas	SemPrune % regras eliminadas
Convicção	1,10	85,71%	113	100,00%	85,71%	113	100,00%
	1,20	85,97%	111	100,00%	85,97%	111	100,00%
	1,30	86,09%	110	100,00%	86,09%	110	100,00%
	1,40	86,35%	108	100,00%	86,35%	108	100,00%
	1,50	87,36%	100	100,00%	87,10%	102	100,00%
Especificidade	0,95	2,40%	772	100,00%	2,40%	772	100,00%
	0,97	9,10%	719	100,00%	9,10%	719	100,00%
	0,98	14,54%	676	100,00%	14,54%	676	100,00%
	0,99	14,54%	676	100,00%	14,54%	676	91,12%
	1,00	14,54%	676	100,00%	14,54%	676	82,99%
Lift	1,00	0,00%	791	100,00%	0,00%	791	100,00%
	1,10	1,14%	782	100,00%	1,14%	782	100,00%
	1,20	19,72%	635	100,00%	19,72%	635	100,00%
	1,30	20,99%	625	100,00%	20,99%	625	100,00%
	1,40	38,05%	490	100,00%	38,05%	490	100,00%
Novidade	0,00	0,00%	791	100,00%	0,00%	791	100,00%
	0,10	100,00%	0	0,00%	100,00%	0	0,00%
	0,20	100,00%	0	0,00%	100,00%	0	0,00%
	0,30	100,00%	0	0,00%	100,00%	0	0,00%
	0,40	100,00%	0	0,00%	100,00%	0	0,00%

Tabela 7.11: Resultados obtidos pelas medidas de interesse com a base Adult

Métricas		CRg e CRm			TRg e TRm		
Medida	Corte	% Regras eliminadas	#Regras deixadas	SemPrune % regras eliminadas	% Regras eliminadas	#Regras deixadas	SemPrune % regras eliminadas
Convicção	1,10	46,96%	61	88,52%	46,96%	61	88,52%
	1,20	56,52%	50	92,00%	56,52%	50	92,00%
	1,30	63,48%	42	95,24%	63,48%	42	95,24%
	1,40	68,70%	36	94,44%	68,70%	36	94,44%
	1,50	71,30%	33	96,97%	71,30%	33	96,97%
Especificidade	0,95	16,52%	96	100,00%	16,52%	96	100,00%
	0,97	31,30%	79	100,00%	31,30%	79	100,00%
	0,98	52,17%	55	100,00%	52,17%	55	100,00%
	0,99	67,83%	37	100,00%	67,83%	37	100,00%
	1,00	77,39%	26	100,00%	77,39%	26	100,00%
Lift	1,00	17,39%	95	91,58%	17,39%	95	91,58%
	1,10	58,26%	48	95,83%	58,26%	48	95,83%
	1,20	66,96%	38	97,37%	66,96%	38	97,37%
	1,30	73,04%	31	100,00%	73,04%	31	100,00%
	1,40	75,65%	28	100,00%	75,65%	28	100,00%
Novidade	0,00	17,39%	95	88,42%	17,39%	95	88,42%
	0,10	100,00%	0	0,00%	100,00%	0	0,00%
	0,20	100,00%	0	0,00%	100,00%	0	0,00%
	0,30	100,00%	0	0,00%	100,00%	0	0,00%
	0,40	100,00%	0	0,00%	100,00%	0	0,00%

Infelizmente, as medidas de interesse objetivas eliminam regras de interesse do usuário e deixam de eliminar as que deveriam. A hipótese adotada foi a de que métodos semânticos buscam o real significado do repositório e as regras obtidas por esses métodos são relevantes para o usuário. Um experimento revelador da comparação entre métodos sintáticos (poda usando convicção, especificidade, lift e novidade) e semânticos (poda usando generalização/especialização) de pós-processamento da mineração de dados pode ser analisado com tabelas de desempenho. À semelhança das matrizes de confusão pode-se considerar que a poda de regras pode ser encarada sob dois aspectos:

- devia (D) ou não devia (ND) ser podada, analisado sob o ponto de vista semântico e exibido nas linhas da tabela,
- foi (F) ou não foi (NF) podada, constatado pela aplicação de filtragem via medida objetiva de interesse e exibido nas colunas da tabela.

As Tabelas 7.12, 7.13, 7.14 e 7.22 são relativas a desempenho, o que aparece na terceira, quarta, sexta e sétima colunas. Nestas, a diagonal principal apresenta o número de regras que deviam e foram podadas e as que não deviam e não o foram. Isto configura acerto dos métodos sintáticos. A diagonal secundária exibe o número de casos em que a medida de interesse objetiva não atuou corretamente, eliminando regras que não deveria, ou mantendo as que deveriam ser eliminadas.

Foram realizados experimentos nas bases de dados do plano amostral, com as quatro medidas objetivas de interesse que vimos trabalhando, utilizando cinco valores de cada uma das referidas medidas como ponto de corte.

Inequivocamente, métodos sintáticos conseguem, variando os pontos de corte, podar um número muito maior de regras do que o método semântico deste trabalho; todavia, eliminam regras que não deveriam ser eliminadas e deixam de eliminar as que deveriam.

As Tabelas 7.12, 7.13 e 7.14 mostram os resultados obtidos para as bases de dados do experimento. A primeira coluna identifica a medida objetiva e respectivo ponto de corte. Para cada configuração existem duas linhas, identificadas na segunda coluna, uma correspondendo à quantidade de regras que deveriam ter sido podadas (D) e outra, de regras que não deveriam (ND). A terceira, quarta e quinta colunas referem-se às métricas CRg e CRm enquanto a sexta, sétima e oitava referem-se às métricas TRg e TRm. A terceira e sexta colunas mostram os números de regras podadas (F) pela medida especificada na primeira coluna. A quarta e a sétima colunas mostram os números de regras não podadas (NF). Na quinta e na oitava colunas aparecem as frações das regras eliminadas pela medida de interesse para a métrica escolhida. Estes números são os mesmos já mostrados na terceira e na sexta colunas das Tabelas 7.9, 7.10 e 7.11.

Tabela 7.12: Tabela de desempenho sobre a base de dados Stulong

Métrica		CRg e CRm			TRg e TRm		
Medida		F	NF	% elim	F	NF	% elim
Convicção = 1,10	D	807	966		807	966	
	ND	546	631	45,85%	546	631	45,85%
Convicção = 1,20	D	923	850		923	850	
	ND	650	527	53,33%	650	527	53,33%
Convicção = 1,30	D	1004	769		1004	769	
	ND	708	469	58,04%	708	469	58,04%
Convicção = 1,40	D	1053	720		1053	720	
	ND	760	417	61,45%	760	417	61,45%
Convicção = 1,50	D	1079	694		1079	694	
	ND	785	392	63,18%	785	392	63,18%
Especificidade = 0,95	D	212	1561		212	1561	
	ND	155	1022	12,44%	155	1022	12,44%
Especificidade = 0,97	D	466	1307		466	1307	
	ND	311	866	26,33%	311	866	26,33%
Especificidade = 0,98	D	72	1051		72	1051	
	ND	48	1129	5,22%	48	1129	5,22%
Especificidade = 0,99	D	943	830		943	830	
	ND	633	544	53,43%	633	544	53,43%
Especificidade = 1,00	D	1048	725		1048	725	
	ND	704	473	59,38%	704	473	59,38%
Lift = 1,00	D	0	1773		0	1773	
	ND	0	1177	0,00%	0	1177	0,00%
Lift = 1,10	D	359	1414		359	1414	
	ND	256	921	20,86%	256	921	20,86%
Lift = 1,20	D	429	1344		429	1344	
	ND	305	872	24,89%	305	872	24,89%
Lift = 1,30	D	436	1337		436	1337	
	ND	310	867	25,30%	310	867	25,30%
Lift = 1,40	D	449	1324		449	1324	
	ND	320	857	26,06%	320	857	26,06%
Novidade = 0,00	D	0	1773		0	1773	
	ND	0	1177	0,00%	0	1177	0,00%
Novidade = 0,10	D	1773	0		1773	0	
	ND	1177	0	100,00%	1177	0	100,00%
Novidade = 0,20	D	1773	0		1773	0	
	ND	1177	0	100,00%	1177	0	100,00%
Novidade = 0,30	D	1773	0		1773	0	
	ND	1177	0	100,00%	1177	0	100,00%
Novidade = 0,40	D	1773	0		1773	0	
	ND	1177	0	100,00%	1177	0	100,00%

Tabela 7.13: Tabela de desempenho sobre a base de dados Labor

Métrica		CRg e CRm			TRg e TRm		
Medida		F	NF	% elim	F	NF	% elim
Convicção = 1,10	D	326	104		333	97	
	ND	352	9	85,71%	345	16	85,71%
Convicção = 1,20	D	326	104		333	97	
	ND	354	7	85,97%	347	14	85,97%
Convicção = 1,30	D	327	103		333	97	
	ND	354	7	86,09%	348	13	86,09%
Convicção = 1,40	D	329	101		335	95	
	ND	354	7	86,35%	348	13	86,35%
Convicção = 1,50	D	337	93		340	90	
	ND	354	7	87,36%	349	12	87,10%
Especificidade = 0,95	D	15	415		14	416	
	ND	4	357	2,40%	5	356	2,40%
Especificidade = 0,97	D	66	364		59	371	
	ND	6	355	9,10%	13	348	9,10%
Especificidade = 0,98	D	104	326		97	333	
	ND	11	350	14,54%	18	343	14,54%
Especificidade = 0,99	D	104	326		97	333	
	ND	11	350	14,54%	18	343	14,54%
Especificidade = 1,00	D	104	326		97	333	
	ND	11	350	14,54%	18	343	14,54%
Lift = 1,00	D	0	430		0	430	
	ND	0	361	0,00%	0	361	0,00%
Lift = 1,10	D	5	425		4	426	
	ND	4	357	1,14%	5	356	1,14%
Lift = 1,20	D	95	335		95	335	
	ND	61	300	19,72%	61	300	19,72%
Lift = 1,30	D	104	326		103	327	
	ND	62	299	20,99%	63	298	20,99%
Lift = 1,40	D	183	247		183	247	
	ND	118	243	38,05%	118	243	38,05%
Novidade = 0,00	D	0	430		0	430	
	ND	0	361	0,00%	0	361	0,00%
Novidade = 0,10	D	430	0		430	0	
	ND	361	0	100,00%	361	0	100,00%
Novidade = 0,20	D	430	0		430	0	
	ND	361	0	100,00%	361	0	100,00%
Novidade = 0,30	D	430	0		430	0	
	ND	361	0	100,00%	361	0	100,00%
Novidade = 0,40	D	430	0		430	0	
	ND	361	0	100,00%	361	0	100,00%

Tabela 7.14: Tabela de desempenho sobre a base de dados Adult

Métrica		CRg e CRm			TRg e TRm		
Medida		F	NF	% elim	F	NF	% elim
Convicção = 1,10	D	49	54		49	54	
	ND	5	7	46,96%	5	7	46,96%
Convicção = 1,20	D	57	46		57	46	
	ND	8	4	56,52%	8	4	56,52%
Convicção = 1,30	D	63	40		63	40	
	ND	10	2	63,48%	10	2	63,48%
Convicção = 1,40	D	69	34		69	34	
	ND	10	2	68,70%	10	2	68,70%
Convicção = 1,50	D	72	31		72	31	
	ND	10	2	71,30%	10	2	71,30%
Especificidade = 0,95	D	8	95		8	95	
	ND	11	1	16,52%	11	1	16,52%
Especificidade = 0,97	D	24	79		24	79	
	ND	12	0	31,30%	12	0	31,30%
Especificidade = 0,98	D	48	55		48	55	
	ND	12	0	52,17%	12	0	52,17%
Especificidade = 0,99	D	66	37		66	37	
	ND	12	0	67,83%	12	0	67,83%
Especificidade = 1,00	D	77	26		77	26	
	ND	12	0	77,39%	12	0	77,39%
Lift = 1,00	D	18	85		18	85	
	ND	2	10	17,39%	2	10	17,39%
Lift = 1,10	D	57	46		57	46	
	ND	10	2	58,26%	10	2	58,26%
Lift = 1,20	D	66	37		66	37	
	ND	11	1	66,96%	11	1	66,96%
Lift = 1,30	D	72	31		72	31	
	ND	12	0	73,04%	12	0	73,04%
Lift = 1,40	D	75	28		75	28	
	ND	12	0	75,65%	12	0	75,65%
Novidade = 0,00	D	18	85		18	85	
	ND	2	10	17,39%	2	10	17,39%
Novidade = 0,10	D	103	0		103	0	
	ND	12	0	100,00%	12	0	100,00%
Novidade = 0,20	D	103	0		103	0	
	ND	12	0	100,00%	12	0	100,00%
Novidade = 0,30	D	103	0		103	0	
	ND	12	0	100,00%	12	0	100,00%
Novidade = 0,40	D	103	0		103	0	
	ND	12	0	100,00%	12	0	100,00%

Para avaliar o efeito da aplicação do modelo SemPrune, quando a inclusão do conhecimento de mundo ocorrer no pós-processamento, foi montada a Tabela 7.15, na qual são mostrados os números de regras mineradas, de eliminadas pelo modelo e de remanescentes. A precisão é definida como a razão entre o número de regras relevantes recuperadas e o de regras recuperadas. Não tendo havido diferença no número de regras relevantes recuperadas pela aplicação do modelo, eventuais ganhos originam-se na redução do denominador. Para a base de dados Adult, o ganho obtido foi de 2924/2409 – 1 = 0,2138 ou 21,38%.

Tabela 7.15 Ganho de Precisão do modelo SemPrune

Base de Dados	Regras mineradas	Regras eliminadas	Regras remanescentes	Ganho de precisão
Adult	2924	515	2409	21,38%
Stulong	34422	8163	26259	31,09%
Labor	181229	32874	148355	22,16%

7.2 MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO COM INCLUSÃO DO CONHECIMENTO DE MUNDO NO PRÉ-PROCESSAMENTO

Segue-se a descrição do que se obteve aplicando o modelo sobre a base de dados descrita no item 6.1.3.

Foram selecionados alguns exemplos de regras liminarmente eliminadas por apresentarem as formas das expressões 5.2.1.1, 5.2.1.2 e 5.2.1.3 (X ^ C $\Rightarrow$Y, X ^ Y ^ A $\Rightarrow$ B e X ^ A $\Rightarrow$ Y ^ B,. respectivamente). A redundância tipo 1 (expressão 5.2.1.1) é caracterizada por uma regra contendo um atributo determinante, em *lhs* e um atributo determinado, em *rhs*. A de tipo 2 (expressão 5.2.1.2) é caracterizada por duas regras, com o mesmo conseqüente, a primeira contendo um atributo determinante em *lhs* e a outra com ambos os atributos, determinante e determinado em *lhs*. A de tipo 3 (expressão 5.2.1.3) é caracterizada por duas regras, a primeira com um atributo determinante em *lhs* e um certo *rhs*. A segunda regra tem o mesmo

antecedente da primeira, e em *rhs* existe, além do que era o *rhs* anterior, mais o atributo determinado pelo determinante de *lhs*.

Na primeira linha da tabela 7.16 encontra-se a regra de **Id** 74, da forma da expressão 5.2.1.1, com a seguinte interpretação: Camembert Pierrot determina Dairy Products não precisando de Alice Mutton ou de Ikura. Logo é redundante. Na terceira linha, verifica-se que Guaraná Fantástica e Ikura determinam Alice Mutton. Em conseqüência, a regra de **Id** 707, tendo, no antecedente estes atributos e ainda mais o atributo Beverages, determinado por Guaraná Fantástica, permanecendo o mesmo conseqüente é redundante.

Tabela 7.16: Regras da base Northwind Traders nas formas das expressões 5.2.1.1, 5.2.1.2 e 5.2.1.3

Regras na forma das expressões 5.2.1.1, 5.2.1.2 e 5.2.1.3 podadas				
Regra ID	Antecedente	Conseqüente	Sup	Conf
74 5.2.1.1	(Alice Moutton), (Ikura)., (Camembert Pierrot)	(Dairy Products)	0,10	1,00
341 5.2.1.1	(Alice Moutton), (Guaraná Fantastica)	(Beverages)	0,10	0,64
709 5.2.1.2 regra 1	(Guaraná Fantastica), (Ikura)	(Alice Moutton)	0,10	0,69
707 5.2.1.2 regra 2	(Guaraná Fantastica), (Ikura), (Beverages)	(Alice Moutton)	0,10	0,90
2444 5.2.1.2 regra 1	(Gorgonzola Telino), (Boston Crab Meat), (Chang)	(Camembert Pierrot)	0,13	0,61
2441 5.2.1.2 regra 2	(Gorgonzola Telino), (Boston Crab Meat), (Dairy Products), (Chang	(Camembert Pierrot)	0,13	1,00
104 5.2.1.3 regra 1	(Alice Moutton), (Chai)	(Camembert Pierrot)	0,10	0,75
105 5.2.1.3 regra 2	(Alice Moutton), (Chai)	(Camembert Pierrot), (Beverages)	0,10	0,75
593 5.2.1.3 regra 1	(Guaraná Fantastica) (Ikura), (Camembert Pierrot)	(Alice Moutton)	0,10	0,90
594 5.2.1.3 regra 2	(Guaraná Fantastica) (Ikura), (Camembert Pierrot)	(Alice Moutton), (Dairy Products)	0,10	0,90

Nos 876 processos de generalização foram eliminadas 5038 regras. Serão descritos, a seguir, alguns exemplos de generalização.

As tabelas 7.16 e 7.17 mostram dois exemplos de generalização. Em ambos, o conseqüente de todas as regras é comum "Dairy Products". No primeiro exemplo, a regra mais geral, referente ao atributo **Beverages**, possui vinte e quatro regras

mais específicas, com o atributo contendo bebidas. O valor de CRg calculado foi de 73%. Esta medida é superior ao valor especificado para o ponto de corte (70%), significando que a regra com o atributo **Beverages** pode substituir as mais especializadas.

Nos dois exemplos que se seguem, o conseqüente de todas as regras, tanto das mais especializadas quanto das mais abstratas, é comum "Dairy Products".

Tabela 7.17: Processo de generalização baseado na relação de dependência taxonômica

Processo de generalização com CRg = 0,73 e CRm = 0,12			
Regra ID	Antecedente	Sup	Conf
Regras mais especializadas a serem podadas			
3595	Ipoh Coffee,Camembert Pierrot,Chai	0,03	1,00
3857	Ipoh Coffee,Camembert Pierrot,Chai	0,03	1,00
4038	Ipoh Coffee,Camembert Pierrot,Beverages	0,03	1,00
4342	Lakkalikoori,Camembert Pierrot,Chang	0,02	1,00
4355	Lakkalikoori,Camembert Pierrot,Raclette Courdavault,Chang	0,02	1,00
4481	Ipoh Coffee,Camembert Pierrot,Chef Anton's Cajun Seasoning	0,03	1,00
5192	Ipoh Coffee,Camembert Pierrot	0,01	1,00
5274	Lakkalikoori,Camembert Pierrot	0,02	1,00
5277	Lakkalikoori,Camembert Pierrot,Pavlova	0,03	1,00
5285	Lakkalikoori,Camembert Pierrot,Pavlova,Raclette Courdavault	0,03	1,00
5298	Lakkalikoori,Camembert Pierrot,Raclette Courdavault	0,03	1,00
6408	Chai	0,02	0,89
6413	Chai,Chang	0,04	0,77
6419	Chai,Chef Anton's Cajun Seasoning	0,03	1,00
6549	Ipoh Coffee,Chai	0,03	0,89
6586	Rhonbrau Klosterbier ,Chai	0,03	0,84
6592	Chai,Steeleye Stout	0,02	0,83
6598	Chai,Tarte au sucre	0,03	0,75
6604	Chai,Thuringer Rostbratwurst	0,03	0,89
6607	Chai,Chang	0,01	0,77
6609	Chai,Chef Anton's Cajun Seasoning	0,03	1,00
6655	Ipoh Coffee,Chai	0,03	0,86
6666	Rhonbrau Klosterbier ,Chai	0,03	0,93
6668	Chai,Steeleye Stout	0,01	0,83
Regra mais geral que representa as regras especializadas podadas			
6673	Beverages	0,65	0,70

No segundo exemplo, a regra mais geral, referente ao atributo **Beverages**, possui seis regras mais específicas, com o atributo contendo bebidas. O valor de CRg calculado foi de 75%. Esta medida ultrapassa a confiança mínima especificada, significando que a regra com o atributo **Beverages** pode substituir as mais especializadas.

Tabela 7.18: Mais um processo de generalização baseado na relação de dependência taxonômica

Processo de generalização com CRg = 0,75 e CRm = 0,43			
Regra ID	Antecedente	Sup	Conf
Regras mais especializadas a serem podadas			
3597	Ipoh Coffee,Camembert Pierrot,Beverages	0,07	1,00
3858	Ipoh Coffee,Camembert Pierrot	0,07	0,90
6415	Beverages, Chang	0,08	0,77
6551	Ipoh Coffee,Beverages	0,07	0,69
6588	Rhonbrau Klosterbier ,Beverages	0,07	0,64
6594	Beverages,Steeleye Stout	0,08	0,83
Regra mais geral que representa as regras especializadas podadas			
6410	Beverages	0,52	0,70

Os processos de especialização, por sua vez, permitiram a eliminação de 7 regras mais gerais.

Na tabela 7.19, pode-se ver um exemplo de processos de especialização com base na relação taxonômica.

Nestes exemplos, a medida CRg possui um valor abaixo do valor de confiança mínima do usuário, o que não permite generalização. Continuando a tentativa, vê-se que o valor de CRm também é inferior à confiança mínima do usuário, mostrando que a regra mais geral não é válida, o que vai ocorrer sempre que o número de regras que alcançaram o patamar de mineração, sem utilizar o suporte do agregador, seja pequeno.

Tabela 7.19: Processo de especialização baseado na relação de dependência taxonômica

Processo de especialização com CRg = 0,66 e CRm = 0,66				
Regra ID	Antecedente	Consequente	Sup	Conf
Regras mais geral que não cobre as regras especializadas				
4928	(Beverages), (Dairy Products)	(Camembert Pierrot), (Chai)	0,27	0,60
Regras mais especializadas que subsistem				
3448	(Steeleye Stout), (Chang)	(Camembert Pierrot), (Chai)	0,09	0,84
3601	(Ipoh Coffee), (Dairy Products)	(Camembert Pierrot), (Chai)	0,08	0,92
3448	(Rhonbrau Klosterbier), (Dairy Products)	(Camembert Pierrot), (Chai)	0,09	0,85

7.2.1.1 Resumo dos resultados

Um resumo dos resultados obtidos pelo método proposto pode ser visto na tabela 7.20, similar à 7.8.

Tabela 7.20: Resumo dos resultados obtidos pelo método SemPrune

Base de Dados	Regras mineradas	Regras eliminadas	Eliminadas/Total
Northwind Traders	9941	5052	50,82%

Observa-se que o filtro semântico atuou como esperado. A taxa de redução foi muito boa, ultrapassando 50%.

A Tabela 7.23 mostra os dados de refutação da hipótese nula, para o caso da inclusão do conhecimento de mundo no pré-processamento das regras de associação mineradas, bem como o ganho em "recall" obtido pela aplicação do modelo SemPrune. Na base de dados utilizada não houve nenhuma poda desnecessária de regra, e a quantidade de informação recuperada diminuiu. O ganho exibido foi de 63%. Todavia, ao contrário do ganho indiscutível, mostrado na Tabela 7.15, este é ilusório, pois calculado como se todas as regras de associação incluídas no conjunto de regras mineradas fosse relevante, o que o modelo não pode afirmar, sem

intervenção humana. Mesmo assim, é absolutamente certo que tendo sido introduzida informação relevante recuperada, o "recall" aumenta, embora em números mais modestos.

7.2.1.2 Análise comparativa entre medidas de interesse objetivas

Tal como nos relacionamentos do tipo A, a verificação da melhora da qualidade do produto obtido no uso destas técnicas serve de comparação entre este trabalho e as técnicas sintáticas usuais.

Processos de generalização/especialização produzem regras mais relevantes, pois acompanham o processo intuitivo humano.

Mais uma vez, o teste desta hipótese pode ser feito comparando-se resultados assim gerados, com a filtragem de pós-processamento, orientado por medidas de interesse objetivas, especificamente convicção, especificidade, lift e novidade, todas descritas no item 4.4.1.

A aplicação dessas quatro medidas objetivas será feita usando-se cinco valores distintos, para cada medida objetiva de interesse, como ponto de corte. A Tabela 7.21 mostra o resultado, e também que os filtros objetivos deixam passar regras redundantes que podem ser detectadas por filtros semânticos. Confirma-se, nesta experiência, que os filtros objetivos têm um ajuste muito sensível, de forma que pequenas variações do ponto de corte apresentam resultados significativamente diferentes. Tal comportamento exige muito do usuário, o que, em geral, leva à baixa qualidade do resultado. Uma variação de apenas 0,02 no valor de corte de especificidade (de 0,97 para 0,99) causou uma variação de 58,0% no número de regras eliminadas, de 0,1 (de 0,00 para 0,10), para a novidade, e de 5294 para 97 no número de regras eliminadas.

Tabela 7.21: Resultados obtidos pelas medidas de interesse com a base Northwind Traders

Métricas		CRg e CRm			TRg e TRm		
Medida	Corte	% Regras eliminadas	#Regras deixadas	SemPrune % regras eliminadas	% Regras eliminadas	#Regras deixadas	SemPrune % regras eliminadas
Convicção	1,10	16,75%	4407	100,00%	16,50%	4515	83,01%
	1,20	16,50%	4407	100,00%	16,50%	4515	83,01%
	1,30	16,75%	4407	90,04%	16,50%	4515	83,01%
	1,40	17,06%	4391	90,00%	16,79%	4499	83,04%
	1,50	17,32%	4377	89,97%	17,05%	4485	83,03%
Especificidade	0,95	59,14%	2163	98,15%	59,20%	2206	84,27%
	0,97	74,05%	1374	98,11%	74,31%	1389	80,49%
	0,98	76,75%	1231	91,47%	77,07%	1240	71,37%
	0,99	83,25%	887	90,64%	83,50%	892	69,06%
	1,00	83,25%	887	85,01%	83,50%	892	62,89%
Lift	1,00	0,00%	5294	89,20%	0,00%	5407	79,67%
	1,10	0,00%	5294	89,20%	0,00%	5407	79,67%
	1,20	0,00%	5294	89,20%	0,00%	5407	79,67%
	1,30	0,00%	5294	89,20%	0,00%	5407	79,67%
	1,40	0,57%	5264	89,13%	0,55%	5377	79,69%
Novidade	0,00	0,00%	5294	89,20%	0,00%	5407	79,67%
	0,10	98,17%	97	100,00%	98,17%	99	100,00%
	0,20	99,92%	4	100,00%	99,93%	4	100,00%
	0,30	100,00%	0	0,00%	100,00%	0	0,00%
	0,40	100,00%	0	0,00%	100,00%	0	0,00%

As matrizes de desempenho, utilizadas no item 7.1.1.5, para inclusão do conhecimento de mundo no pré-processamento, são exibidas na Tabela 7.22, que comprova o diferencial da proposta semântica, objeto da presente pesquisa.

Tabela 7.22: Resultados obtidos pelas medidas de interesse com a base Northwind Traders

Métricas		CRg e CRm		TRg e TRm	
Medida		F	NF	F	NF
Convicção = 1,10	D	754	3968	561	3747
	ND	133	439	331	768
Convicção = 1,20	D	754	3968	561	3747
	ND	133	439	331	768
Convicção = 1,30	D	754	3968	561	3747
	ND	133	439	331	768
Convicção = 1,40	D	770	3952	572	3736
	ND	133	439	336	763
Convicção = 1,50	D	784	3938	584	3724
	ND	133	439	338	761
Especificidade = 0,95	D	2821	1901	2671	1637
	ND	310	262	530	569
Especificidade = 0,97	D	3535	1187	3353	955
	ND	385	187	665	434
Especificidade = 0,98	D	3658	1064	3454	854
	ND	405	167	713	386
Especificidade = 0,99	D	3968	754	3747	561
	ND	439	133	768	331
Especificidade = 1,00	D	3968	754	3747	561
	ND	439	133	768	331
Lift = 1,00	D	0	4722	0	4308
	ND	0	572	0	1099
Lift = 1,10	D	0	4722	0	4308
	ND	0	572	0	1099
Lift = 1,20	D	0	4722	0	4308
	ND	0	572	0	1099
Lift = 1,30	D	0	4722	0	4308
	ND	0	572	0	1099
Lift = 1,40	D	30	4692	23	4285
	ND	0	572	7	1092
Novidade = 0,00	D	0	4722	0	4308
	ND	0	572	0	1099
Novidade = 0,10	D	4641	81	4235	73
	ND	556	16	1073	26
Novidade = 0,20	D	4718	4	4304	4
	ND	572	0	1099	0
Novidade = 0,30	D	4722	0	4308	0
	ND	572	0	1099	0
Novidade = 0,40	D	4722	0	4308	0
	ND	572	0	1099	0

A avaliação do efeito da aplicação do modelo SemPrune, quando a inclusão do conhecimento de mundo ocorrer no pré-processamento, foi montada na Tabela 7.23, na qual são mostrados: o número de regras originalmente mineradas, de regras enriquecidas, obtidas pela aplicação do modelo SemPrune, o de regras eliminadas pelo modelo, e o número de regras remanescentes. O "recall" é definido como a razão entre o número de regras relevantes recuperadas e o número de regras relevantes. Com a inclusão do conhecimento de mundo, aumentou muito o número de regras recuperadas. Não tendo havido diferença no número de regras relevantes, eventuais ganhos originam-se na ampliação do numerador. Supondo que todas as regras obtidas pelo enriquecimento semântico fossem relevantes, o resultado obtido seria 9941/6070 – 1 = 0,6377 ou 63,77%. Evidentemente, a suposição de relevância está exagerada, mas a determinação real, da quantidade de regras relevantes revelada pela ontologia, só pode ser feita por inspeção humana. De qualquer forma, por menor que seja a fração de regras relevantes incluída pelo modelo SemPrune, ela será maior do que zero e, em conseqüência, o ganho em "recall" não será nulo. A precisão, sendo a razão entre o número de regras relevantes recuperadas e o número de regras recuperadas, cresce neste experimento, pois o número de regras remanescentes é menor do que o de regras originais. Todavia, este ganho não pode ser generalizado, pois é dependente da ontologia. Sempre que o número de atributos que possuam ancestrais seja elevado, na base de dados, aumenta a probabilidade de poda (muito alta no caso da base de dados Northwind Traders). Não obstante, a hipótese alternativa fixava apenas o ganho de "recall", o que foi alcançado.

Tabela 7.23 Ganho de "recall" do modelo SemPrune

Base de Dados	Regras originais	Regras enriquecidas	Regras eliminadas	Regras remanescentes	Ganho de recall
Northwind Traders	6070	9941	5052	4889	63,77%

8 CONCLUSÕES E FUTURO DAS PESQUISAS

Os seres humanos, quando pesquisam dados para tomadas de decisão, buscam a descoberta de configurações que façam sentido para o pesquisador. Estas configurações são algo do tipo classe social de um consumidor, esbeltez de uma pessoa, qualidade de vida de uma vizinhança, escola mais adequada para matrícula de filhos. Usualmente, as bases de dados legadas não registram explicitamente estas configurações ou atributos que são decisivos para tomada de decisão.

A pesquisa direcionou-se à verificação de se o conhecimento de mundo, presente nas ontologias, tornaria possível extrair o valor desses atributos das bases de dados e incluí-los nas regras de associação mineradas sobre estas bases. Ainda mais se desejava demonstrar que este conhecimento permitiria reduzir o número de regras de associação mineradas, podando aquelas que, sob o ponto de vista do conhecimento do mundo, fossem redundantes.

Para desenvolver a pesquisa foi necessário estudar as alternativas sintáticas de poda de regras de associação e montar um ambiente de testes.

O conhecimento de mundo foi incorporado à análise das regras de associação, por meio dos relacionamentos entre atributos de dados dos domínios dos tipos "é-um" e "parte-de". Neste estudo, evidenciou-se que, dentro de cada tipo de relacionamento, havia diferença entre cardinalidades de relacionamentos 1:n e m:n.

Um problema que teve de ser estudado foi quando deveria ser feito o enriquecimento de informação, se no pré-processamento, durante o processamento, ou no pós-processamento. Constatou-se que o enriquecimento no pós-processamento poderia trazer todos os resultados esperados e deveria ser o caminho preferencial.

A investigação em curso mostrou situações, já descritas na literatura [SRI95], nas quais a agregação de informação e o enriquecimento deveriam ser feitos no pré-processamento da mineração de regras. Isto se devia ao fato de pequenos suportes, de itens individuais não agregados, não atingirem os patamares mínimos de suporte de itens freqüentes.

A pesquisa revelou que a diferença de tratamento entre os tipos de relacionamento e suas cardinalidades tem pouca importância, pois o modelo aplicado pode ser o mesmo, com pequenas modificações na representação da ontologia do domínio, escolhendo-se árvores de preferências dentre os grafos direcionados, característicos das cardinalidades m:n. Não obstante, a ocasião de inclusão do conhecimento de mundo mostrou ser de grande importância, obrigando ao desenvolvimento de dois tipos paralelos de algoritmos, um para o enriquecimento de informações no pós-processamento e outro, no pré-processamento.

Constatou-se que os métodos sintáticos de poda existentes são, absolutamente, ineficazes para detectar os relacionamentos semânticos, trazidos pelo conhecimento de mundo e, portanto, quando aplicados a conjuntos de regras de associação enriquecidas têm comportamento aleatório, na tomada de decisão de podá-las ou não.

Foi desenvolvido um modelo, denominado SemPrune, que centralizou a proposta de linha de ação adotada no trabalho.

O modelo SemPrune apresenta-se em duas versões, uma para tratamento de bases de dados, com enriquecimento no pós-processamento, e outra, no pré-processamento.

As hipóteses e propostas foram validadas em experimentos realizados com diversas bases de dados, com resultados bastante satisfatórios. As taxas de redução do conjunto de regras de associação para bases de dados, com enriquecimento no pós-processamento, variaram de 210% até 34%. O tratamento no pós-processamento não perdeu nenhuma regra relevante, mas reduziu o número de regras recuperadas, ampliando a precisão. Para a base de dados de transações, com enriquecimento no pré-processamento, a taxa de redução do conjunto de regras foi de 50,00%. Este valor parece muito melhor, mas não se pode esquecer que, em relações taxonômicas, todos os itens possuem ancestrais, o que aumenta, em muito, o número de regras de associação virtuais criadas. Neste tipo de aplicação, o modelo SemPrune ampliou o "recall", por manter a informação relevante e ampliar a recuperação por meio da informação oculta.

Pelo exposto, pode-se rejeitar a hipótese nula, aceitando-se a hipótese alternativa.

Inequivocamente, métodos sintáticos conseguem, variando os pontos de corte, podar um número muito maior de regras do que o método semântico deste trabalho, todavia eliminam regras que não deveriam ser eliminadas e deixam de eliminar as que deveriam.

O modelo SemPrune não tem a liberdade de variação de parâmetros de definição do ponto de corte como os métodos sintáticos. Pode-se filtrar regras de associação com corte, correspondendo a 1,00 de lift ou a 1,40 de lift, por exemplo, pois trata-se de valores estatísticos e probabilísticos. Contudo, para parâmetros de corte semântico, essa flexibilidade é mínima. A poda semântica ocorre apenas, quando existem diversas regras de associação com o mesmo conseqüente, e com antecedentes especializando uma regra mais geral. Se o número dessas regras específicas, para cada regra geral for pequeno, não adianta alterar valor de parâmetro algum, pois não é possível podar mais.

8.1 CONTRIBUIÇÃO

A maior contribuição do modelo é mostrar que o enriquecimento semântico das regras de associação mineradas pode ser tratado algoritmicamente, não necessitando ser realizado na mente do analista. Ficou, ainda, evidenciado que os métodos de poda sintática podem desfazer o enriquecimento semântico. Os experimentos demonstraram que a diferença de comportamento entre as diversas medidas de interesse objetivas, no tocante às dicotomias "devia ter sido podada/não devia ter sido podada" e "foi podada/não foi podada" é muito grande e que, para uma mesma medida, pequenos valores do parâmetro de corte apresentam variações de desempenho muito longe da linearidade e/ou previsibilidade.

Resumindo:

- a poda semântica só tem sentido se for aplicada após um enriquecimento semântico das regras de associação;

- os métodos sintáticos não conseguem sequer perceber o enriquecimento semântico e, quando realizam poda, podem podar o que não deveria sê-lo e deixar de podar o que poderia;
- os métodos semânticos, quando aplicáveis, podem reduzir substancialmente o número de regras de associação mineradas.

Curiosamente, constatou-se que as duas distintas classes de métricas, utilizadas para verificar a conveniência da generalização de regras (CRg e CRm ou TRg e TRm), têm comportamento extremamente parecido. O valor da geração dessas métricas foi apenas para conhecimento acadêmico, tendo em vista que para um usuário final, a adoção de qualquer par de medidas será pouco diferente da opção de estimar uma distância, em quilômetros ou milhas, por exemplo. As tabelas 7.13 a 7.27 e 7.25 a 7.26 ilustram esta constatação.

8.2 LIMITAÇÕES DA PESQUISA

A primeira limitação do trabalho refere-se ao seu escopo. Foram considerados apenas dois casos do uso de ontologias que envolvam tipos específicos de relações:

as de generalização-especialização do tipo "é-um",

as de composição do tipo "parte-de".

Os recursos computacionais exigidos pelo modelo SemPrune são intensivos em uso de memória, pelo menos na implementação experimental. O módulo gerador de regras mais abrangentes classifica todas as regras disponíveis pelo conseqüente antes de gerar os agrupamentos de regras mais gerais e as respectivas mais específicas. As estruturas de dados utilizadas para armazenamento das regras são pesadas, compreendendo os itens, as medidas objetivas de interesse e as tabelas de dependência obtidas do conhecimento de mundo. Para conjuntos de 181.229 regras, como é o caso da base de dados Labor, a classificação interna (na qual toda a coleção de objetos está presente na memória principal) demora 15 minutos em uma máquina Intel(R) Core(TM) 2 CPU com 1.86 GHz e 1Gb de RAM. Isto é uma

limitação, se considerarmos que caso as regras escolhidas tivessem, por exemplo, sete itens, em vez de cinco, a explosão combinatória tornaria inviável o processamento sem usar classificação externa (na qual a coleção de objetos não está toda presente na memória principal, utilizando memória secundária auxiliar). Os algoritmos de classificação externa são bem conhecidos, mas não foram empregados no experimento realizado.

Uma limitação do trabalho é o fato de resultados satisfatórios serem totalmente dependentes da existência de ontologia do domínio. Além disso, existe o trabalho adicional de adaptação da ontologia existente, ao padrão de dependência esperado pela implementação do modelo SemPrune.

Finalmente, o enriquecimento do conjunto de regras, na etapa de pré-processamento, aumenta o conjunto de regras de associação mineradas. O "recall" melhora, mas o mesmo não se pode afirmar quanto à precisão. Por esta razão, só deve ser utilizado quando houver perda de informação minerada por baixo suporte, pelos níveis mais baixos de uma hierarquia taxonômica e, mesmo assim, enquanto não for desenvolvido um algoritmo para trazer o enriquecimento para o processo de mineração das regras de associação.

8.3 TRABALHOS FUTUROS

Esta pesquisa atingiu plenamente seus objetivos, mostrando os ganhos obtidos pela utilização do conhecimento de mundo no enriquecimento dos conjuntos de regras de associação mineradas e na poda semântica. Os resultados mais expressivos foram obtidos quando a aplicação dos conhecimentos de mundo ocorria na etapa de pós-processamento.

A utilização exclusiva da inclusão dos conhecimentos semânticos nesta etapa, enfrenta como obstáculo, o fato de que, em algumas situações, itens semelhantes apresentam suporte acima de um patamar de corte, quando considerados como categoria, mas têm suporte baixo para cada tipo particular. Este tipo de ocorrência faz com que os itens individuais não sejam considerados freqüentes e, portanto,

ignorados pelos algoritmos de mineração de regras de associação. Na etapa de pós-processamento, o enriquecimento do conjunto de regras não vai encontrar os itens individuais para poder agrupá-los em regras mais gerais.

Contornou-se esta restrição fazendo o enriquecimento do conjunto de regras na etapa de pré-processamento. O resultado foi satisfatório, quanto ao enriquecimento do conjunto de regras, permitindo que não se perdesse a informação advinda da existência de regras, inicialmente ocultas, contendo itens agregadores, oriundos da ontologia do domínio.

Não obstante, ao fazer o enriquecimento do conjunto de regras na etapa de pré-processamento, o número de atributos da base de transações cresce, surgindo muitas regras redundantes. Este fato foi parcialmente enfrentado, criando-se mecanismos adicionais de poda de redundâncias. Mesmo assim, o conjunto de regras de associação mineradas cresceu.

Uma linha de pesquisa, aparentemente promissora, consiste na criação de um novo algoritmo de mineração de regras de associação, que faça o enriquecimento do conjunto de regras mineradas, não na etapa de pré-processamento nem na de pós-processamento, mas na própria etapa de mineração.

Este algoritmo poderia ser uma variação do clássico Apriori. O algoritmo começaria analisando os itens isolados, cujo suporte alcançasse o patamar especificado para seu enquadramento como freqüente. Depois disto, os conjuntos de itemsets freqüentes iriam aumentando de tamanho, enquanto possível, sendo os itens não freqüentes desconsiderados.

A novidade do algoritmo seria a manutenção, em paralelo, dos conjuntos de itens freqüentes e de itens potencialmente freqüentes, à luz da ontologia de domínio. Este novo algoritmo poderia estender toda a potencialidade do modelo SemPrune aos domínios de povoamento, esparso nas folhas das hierarquias.

Quanto às aplicações futuras do modelo SemPrune, acreditamos que o cenário mais promissor seja o de seu emprego nas bases de dados com relacionamentos do tipo "é-um" 1:n, com enriquecimento no pré-processamento, nas quais existe uma enorme área de aplicações biológicas, particularmente a área de DNA, que parece ser um terreno extremamente fértil para este tipo de aplicação.

9 REFERÊNCIAS

[ABL03] Ableson, A., Glasgow, J. Efficient statistical pruning of association rules. *In*: **PKDD 2003 : Knowledge Discovery in Databases** (Cavtat-Dubrovnik, 22-26 September 2003) - European conference on principles and practice of knowledge discovery in databases No7, Cavtat-Dubrovnik , CROATIE (22/09/2003) 2003, vol. 2838, pp. 23-34[Note(s) : XVI, 508 p.,] [Document : 12 p.] (21 ref.) ISBN 3-540-20085-1

[ADA01] ADAMO, J. M. **Data Mining for Association Rules and Sequential Patterns.** [S.l.]: NY: Springer-Verlag, 2001.

[AGR93] Agrawal, R.; Imielinski, T.; Swami, A. Mining association rules between sets of items in large databases. In: BUNEMAN, P.; JAJODIA, S. (Ed.). **Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data**. Washington, D.C.: [s.n.],1993. p. 207–216.

[AGR94] AGRAWAL, R.; SRIKANT, R. Fast algorithms for mining association rules. *In*: BOCCA, J. B.; JARKE, M.; ZANIOLO, C. (Ed.). **Proc. 20th Int. Conf. Very Large Data Bases**, VLDB. Morgan Kaufmann, 1994. p. 487–499. ISBN 1-55860-153-8. Disponível em: <http://citeseer.ist.psu.edu/article/agrawal94fast.html>.

[ALD84] Aldenderfer, M. S.,. Blashfield, R. K. Cluster analysis. Series title: **Sage university papers series. Quantitative applications in the social sciences no. 07-044**. Beverly Hills: Sage Publications, c1984.

[BAR05] Bart Goethals, B., Muhonen, J., Toivonen, H. Mining Non-Derivable Association Rules. *In*: **2005 SIAM International Conference on Data Mining** – Newport Beach, 2005.

[BAS00] BASTIDE, Y. et al. **Mining minimal non-redundant association rules using frequent closed itemsets**. 2000. Disponível em: <http://citeseer.ifi.unizh.ch/article/bastide00mining.html>.

[BET96] Bettini, C., Wang, X., Jajodia, S. Testing Complex Temporal Relationships Involving Multiple Granularities and Its Application to Data Mining. *In*: **Proceedings of the Fifteenth ACM SIGACT-SIGMODSIGART. Symposium on Principles of Database Systems (PODS'96)**, pages 68 78, Montreal, Canada, June 1996.

[BRA83] Brachman, R. 1983. What IS-A Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks. **IEEE Computer**, 16(10): 30-36.

[BRI97] Brin, S. et al. Dynamic itemset counting and implication rules for market basket data. In: Peckham, J. (Ed.). **SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data**, May 13-15, 1997, Tucson, Arizona, USA. ACM Press, 1997. p. 255–264. Disponível em: <http://citeseer.ist.psu.edu/brin97dynamic.html>.

[BUR06] Bürkle, P. **Um Método de Pós-processamento de Regras de Associação com Base nas Relações de Dependência entre os Atributos**. Dissertação (Mestrado) — Instituto de Computação, Universidade Federal Fluminense, Niterói, 2006.

[CAV98] CAVOUKIAN, Ann. Data Mining: Stanking a Clain on Your Privacy. 1998. Disponível em: <http://www.ipc.on.ca/web_site.eng/MATTERS/SUM_PAP/PAPERS/datamine.htm>.

[CHA07] CHAPOUTO , R. **Association Rules Enrichment. Identificação de fragmentos moleculares freqüentes utilizando conhecimento de domínio**. Dissertação (Mestrado em ENGENHARIA INFORMÁTICA E DE COMPUTADORES). Instituto Superior Técnico – Universidade de Lisboa. Lisboa, 2007.

[CHA88] Chaffin, R., Herrmann, D. (1988). The nature of semantic relations: a comparison of two approaches. *In*: Evens, Martha W. (ed.). **Relational Models of the Lexicon**. Cambridge University Press.

[CHE03] Chen, Y, Chiang, M., Ko, M. - Discovering time-interval sequential patterns in sequence databases *In:* **Expert Systems with Applications** 25 (2003) 343-354.

[CHE08] Chen,P.,Verma, R., Meininger J., Chan, W. **Semantic Analysis of Association Rules**. International Florida Artificial Intelligence Research Society Conference (FLAIRS) -AAAI Digital Library – 2008. Disponível em: <http://www.aaai.org/Library/FLAIRS/2008/flairs08-068.php>.

[CIO99] Ciocoiu, M., Nau, D. **Ontology-Based Semantics**. Dept. of Computer Science and Institute for Systems Research - University of Maryland - College Park – 1999 Disponível em: <http://www.nist.gov/psl/pubs/mihai.pdf>.

[DOM04a] DOMINGUES, M. A. **Generalização de regras de associação**. Dissertação (Mestrado)— USP - São Carlos, 2004. Instituto de Ciências Matemáticas e de Computação.

[DOM04b] DOMINGUES, M. A.; REZENDE, S. O. **Descrição de um algoritmo para generalização de regras de associação**. [S.l.], 2004. Relatório Técnico do ICMC/USP - Número 228.

[ECM04] ECML/PKDD 2004. **Discovery Challenge Homepage**. Disponível em: <http://lisp.vse.cz/challenge/ecmlpkdd2004/>.

[EVE93] Everitt, B. S., **Cluster analysis**. 3rd ed. London, E. Arnold; New York, Halsted Press, c1993.

[EVS01] Evsukoff,A.- **Análise de agrupamentos.** Relatório Técnico – ADDLabs – Niterói – 2001.

[FIO06] Fiot, C. **Extended Time Constraints for Generalized Sequential Patterns lirmm-00106897, version 1** - Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (LIRMM) - CNRS - MONTPELLIER France 2006.

[GAR06] Garcia, A. C. B.; Ferraz, I. N.; Cotrim, V. M. S. Web ontology to enable e-Citizenship. *In*: **2006 AAAI Spring Symposium the Semantic Web meets eGovernment**, 2006, California, USA. Anais, 2006.

[GAR09] Garcia, A. C. B.; Ferraz, Vivacqua, A. From Data to Knowledge Mining. *In*: **Artificial Intelligence for Engineering Design, Analysis and Manufacturing AIEDAM** Special Issue, Winter 2009, Vol.23, No.1

[GEL08] Geller, J. **What is an Ontology**. Semantic Web And Ontologies Lab - Department of Computer Science – New Jersey's Science & Technology University. Disponível em: <http://web.njit.edu/~geller/what_is_an_ontology.html>.

[GIR87] Jean-Yves Girard. Linear Logic. **Theoretical Computer Science**, 50:1--102, 1987

[GRA97] Gray, J., Bosworth, A., Layman, A., and Pirahesh, H. "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals," **Data Mining and Knowledge Discover**, Vol. 1, No. 1, 1997, pp. 29-53. Publisher: Institute of Electrical and Electronics Engineers, Inc.

[GRU93] Gruber, T. R. A translation approach to portable ontologies. **Knowledge Acquisition**, 5(2):199-220, 1993.

[GUA00] Guarino, N., Welty, C. Ontological Analysis of Taxonomic Relationships. *In*: Laender, A., and Storey, V., eds., **Proceedings of ER-2000: The InternationalConference on Conceptual Modeling**. October, 2000. Springer-Verlag LNCS

[HAH03] Hahn, U., & Schulz, S. (2003). Towards a broad-coverage biomedical ontology based on descriptionlogics. *In*: **Pacific Symposium on Biocomputing 2003**. World Scientific: New Jersey: 577–588. Disponível em: <http://psb.stanford.edu/psb-online/proceedings/psb03/hahn.pdf>

[HAM03] Hamilton, H. J. – **Knowledge Discovery in Databases**. Disponível em: <http://www2.cs.uregina.ca/~hamilton/courses/831/notes/ml/dtrees/4_dtrees2.html>. Acessado em 20/11/03

[HAN00] Han, J., Pei, J., and Yin, Y. Mining frequent patterns without candidate generation. *In*: Chen, W., Naughton, J. F., and Bernstein, P. A., editors, **Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data**, pages 1–12, Dallas, Texas, U.S.A. ACM Press

[HAN95] Han, J., Fu, Y. Discovery of Multiple-Level Association Rules from Large Databases. **VLDB**, 1995, 420-431

[HAW04] Hawkins, D. et al. Consumer Behavior: **Building Marketing Strategy.** 9. ed. [S.l.]: McGraw-Hill/Irwin, 2004.

[HEY90] Heylighen F. **Representation and Change. A Metarepresentational Framework for the Foundations of Physical and Cognitive Science**, (Communication & Cognition, Gent), 200 p, 1990.

[HIP00] Hipp, J., Guntzer, U., Nakhaeizadeh, G.: Algorithms for association rule mining -- a general survey and comparison. *In* : **Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, Boston, 2000.

[HOL58] Hollingshead, A., Redlich, F. **Social class and mental illness. A community study**. John Wiley e Sons Inc., New York -1958.

[HSU03] Hsu, P., Chen, Y., Lingb, C Algorithms for mining association rules in bag databases. *In:* **Information Sciences**. v. 166, Issues 1-4, 29 October 2004, Pages 31-47 2003 Elsevier Inc.

[JAG08] Jager, A. **Mining of frequent sets using pruning, based on background knowledge**. Dissertação (Mestrado). In: Document Server@UHasselt - Education - School voor Informatietechnologie: Eindverhandelingen 2008.

[JAR02] Jaroszewicz, S., Simovici, D. Pruning Redundant Association Rules Using Maximum Entropy Principle. *In*: **Lecture Notes on Computer Science** Springer Berlin / Heidelberg. Chen, M., Yu, P., B. Liu, B. (Eds.): PAKDD 2002, LNAI 2336, pp. 135–147, 2002.

[JOS99] Joshi, M., Karypis, G., Kumar, V. A Universal Formulation of Sequential Patterns - University of Minnesota - **Computer Science and Engineering Report Number: 99-021** – Minneapolis – 1999.

[KAU90] Kaufman, L., Rousseeuw, P. Finding groups in data: an introduction to cluster analysis. Series title: **Wiley series in probability and mathematical statistics**. New York Wiley, c1990.

[LAN87] Landis, T., Hei-imann, D., and Chaffin, R. Development differences in the comprehension of semantic relations. **ZeitschriflfiirPsychologie**, 195(2):129-139, 1987.

[LAV99] Lavrac, N.; Flach, P.; Zupan, B. **Rule evaluation meaures: a unifying view**. Dzeroski, S., Flach, P. (Eds.) ILP-99, LNAI 1634, pp. 174-185. Springer-Verlag Berlin Heidelberg 1999

[LIU98] Liu, B., Hsu, W., Ma, Y. "Integrating Classification and Association Rule Mining." **Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)**, New York, USA, 1998.

[MAC67] MacQueen, J. B. Some Methods for classification and Analysis of Multivariate Observations. **Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability.** Berkeley, University of California Press, 1:281-297 1967.

[MAS04] Masseglia, F., Poncelet, P., Teisseire, M. "Pre-Processing Time Constraints for Efficiently Mining Generalized Sequential Patterns," time, pp. 87-95. **11th International Symposium on Temporal Representation and Reasoning** (TIME'04), Tatihou - Normandie - France 2004.

[MEL04] Melanda, E. **Pós-processamento de regras de associação**. Tese (Doutorado) — Instituto Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2004.

[MER98] Merz, C., and Murphy. P. **UCI repository of machine learning databases**. University of California, Irvine, Department of Information and Computer Sciences, 1998. Disponível em: <http://www.ics.uci.edu/mlearn/MLRepository.html>.

[MIT03] Mitra, S. CS 313E: **Elements of Software Design** (Fall 2003) Lecture Notes Department of Computer Sciences University of Texas at Austin 2003. Disponível em: <http://www.cs.utexas.edu/users/mitra/csFall2003/cs313/lectures/lect2.html>.

[NAT05] Natarajan, R., Shekar, B. Interestingness of association rules in data mining: Issues relevant to e-commerce in **S¯adhan¯a** Vol. 30, Parts 2 & 3, April/June 2005, pp. 291–309. Kolkata, India, 2005.

[ORS06] Orsborn, K. **An introductory class in data mining** - Uppsala Database Laboratory - Department of Information Technology, Uppsala University, Uppsala, Sweden, Jul 2006. Disponível em: <http://www.it.uu.se/edu/course/homepage/infoutv/ht06>.

[PAD07] Padmanabhan, R. **Report # 2 (June 4th - June 8th, 2007) TANGO, Doc Lab, Rensselaer Polytechnic Institute**, Troy, NY 2007. Disponível em: <http://tango.byu.edu/papers/Raghav_Report_2.pdf>.

[PAR95] Park, J., Chen, M., Yu, P.: An effective hash-based algorithm for mining association rules. *In*: **Proc. of 1995 ACM-SIGMOD Int. Conf. on Management of Data**. (1995)

[PEI06] Pei, J., Zaiane, O. A Synthetic Data Generator for Clustering and Outlier Analysis. **Technical Report TR06-15, Jun 2006 – Department of Computing Science – Uviversity of Alberta.**

[PIA91] Piatetsky-Shapiro, G. Analysis and presentation of strong rules. **Knowledge Discovery in Databases**. AAAI/MIT Press, 1991.

[REC06] Rector, A. Coordenating taxonomies: Key to re-usable concept representations **Book Series Lecture Notes in Computer Science** Publisher Springer Berlin / Heidelberg ISSN 0302-9743 (Print) 1611-3349 (Online) Volume 934/1995 - Book Artificial Intelligence in Medicine DOI 10.1007/3-540-60025-6 Copyright 1995 ISBN 978-3-540-60025-1 DOI 10.1007/3-540-60025-6_122 Pages 15-28 Subject Collection Computer Science SpringerLink Date January 20, 2006

[SAV04] Savovaa, G. et al. Combining Rule-Based Methods and Latent Semantic Analysis for Ontology Structure Construction. *In:* **MEDINFO 2004** M. Fieschi at al. (Eds) Amsterdam: IOS Press 2004 IMIA, Amsterdam, 2004.

[SAV95] Savasere, A., Omiecinski, E. and Navathe, S, An Efficient Algorithm for Mining Association Rules in Large Databases. **Proc. 21st Int'l Conf. Very Large Data Bases***,* pp. 432-444, Sept. 1995.

[SIL95] Silberschatz, A.; Tuzhilin, A. On subjective measures of interestingness in knowledge discovery. *In*: **1st ACM SIGKDD Conf. on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 1995. p. 275281.

[SIP99] Sipina-W. University of Lyon. **Laboratoire E.R.I.C**. Disponível em: <http://eric.univ-lyon2.fr/~ricco/sipina.html>. 1999.

[SMI06] Smith, B., Kusnierczyk, W., Schober, D., Ceusters, W. Towards a Reference Terminology for Ontology Research and Development in the Biomedical Domain. **KR-MED 2006 "Biomedical Ontology in Action**" November 8, 2006, Baltimore, Maryland, USA.

[SON05] Song, M., Song, i., Xiaohua, H., Allen, R. **Semantic Query Expansion Combining Association Rules with Ontologies and Information Retrieval Techniques** - A Min Tjoa and J. Trujillo (Eds.): DaWaK 2005, LNCS 3589, pp. 326 – 335, 2005. Springer-Verlag Berlin Heidelberg 2005

[SRI95] Srikant, R.; Agrawal, R. Mining generalized association rules. **Proc. of the 21st Int'l Conference on Very Large Databases**, p. 407–419, 1995.

[STO93] Storey, V. Understanding Semantic Relationships. **VLDB Journal**,2, 455-488 (1993), Fred J. Maryanski, Editor.

[SZA69] Szabo, M. E. (ed.) **The collected papers of Gerhard Gentzen, Studies in Logic and the Foundations of Mathematics**. North-Holland. 1969

[TAN05] Tan, P., Steinbach, M., Kumar, V. **Introduction to Data Mining** – Addison-Wesley - Published: 05/02/2005. Disponível em: <http://www-users.cs.umn.edu/~kumar/dmbook/index.php>.

[TAO07] Tao Jiang, T., Tan, A., Wang, K. Mining Generalized Associations of Semantic Relations from Textual Web Content. *In*: **IEEE Transactions on Knowledge and Data Engineering.** vol. 19, no. 2, pp. 164-179, Feb , 2007

[TOI95] Toivonen, H., Klemettinem, M., Ronkainem, P., Hätönen, K., Mannila, H. - Pruning and Grouping Discovered Association Rules. *In:* **Workshop Notes of the ECML-95 Workshop on Statistics, Machine Learning, and Knowledge Discovery in DataBases** Heraclion, Crete, Greece, April 25 - 27, 1995.

[WAN99] Wand, Y., Storey, V., Weber, R. An Ontological Analysis of the Relationship Construct in Conceptual Modeling. **ACM Transactions on Database Systems**, Vol. 24, No. 4, December 1999, Pages 494–528.

[WIT05] Witten, I. H.; Frank, E. **Data Mining: Practical machine learning tools and techniques**. [S.l.]: Morgan Kaufmann, 2005.

[XIA05] Xiaohua, H., Xuheng, X. Mining novel connections from online biomedical text databases using semantic query expansion and semantic-relationship pruning. Interscience Publishers Publication. **International Journal of Web and Grid Services**. v. 1, Number 2/2005 Geneve Switzerland.

[XU004] Xu, X., Cong, G., Ooi, B., Kian-Lee, O. Tan, K., Tung, A. Semantic Mining and Analysis of Gene Expression Data. *In:* **Proceedings of the 30th VLDB Conference**, Toronto, Canada, 2004

[YIB00] Yibin, S., supervisionado por Hamilton, H., Lui, M. **Apriori implementation**. University of Regina – Regina, Canadá – Abril de 2000. Disponivel em: <http://www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/apriori.java>.

[YOO06] Yoo, I. **Semantic Text Mining and its Application in Biomedical Domain**. Tese (Doutorado). College of Information Science and Technology, Drexel University, Philadelphia, 2006.

[ZAI07] Zaïane, O. - Principles of Knowledge Discovery in Data. **Lecture Notes Chapter 3: Sequential Pattern Analysis** - University of Alberta - Winter 2007.

[ZAK00] ZAKI, M. J. Generating non-redundant association rules. In: **Conference on Knowledge Discovery in Data**. **Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2000.

[ZAK97] Zaki, M. ; Parthasarathy, S.; Ogihara, M.; and Li, W. 1997. New Algorithms for Fast Discovery of Association Rules. *In*: **Proc. of the Third Int'l Conf. on Knowledge Discovery in Databases and Data Mining**. 283-286. Newport Beach, California, August 1997.

[ZAL08] Zalta, E. (ed.). **Stanford Encyclopedia of Philosophy**. Library of Congress Catalog Data: ISSN 1095-5054 - Copyright © 2008 by The Metaphysics Research Lab - Center for the Study of Language and Information, Stanford University, Stanford, 2008. Disponível em: <http://plato.stanford.edu/>.

10 APÊNDICES

Serão exibidos a seguir os algoritmos de 2 a 7, que detalham as funções chamadas pelos algoritmos 1 e 8.

10.1 APÊNDICE 01 - FUNÇÃO PODA-REGRAS-1

A função poda-regras-1 é apresentada no algoritmo 2, e recebe como parâmetro um conjunto de regras de associação *R*, e um conjunto de relações de dependência entre os atributos *RD*. Inicialmente, o conjunto R', que irá armazenar o conjunto filtrado de regras de associação, recebe o conjunto R. Em seguida, para cada regra r presente em R, o algoritmo verifica se ela possui no conseqüente apenas um item. Caso verdadeiro, se o atributo deste item i for um atributo dependente, o algoritmo verifica se os itens que o determinam estão presentes no antecedente da regra. Caso verdadeiro, a regra r é removida do conjunto R', e o algoritmo "pula" para a próxima regra presente em R'. Por fim, o algoritmo retorna o conjunto R'.

Algoritmo 2 Função poda-regras-1

Require: <R,RD>

1: R' := R
2: **for all** r tal que r $\in$ R' **do**
3: **if** RHS(s) contém apenas um item i **then**
4: **if** atributo de i é um atributo dependente **then**
5: **if** itens que determinam i estão presentes em LHS(r) **then**
6: R' := R' - r
7: "pula" para a próxima regra r
8: **end if**
9: **end if**
10: **end if**
11: **end for**
12: **return** R'

10.2 APÊNDICE 02 - FUNÇÃO PODA-REGRAS-2

A função poda-regras-2 é apresentada no algoritmo 3, e recebe como parâmetro um conjunto de regras de associação *R*, um conjunto de relações de dependência entre os atributos *RD*, e o lado l da regra a ser analisado.

Algoritmo 3 Função poda-regras-2

Require: <R,RD,l>

1: R' := R
2: **for all** r tal que r ∈ R' **do**
3: **for all** i tal que i ∈ lado(l,r) **do**
4: **if** atributo de i é um atributo dependente **then**
5: **if** itens que determinam i estão presentes em lado(l,r) **then**
6: R' := R' - r
7: "pula" para a próxima regra r
8: **end if**
9: **end if**
10: **end for**
11: **end for**
12: **return** R'

Inicialmente, o conjunto R', que irá armazenar o conjunto filtrado de regras de associação, recebe o conjunto R. Em seguida, para cada regra r presente em R, o algoritmo percorre os itens do lado l da regra. Se o atributo do item corrente i for um atributo dependente, o algoritmo verifica se os itens que determinam i estão presentes no lado l da regra. Caso verdadeiro, a regra r é removida do conjunto R', e o algoritmo "pula" para a próxima regra presente em R'. Por fim, o algoritmo retorna o conjunto R'.

10.3 APÊNDICE 03 - FUNÇÃO PODA-REGRAS-3

A função poda-regras é apresentada no Algoritmo 4, e recebe como parâmetro um conjunto de regras de associação *R*, e um conjunto de relações de dependência entre os atributos *RD*.

Algoritmo 4 Função poda-regras-3

Require: <R,RD>

```
1: R' := R
2: for all r tal que r ∈ R' do
3:   if RHS(r) tem mais de um item then
4:     for all i tal que i ∈ RHS(r) do
5:       if atributo de i é um atributo dependente then
6:         if itens que determinam i estão presentes em LHS(r) then
7:           R' := R' - r
8:           "pula" para a próxima regra r
9:         end if
10:      end if
11:    end for
12:  end if
13: end for
14: return R'
```

Inicialmente, o conjunto R' que irá armazenar o conjunto filtrado de regras de associação recebe o conjunto R. Em seguida, para cada regra r presente em R, que possui mais de um item no conseqüente, o algoritmo percorre os itens do conseqüente da regra. Se o atributo do item corrente i for um atributo dependente, o algoritmo verifica se os itens que especificam este item estão presentes no antecedente da regra. Caso verdadeiro, a regra r é removida do conjunto R', e o algoritmo "pula" para a próxima regra presente em R'. Por fim, o algoritmo retorna o conjunto R'.

10.4 APÊNDICE 04 - FUNÇÃO GERA-REGRAS-GERAIS

A função gera-regras-gerais é apresentada no algoritmo 5, e recebe como parâmetro um conjunto de regras de associação *R*, e um conjunto das relações de dependência entre os atributos *RD*. A função retorna o conjunto das regras mais gerais (da forma da expressão 5.2.1.4) presentes em R. Inicialmente, o conjunto R' que irá armazenar o conjunto de regras de associação gerais recebe o conjunto vazio (∅). Em seguida, para cada regra r presente em R, o algoritmo percorre os itens do antecedente da regra. Se o atributo do item corrente for um atributo dependente, a regra é adicionada ao conjunto R', e o algoritmo "pula" para a próxima regra presente em R.

Por fim, o algoritmo retorna o conjunto R'.

Algoritmo 5 Função gera-regras-gerais

Require: <R, RD>

1: R' := ∅;
2: **for all** r tal que r ∈ R **do**
3: **for all** *item* tal que *item* ∈ LHS(r) **do**
4: **if** atributo de item é um atributo dependente **then**
5: R' := R' ∪ r
6: "pula" para a próxima regra
7: **end if**
8: **end for**
9: **end for**
10: **return** R'

10.5 APÊNDICE 05 - FUNÇÃO ENRIQUECIMENTO

O algoritmo 6 mostra o processo de *enriquecimento*. Recebe como parâmetros de entrada: um conjunto *R* de regras de associação, um conjunto *RD* das relações de dependência entre os atributos, um mapeamento *m_mapAtributosComDep,* que relaciona os atributos que possuam dependências e um mapeamento *m_atributos,* que armazena os atributos que são dependentes.

Algoritmo 6

Require: <R, RD, m_dependencias, m_atributos>

```
1: R' := R
2: for all regra r ∈ R do
3:    Criar_lista_atributos_dependentes_ inicialmente_vazia (lv ∈ RD)
4:    for all itens ((i ∈ LHS(r)) ∧ (i ∈ m_mapAtributosComDep)) do
5:        for all itens (idep ∈ m_atributos) do
6:           if (idep ∉ lv)
7:                              procurarAtributosEspecificamY(idep, atributosEspecificamY,
nivelDependencia);
8:                 if D ∈ r
9:                    val := calcularValorAtributo(idep, LHS(r), nivelDependencia)
10:                   nr := Criar_copia_regra(r)
11:                   nr := Subst_itens(atributosEspecificamY, idep, val)
12:                   R' := R' + r
13:                   lv := lv + idep
14:                end if
15:           end if
16:       end for
17:   end for
18: end for
19: return R'
```

Inicialmente, o conjunto R' que irá armazenar o conjunto enriquecido de regras de associação recebe o conjunto R. Em seguida, para cada regra *r* presente em R, o algoritmo cria uma lista, inicialmente vazia, de atributos com dependentes presentes na regra. Depois percorre os itens do lado *l* da regra. Se o atributo do item corrente *i* for um atributo com dependente, o algoritmo recupera todos os itens *idep* dependentes de *i*. A função *procurarAtributosEspecificamY* retorna os atributos *atributosEspecificamY*, com os quais se calcula o valor de *idep*. Se algum atributo de *atributosEspecificamY*, não estiver presente em *r* nada se faz. Caso contrário, com este conjunto *atributosEspecificamY*, calcula-se o valor *val* do atributo dependente por meio da função *calcularValorAtributo*. Uma cópia *nr* da regra r é criada. A função faz a substituição dos atributos *atributosEspecificamY* em *nr* pelo atributo dependente *val* . O conjunto enriquecido de regras recebe *nr* e o item com dependência *idep* é acrescido à lista *lv*.

10.6 APÊNDICE 06 - FUNÇÃO GERA-REGRAS-ESPECÍFICAS

A função gera-regras-específicas é apresentada no algoritmo 7, e recebe como parâmetro um conjunto de regras de associação *R*, um conjunto de relações de dependência entre os atributos *RD*, uma regra mais geral *rg*, e o grau de dependência entre os atributos a ser considerado *n*. A função retorna o conjunto das regras mais específicas (da forma da expressão 5.2.1.5) presentes em R, que são redundantes com rg.

Algoritmo 7 Função gera-regras-específicas

Require: <R,RD,rg,n>

```
1:  R' := ∅;
2:  y := atributo dependente de rg
3:  X := atributos que determinam y considerando o grau de dependência n
4:  for all r tal que r ∈ R do
5:    if LHS(r) possui os atributos de X then
6:      v := calcula-valor(y, r);
7:      if v ≠ valor do atributo y em rg then
8:        "pula" para a próxima regra
9:      end if
10:     for i = 0 to número-itens(LHS(r)) do
11:       if (atributo de r.itens[i] ∉ X and r.itens[i] ∉ rg) or (atributo de r.itens[i] = y) then
12:             "pula" para a próxima regra
13:       end if
14:     end for
15:     R' := R' ∪ r
16:   end if
17: end for
18: return R'
```

Inicialmente, o conjunto R' que irá armazenar o conjunto de regras mais específicas recebe o conjunto vazio (∅). Em seguida, a variável y recebe o atributo dependente da regra geral e a variável X recebe o conjunto dos atributos que especificam y. Neste passo, apenas as relações de dependência de grau n são consideradas. O próximo passo do algoritmo é percorrer todas as regras presentes em R. Para cada regra r em R, o algoritmo verifica se a regra possui os atributos que especificam y no antecedente. Caso verdadeiro, a variável v recebe o resultado da função calcula-valor. Esta função calcula o valor do atributo y a partir dos atributos de X presentes em r. Se o valor calculado for diferente do valor do atributo y de rg, então a regra r não é redundante com rg, e o algoritmo "pula" para a próxima regra em R. Caso contrário, o algoritmo continua a verificação da regra r. O próximo passo consiste em verificar se os demais elementos da regra r são os mesmos da regra rg. Para tanto, o algoritmo percorre os itens do antecedente da regra r e faz a seguinte verificação: se o atributo do item não é um dos que especificam y, e este item não existe na regra rg, ou se o atributo do item é o atributo y. Caso verdadeiro, a regra r não é redundante com rg, e o algoritmo "pula" para a próxima regra r. Se o algoritmo chegar à linha 15, a regra r é redundante com rg e é adicionada ao conjunto R'. Por fim, o algoritmo retorna o conjunto R'.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo