Generalização de regras de associação

Marcos Aurélio Domingues

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 19/02/2004

Assinatura: _____

Generalização de regras de associação¹

Marcos Aurélio Domingues

Orientadora: *Prof^{ta} Dr^a Solange Oliveira Rezende*

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

USP - São Carlos
Fevereiro/2004

¹Trabalho realizado com auxílio financeiro da FAPESP e CAPES.

Este documento foi preparado utilizando-se o formatador de textos $\text{\LaTeX}$. Sua bibliografia é gerada automaticamente pelo $\text{\BibTeX}$, utilizando o estilo Chicago.

©Copyright 2004 - Marcos Aurélio Domingues
Todos os direitos Reservados

Dedicatória

*À minha família.
Com carinho especial, a meus
pais Isabete e José Pedro e a meus
irmãos Marcelo e Márcio.*

Agradecimentos

Aos meus pais, José Pedro e Isabete, e meus irmão Marcelo e Márcio pelo amor, carinho e respeito tão importantes para que pudesse realizar tudo que sempre desejei.

À professora Solange que foi muito mais que uma orientadora. Desde que a conheci, no início de 2002, descobri que entre um professor e seu orientando pode existir uma verdadeira amizade, muito mais do que uma simples relação “professor-aluno”. Com o seu bom humor e carisma suportou as minhas brincadeiras, “reclamações” e “momentos de grande concentração nos estudos”. Muito obrigado Solange pelo apoio, incentivo e paciência. Você contribuiu muito para o meu crescimento profissional e pessoal. Você é um exemplo de pessoa que devemos nos espelhar.

Aos amigos, Daniel Dosualdo, Veronica, Marina, Rodrigo Bianchi, Débora, Rodrigo Felício, Edson Takashi, Marcelo Muniz, Thiago, Walter, Camila, Jean, Tanaka, Camilo, Valmir, Gedson, Gustavo, Cláudia, Claudinha, Katti, entre vários outros amigos que contribuíram na minha formação e também pelos bons momentos que passamos juntos.

Às amigas Jaqueline e Patricia pelos momentos de alegria e descontração.

Ao amigo Ronaldo por ajudar nas minhas dúvidas sobre L^AT_EX.

Ao amigo Marcos Paula, em especial, por tanto ter colaborado no desenvolvimento deste trabalho. Sem a sua ajuda, este trabalho teria sido muito mais difícil e levaria um tempo muito maior para ser concluído.

À amiga Roberta, em especial, por ter me auxiliado nas atualizações do Ambiente de Exploração de Regras *RulEE*, um dos resultados do projeto de mestrado do amigo Marcos Paula, ao qual este projeto foi integrado, e também por ter me ajudado na leitura e correção desta dissertação.

Ao professor e amigo Edson Melanda pelas dicas e sugestões acrescidas a este trabalho.

Ao professor Joaquim Quinteiro Uchôa, por tanto tempo meu orientador de Iniciação Científica, e, por conseguinte, um dos responsáveis pela minha incursão na área científica.

Aos professores Maria Carolina e André. Esses professores juntamente com a professora Solange tornam o LABIC um ótimo ambiente, propício ao trabalho.

A todos do LABIC pelos bons momentos e por fazer do laboratório um ambiente agradável, propício para a realização de qualquer trabalho.

A todos os funcionários do ICMC pela atenção, respeito, dedicação e educação.

À Universidade de São Paulo pela oportunidade.

À FAPESP e CAPES pelo apoio financeiro.

À Deus por ter me dado condições para cumprir mais uma etapa da minha vida.

Mineração de Dados é um processo de natureza iterativa e interativa responsável por identificar padrões em grandes conjuntos de dados, objetivando extrair conhecimento válido, útil e inovador a partir desses. Em Mineração de Dados, Regras de Associação é uma técnica que consiste na identificação de padrões intrínsecos ao conjunto de dados. Essa técnica tem despertado grande interesse nos pesquisadores de Mineração de Dados e nas organizações, entretanto, a mesma possui o inconveniente de gerar grande volume de conhecimento no formato de regras, dificultando a análise e interpretação dos resultados pelo usuário.

Nesse contexto, este trabalho tem como objetivo principal generalizar e eliminar Regras de Associação não interessantes e/ou redundantes, facilitando, dessa maneira, a análise das regras obtidas com relação à compreensibilidade e tamanho do conjunto de regras. A generalização das Regras de Associação é realizada com o uso de taxonomias.

Entre os principais resultados deste trabalho destacam-se a proposta e a implementação do algoritmo *GART* e do módulo computacional *RuleE-GAR*. O algoritmo *GART* (*Generalization of Association Rules using Taxonomies* – Generalização de Regras de Associação usando Taxonomias) utiliza taxonomias para generalizar Regras de Associação. Já o módulo *RuleE-GAR*, além de facilitar o uso do algoritmo *GART* durante a identificação de taxonomias e generalização de regras, provê funcionalidades para analisar as Regras de Associação generalizadas.

Os experimentos realizados, neste trabalho, mostraram que o uso de taxonomias na generalização de Regras de Associação pode reduzir o volume de um conjunto de regras.

Abstract

Data Mining refers to the process of finding patterns in large data sets. The Association Rules in Data Mining try to identify intrinsic behaviors of the data set. This has motivated researchers of Data Mining and organizations. However, the Association Rules have the inconvenient of generating a great amount of knowledge in the form of rules. This makes the analysis and interpretation of the results difficult for the user.

Taking this into account, the main objective of this research is the generalization and elimination of non-interesting and/or redundant Association Rules. This facilitates the analysis of the rules with respect to the comprehensibility and the size of the rule set. The generalization is realized using taxonomies.

The main results of this research are the proposal and the implementation of the algorithm *GART* and of the computational module *RuleE-GAR*. The algorithm *GART* (Generalization of Association Rules using Taxonomies) uses taxonomies to generalize Association Rules. The module *RuleE-GAR* facilitates the use of the algorithm *GART* in the identification of taxonomies and generalization of rules and provides functionalities to the analysis of the generalized Association Rules.

The results of experiments showed that the employment of taxonomies in the generalization of Association Rules can reduce the size of a rule set.

Sumário

1	Introdução	1
1.1	Motivação e Objetivo	2
1.2	Principais Contribuições desta Dissertação	3
1.3	Organização desta Dissertação	4
2	Mineração de Dados e Regras de Associação	7
2.1	Mineração de Dados	8
2.1.1	Identificação do Problema	10
2.1.2	Pré-Processamento	11
2.1.3	Extração de Padrões	14
2.1.4	Pós-Processamento	16
2.1.5	Utilização do Conhecimento	17
2.2	Regras de Associação	17
2.2.1	Conceitos e Definições	18
2.2.2	O Algoritmo <i>Apriori</i>	24
2.2.3	Algoritmo Simples para Gerar Regras de Associação	30
2.3	Considerações Finais	32
3	Uso de Taxonomias em Regras de Associação	33
3.1	Taxonomias e Regras de Associação	34

3.1.1	Notações e Definições	36
3.1.2	Cálculo do Suporte Utilizando Taxonomias	38
3.2	Algoritmos para Gerar <i>Itemsets</i> Frequentes com Taxonomias	40
3.2.1	Algoritmo Básico para Gerar <i>Itemsets</i> Frequentes com Taxonomias	40
3.2.2	Algoritmo <i>Cumulate</i>	41
3.2.3	Algoritmo <i>Stratify</i>	42
3.3	Uso de Taxonomias na Remoção de Regras Não Interessantes	43
3.3.1	Medida de Interesse	43
3.3.2	Algoritmo para Remover Regras Não Interessantes	47
3.4	Considerações Finais	48
4	Algoritmo Proposto para Generalização de Regras de Associação	49
4.1	Proposta de um Processo para Generalizar Regras de Associação	50
4.2	O Algoritmo <i>GART</i>	53
4.2.1	Função generaliza-regras	55
4.2.2	Função generaliza-uma-regra	57
4.2.3	Função desfaz-última-generalização	58
4.2.4	Função calcula-TC	58
4.3	Implementação do Algoritmo <i>GART</i>	66
4.3.1	Biblioteca de Classes do Algoritmo <i>GART</i>	66
4.3.2	Arquivos de Entrada e Saída	68
4.3.3	Interface com o Usuário	72
4.4	Considerações Finais	74
5	Módulo Computacional para Regras de Associação <i>RuleE-GAR</i>	75
5.1	O Projeto <i>Discover</i>	76
5.2	O Ambiente para Exploração de Regras <i>RuleE</i>	80
5.3	O Módulo Computacional <i>RuleE-GAR</i>	84
5.3.1	Descrição da Interface do Módulo <i>RuleE-GAR</i>	87

5.3.2	Generalização de Regras de Associação Utilizando o Módulo <i>RuleE-GAR</i>	90
5.3.3	Análise de Regras de Associação Generalizadas Utilizando o Módulo <i>RuleE-GAR</i>	94
5.4	Considerações Finais	99
6	Experimentos Realizados	101
6.1	Descrição da Base de Dados Utilizada	102
6.2	Pré-Processamento da Base de Dados	105
6.3	Mineração das Regras de Associação	107
6.4	Pós-Processamento das Regras de Associação	108
6.4.1	Pós-Processamento Utilizando Taxonomias	108
6.4.2	Pós-Processamento Utilizando Taxonomias sem Classificação	120
6.4.3	Combinação de Taxonomias, com e sem Classificação, no Pós-Processamento de Regras de Associação	124
6.5	Síntese Geral dos Experimentos	130
6.6	Considerações Finais	131
7	Conclusão	133
7.1	Principais Contribuições deste Trabalho	134
7.2	Resultados dos Experimentos Realizados	135
7.3	Trabalhos Futuros	136
	Referências	145

Lista de Figuras

2.1	Etapas do processo de Mineração de Dados	10
2.2	Contextualização de Regras de Associação em Mineração de Dados	18
2.3	Espaço de busca para o conjunto de itens $A = \{a, b, c, d, e\}$	20
2.4	Separação de <i>itemsets</i> freqüentes e não-freqüentes no espaço de busca de $A = \{a, b, c, d, e\}$	21
2.5	<i>Itemsets</i> candidatos armazenados em uma árvore- <i>hash</i>	27
2.6	Espaço de busca do conjunto de itens $A = \{\text{bermuda, calça, camiseta, sandália, tênis}\}$	28
3.1	Exemplo de uma taxonomia para vestuário	34
3.2	Exemplo de uma taxonomia sem classificação para vestuário	35
3.3	Uma taxonomia τ para vestuário	45
4.1	Generalização de Regras de Associação usando uma taxonomia	50
4.2	Generalização de Regras de Associação usando duas taxonomias	51
4.3	Processo proposto para generalização de Regras de Associação	52
4.4	Exemplo 8 – Conjunto de taxonomias $\mathcal{T}$	61
4.5	Exemplo de um arquivo de dados transacionais <code>.apr.data</code>	68
4.6	Exemplo de um arquivo de regras <code>.apr.dcar</code>	69
4.7	Exemplo do arquivo de taxonomias <code>.tax</code> com as taxonomias da Figura 4.8 .	69

4.8	Um conjunto de taxonomias $\mathcal{T}$ com 4 taxonomias	70
4.9	Um conjunto $\mathcal{T}$ contendo uma taxonomia de 3 níveis	70
4.10	Exemplo do arquivo <code>.tax</code> contendo a taxonomia de 3 níveis da Figura 4.9	70
4.11	Exemplo de um arquivo de relatório <code>.gar.report</code>	71
4.12	Exemplo de um arquivo de regras generalizadas <code>.gar.dcar</code>	71
4.13	Exemplo do relatório <code>.gar.dcar.report</code>	72
5.1	Funcionalidades do projeto <i>Discover</i>	76
5.2	Arquitetura geral do ambiente <i>RuleE</i>	80
5.3	Interface do ambiente <i>RuleE</i>	82
5.4	Contextualização e funcionalidades do módulo computacional <i>RuleE-GAR</i> no ambiente <i>RuleE</i>	84
5.5	Diagrama físico da Base de Dados do ambiente <i>RuleE</i>	86
5.6	<i>Login</i> no ambiente <i>RuleE</i>	88
5.7	<i>Menu</i> principal do ambiente <i>RuleE</i>	88
5.8	Interface para generalização de Regras de Associação	88
5.9	Interface para análise das Regras de Associação generalizadas	89
5.10	Sistema de ajuda (<i>Help</i>) do módulo <i>RuleE-GAR</i>	90
5.11	Seleção de um conjunto de Regras de Associação para ser generalizado	92
5.12	Definição das taxonomias para a generalização das regras	92
5.13	Término do processo de generalização das Regras de Associação	93
5.14	Resultado de uma consulta a uma base de Regras de Associação generalizadas	95
5.15	Visualização do arquivo de dados transacionais (<i>link Data Set</i>)	96
5.16	Visualização do arquivo de regras originais (<i>link Rule Set</i>)	96
5.17	Visualização do arquivo de regras generalizadas (<i>link Generalized Rule Set</i>)	96
5.18	Visualização do arquivo de taxonomias (<i>link Taxonomy Set</i>)	96
5.19	Tela de visualização de uma Regra de Associação generalizada expandida	97
5.20	Tela de visualização das Regras de Associação que geraram uma regra ge- neralizada	97

5.21	Tela de visualização dos valores das medidas suporte e confiança de uma regra generalizada	98
5.22	Tela de visualização de itens originais a partir de um item generalizado . .	98
6.1	Exemplo das linhas “03”, “02” e “01” em um arquivo de vendas	102
6.2	Diagrama físico das tabelas criadas no SGBD	106
6.3	Conjunto de taxonomias Tax01	109
6.4	Conjunto de taxonomias Tax02	110
6.5	Conjunto de taxonomias Tax03	110
6.6	Conjunto de taxonomias Tax04	111
6.7	Conjunto de taxonomias Tax05	111
6.8	Conjunto de taxonomias Tax06	112
6.9	Conjunto de taxonomias Tax07	112
6.10	Conjunto de taxonomias Tax08	113
6.11	Conjunto de taxonomias Tax09	113
6.12	Conjunto de taxonomias Tax10	114
6.13	Conjunto de taxonomias Tax11	114
6.14	Conjunto de taxonomias Tax12	115
6.15	Conjunto de taxonomias Tax13	115
6.16	Gráficos dos resultados obtidos com o uso de conjuntos de taxonomias na generalização dos conjuntos de Regras de Associação	119
6.17	Tsc01 – Conjunto contendo 6 taxonomias sem classificação	121
6.18	Tsc02 – Conjunto contendo 16 taxonomias sem classificação	121
6.19	Tsc03 – Conjunto contendo 26 taxonomias sem classificação	121
6.20	Gráficos dos resultados obtidos com o uso de conjuntos de taxonomias sem classificação na generalização dos conjuntos de Regras de Associação	123
6.21	Gráficos dos resultados obtidos com o uso de combinações de conjuntos de taxonomias, com e sem classificação, na generalização dos conjuntos de Regras de Associação	127

6.22 Gráficos comparando as taxas de redução obtidas com os conjuntos de taxonomias, com e sem classificação, e com a combinação de ambos os conjuntos	128
--	-----

Lista de Tabelas

2.1	Conjunto de exemplos no formato atributo-valor	12
2.2	Tabela booleana de itens-transações	19
2.3	Tabela de itens-transações	19
2.4	Relação de itens comprados por transação do Exemplo 1	23
2.5	<i>Itemsets</i> freqüentes do Exemplo 1	24
2.6	Conjunto <i>Resposta</i> contendo os <i>itemsets</i> freqüentes	30
3.1	Relação de itens comprados por transação do Exemplo 6	38
3.2	Relação de itens comprados por transação do Exemplo 7	46
4.1	Tabela de Contingência	59
4.2	Exemplo 8 – Relação de itens comprados por transação	60
4.3	Exemplo 8 – Conjunto de Regras de Associação R	61
4.4	Exemplo 8 – Subconjuntos de Regras de Associação com conseqüentes semelhantes	62
4.5	Exemplo 8 – Conjunto de Regras de Associação generalizadas R_g	65
4.6	Descrição dos métodos da classe ARigart	66
4.7	Descrição dos métodos da classe ARtaxonomy	67
4.8	Descrição dos métodos da classe ARrule	67

5.1	Descrição dos métodos implementados para a manipulação de Regras de Associação generalizadas	87
6.1	Descrição dos campos de dados da linha “01”	103
6.2	Descrição dos campos de dados da linha “02”	103
6.3	Descrição dos campos de dados da linha “03”	104
6.4	Descrição dos conjuntos de dados utilizados nos experimentos	107
6.5	Descrição do número de taxonomias armazenadas em cada conjunto	117
6.6	Resultados obtidos com o uso de conjuntos de taxonomias na generalização dos conjuntos de Regras de Associação	118
6.7	Resultados obtidos com o uso de conjuntos de taxonomias sem classificação na generalização dos conjuntos de Regras de Associação	122
6.8	Combinação dos conjuntos de taxonomias com e sem classificação	125
6.9	Resultados obtidos com o uso de combinações de conjuntos de taxonomias, com e sem classificação, na generalização dos conjuntos de regras	126

Lista de Algoritmos

1	<i>Apriori</i>	25
2	Função apriori-gen	26
3	Gera Regras de Associação	30
4	Remove Regras Não Interessantes	47
5	<i>GART</i>	54
6	Função generaliza-regras	55
7	Função generaliza-uma-regra	57
8	Função desfaz-última-generalização	58
9	Função calcula-TC	59

Introdução

A evolução da computação impulsionada pelo aumento do poder de processamento dos computadores, pelo armazenamento contínuo de grandes quantidades de dados a baixo custo, pela introdução de novas tecnologias de transmissão e disseminação de dados tem dado às organizações a capacidade de armazenar informações detalhadas sobre cada transação que efetuam, gerando grandes Bases de Dados. As organizações reconhecem o valor das informações contidas em suas Bases de Dados e têm investido na aquisição e desenvolvimento de ferramentas de análise que produzam informações úteis.

Durante anos, métodos predominantemente manuais têm sido utilizados para transformar dados em conhecimento. Porém, o uso desses métodos tem se tornado dispendioso (em termos financeiros e de tempo), subjetivo e inviável, quando aplicados a grandes Bases de Dados (Fayyad, Piatetsky-Shapiro, & Smyth 1996a).

Devido aos problemas com os métodos manuais, tornou-se necessário o desenvolvimento de processos de análise automática, como o Processo de Extração de Conhecimento de Bases de Dados ou Mineração de Dados. Esse processo, de natureza iterativa e interativa, tem despontado por seu desempenho em diversos domínios, na extração de padrões válidos, novos, e potencialmente úteis dos dados (Fayyad, Piatetsky-Shapiro, & Smyth 1996a).

O processo de Mineração de Dados pode ser dividido em três grandes etapas (Rezende, Pugliesi, Melanda, & Paula 2003): Pré-processamento dos dados, Extração de Padrões e Pós-processamento do conhecimento. O Pré-processamento dos dados inclui a realização de procedimentos de transformação e limpeza dos dados, de redução do volume de dados ou atributos, entre outros. A etapa de Extração de Padrões compreende a identificação da função de Mineração de Dados a ser empregada, a escolha do algoritmo de mineração e a realização da extração de conhecimento (padrões). Por fim, a etapa de Pós-processamento possibilita a avaliação, interpretação, explanação, filtragem e integração do conhecimento extraído.

Após passar pela etapa de Pós-processamento, o conhecimento extraído pode ser incorporado a um Sistema Inteligente ou disponibilizado ao usuário, para que possa ser utilizado como apoio em processos de tomada de decisão.

1.1 Motivação e Objetivo

Embora tenha se tornado necessária a utilização do processo de Mineração de Dados para extrair conhecimento a partir de dados, sua aplicação pode gerar uma elevada quantidade de padrões, muitos dos quais podem não ser importantes, relevantes ou interessantes para o usuário. Fornecer ao usuário uma grande quantidade de padrões não é produtivo pois, geralmente, ele procura poucos padrões que sejam interessantes. Portanto, é de vital importância o desenvolvimento de técnicas de apoio no sentido de fornecer aos usuários apenas os padrões mais interessantes (Silberschatz & Tuzhilin 1995). Buscando resolver esse problema, pesquisas em Pós-processamento de conhecimento, principalmente relacionadas com a avaliação da qualidade, interessabilidade, compreensibilidade e visualização do conhecimento extraído, vêm sendo realizadas nos últimos anos (Bruha & Famili 2000).

O problema de se gerar grandes quantidades de padrões recebe uma maior ênfase em Regras de Associação, uma das técnicas de Mineração de Dados que recentemente tem despertado grande interesse (Baesens, Viaene, & Vanthienen 2000). Na área acadêmica pesquisas vêm sendo desenvolvidas com essa técnica e as organizações têm utilizado os resultados no comércio, em contratos de seguro, na saúde, no geoprocessamento, na biologia molecular entre outras áreas (Liu, Hsu, Chen, & Ma 2000; Clementini, Felice, & Koperski 2000; Semenova, Hegland, Graco, & Williams 2001). Um exemplo do emprego de Regras de Associação na área comercial pode ser observado no site da companhia Amazon¹, com sugestões do tipo “... quem comprou o livro ‘*Nome do Livro*’, que foi procurado, também

¹Amazon – <http://www.amazon.com>.

comprou esses livros...” (Melanda 2002).

Uma Regra de Associação caracteriza o quanto a presença de um conjunto de itens nos registros de uma Base de Dados implica na presença de algum outro conjunto distinto de itens nos mesmos registros (Agrawal & Srikant 1994).

Embora haja um grande entusiasmo com a técnica de Regras de Associação, a mesma possui o inconveniente de gerar grande volume de conhecimento no formato de regras.

Uma abordagem para solucionar o problema da grande quantidade de padrões extraídos pela técnica de Regras de Associação é a aplicação de taxonomias (Srikant & Agrawal 1997a; Srikant & Agrawal 1997b; Liu, Hsu, Chen, & Ma 2000; Adamo 2001). As taxonomias refletem uma visão coletiva ou individual de como os itens podem ser hierarquicamente classificados, podendo ser utilizadas para eliminar regras não interessantes e/ou redundantes (Adamo 2001).

Diante desse contexto, este trabalho tem como objetivo principal generalizar e eliminar Regras de Associação não interessantes e/ou redundantes, facilitando, dessa maneira, a análise das regras obtidas. A generalização das regras é realizada com o uso de taxonomias. A seguir são apresentadas as principais contribuições desta dissertação.

1.2 Principais Contribuições desta Dissertação

As principais contribuições deste trabalho são a proposta e a implementação do algoritmo *GART*, para a generalização de Regras de Associação, e do módulo computacional *RuleE-GAR* para a análise das regras generalizadas.

O algoritmo proposto, que recebeu o nome de *GART* (*Generalization of Association Rules using Taxonomies* – Generalização de Regras de Associação usando Taxonomias), utiliza taxonomias para generalizar Regras de Associação na etapa de Pós-processamento do conhecimento extraído. Desse modo, é possível reduzir o volume de Regras de Associação geradas, facilitando, assim, a análise e o uso dessas regras.

Já o módulo computacional *RuleE-GAR* disponibiliza funcionalidades para a análise das Regras de Associação generalizadas com o uso do algoritmo *GART*. O módulo foi implementado e integrado ao *RuleE*, que é um Ambiente para Exploração de Regras de Classificação, Regressão e Associação.

Cabe ressaltar que este trabalho faz parte do *Discover* (Baranauskas & Batista 2000; Rezende, Monard, Batista, Prati, Pugliesi, & Melanda 2004), um projeto em desen-

volvimento pelos pesquisadores do LABIC² (Laboratório de Inteligência Computacional - ICMC/USP) e que tem como objetivo fornecer um ambiente integrado para apoiar as etapas do processo de Extração de Conhecimento de Dados e Textos.

1.3 Organização desta Dissertação

Neste capítulo foi apresentado o contexto em que se insere este trabalho, bem como o objetivo e as principais contribuições do mesmo. O restante da dissertação está organizada da maneira descrita a seguir.

No Capítulo 2 é descrito, de modo geral, o processo de Mineração de Dados, abordando as etapas de Pré-processamento dos dados, Extração de Padrões e Pós-processamento do conhecimento extraído. Como a técnica de Regras de Associação é objeto de estudo deste trabalho, também é apresentada nesse capítulo uma definição da técnica, bem como exemplificações de uso, o algoritmo *Apriori* (que sob aspecto histórico é considerado um dos mais importantes) e um algoritmo simples para gerar Regras de Associação proposto em Agrawal & Srikant (1994).

No Capítulo 3 são apresentados aspectos gerais da aplicação de taxonomias em Regras de Associação, sendo abordados a definição da técnica, alguns algoritmos utilizados na obtenção de Regras de Associação com taxonomias e uma medida para remover regras não interessantes utilizando as informações das taxonomias aplicadas à regra. Além disso, são apresentados alguns problemas que foram identificados nos algoritmos que utilizam taxonomias para minerar Regras de Associação, descritos nesse capítulo.

No Capítulo 4 é proposto um algoritmo que utiliza taxonomias para generalizar Regras de Associação na etapa de Pós-processamento. Esse algoritmo, denominado *GART*, é detalhadamente apresentado nesse capítulo.

No Capítulo 5 são descritas as principais funcionalidades do módulo computacional *RuleE-GAR*. Esse módulo foi proposto e desenvolvido com o intuito de facilitar o uso do algoritmo *GART* (descrito no Capítulo 4) na generalização de Regras de Associação e de fornecer funcionalidades para a análise das regras generalizadas. Além disso, nesse capítulo são apresentados o Ambiente para Exploração de Regras *RuleE* (Paula 2003), ao qual o módulo *RuleE-GAR* foi integrado, e o Projeto *Discover* (Baranauskas & Batista 2000; Rezende, Monard, Batista, Prati, Pugliesi, & Melanda 2004), ao qual este trabalho também faz parte.

No Capítulo 6 são apresentados alguns experimentos utilizando o algoritmo *GART*.

²LABIC – <http://labic.icmc.usp.br>.

Os experimentos foram realizados para mostrar que o uso de taxonomias na generalização de um conjunto de Regras de Associação pode reduzir o volume desse conjunto.

No Capítulo 7 são apresentadas as conclusões deste trabalho e propostas para trabalhos futuros.

Mineração de Dados e Regras de Associação

A Mineração de Dados é um processo de natureza **iterativa** (podendo ser repetido várias vezes para ajustar parâmetros ou para melhorar o processo de escolha dos dados, objetivando melhores resultados numa próxima iteração) e **iterativa** (baseado na interação dos especialistas de domínio, responsáveis pela execução do processo e usuários). Esse processo se apresenta como uma tecnologia automatizada para análise de grandes volumes de dados e obtenção de conhecimento, podendo ser classificado como Mineração de Dados *Preditiva* ou *Descritiva*, de acordo com os objetivos que se quer atingir com a utilização do processo (Rezende, Pugliesi, Melanda, & Paula 2003).

A Mineração de Dados Preditiva consiste na generalização de exemplos com suas respectivas classes (atributo meta) conhecidas em uma linguagem ou modelo capaz de prever o atributo meta de um novo exemplo. Os dois principais tipos de problemas de predição são Classificação (Mitchell 1998; Monard & Baranauskas 2003a; Monard & Baranauskas 2003b) e Regressão (Weiss & Indurkha 1998).

Já Mineração de Dados Descritiva consiste na identificação de padrões intrínsecos ao conjunto de dados, sendo que esses dados não possuem um atributo meta especificado,

como acontece na predição. Alguns dos problemas de descrição são Regras de Associação (Agrawal & Srikant 1994; Adamo 2001), *Clustering* e Sumarização (Mitchell 1998).

Dentre os problemas descritivos, a técnica de Regras de Associação vem obtendo grande destaque nos últimos anos, seja na área acadêmica (são mais de 3300 artigos no *Citeseer*¹ (Srikant 2001)) ou na área de aplicações práticas (indicado pela disponibilidade de ferramentas, que no início de 1998 não existiam (Piatetsky-Shapiro 1999) e no final de 2003 totalizavam 18 ferramentas²).

O objetivo deste capítulo é apresentar uma visão geral do processo de Mineração de Dados e da técnica de Regras de Associação. O processo de Mineração de Dados é descrito na Seção 2.1, sendo abordadas principalmente as etapas do processo referentes ao Pré-processamento dos dados, Extração de Padrões e Pós-processamento do conhecimento extraído. Na Seção 2.2 é apresentada a técnica de Mineração de Dados denominada Regra de Associação, sendo abordadas a definição da técnica, exemplificações de uso, o algoritmo *Apriori* e um algoritmo simples para gerar Regras de Associação proposto por Agrawal & Srikant (1994). Por fim, na Seção 2.3 são feitas algumas considerações finais do capítulo.

2.1 Mineração de Dados

O processo de identificação de conhecimento em Bases de Dados é conhecido como Extração de Conhecimento de Bases de Dados ou Mineração de Dados, sendo geralmente referenciado na literatura como *Knowledge Discovery in Databases (KDD)*. Alguns autores consideram a Extração de Conhecimento de Bases de Dados e a Mineração de Dados processos distintos (Fayyad, Piatetsky-Shapiro, & Smyth 1996a). Entretanto, neste trabalho, os processos serão tratados indistintamente, sendo referenciados como processos de extrair conhecimento a partir de dados.

O processo de Mineração de Dados é definido em Fayyad, Piatetsky-Shapiro, & Smyth (1996b) como:

Processo de identificação de padrões válidos, inovadores, potencialmente úteis e principalmente compreensíveis em conjuntos de dados.

¹ *Citeseer* é o site de artigos científicos do NEC Research Institute - <http://www.citeseer.org>.

² Em pesquisa ao site <http://www.kdnuggets.com> foi obtida uma lista de 15 ferramentas para Regras de Associação, sendo acrescentadas a essa lista as ferramentas Oracle 9i, Microsoft Data Analyzer 2002 e Weka.

A Extração de Conhecimento de Bases de Dados envolve a realização de várias atividades como:

- Identificação do domínio;
- Seleção e limpeza dos dados;
- Redução do volume de dados;
- Escolha da função de Mineração de Dados;
- Escolha do algoritmo de Mineração de Dados;
- Busca por padrões;
- Avaliação do conhecimento;
- Refinamento do conhecimento.

Essas atividades podem ser agrupadas em etapas e repetidas por meio de várias iterações (caso seja necessário), o que, às vezes, torna o processo não trivial.

O conhecimento extraído dos dados deve ser válido para um novo conjunto de dados com um determinado grau de certeza. Esse conhecimento também deve ser compreensível (essa compreensão pode ocorrer após algum Pós-processamento) para que possa ser utilizado em Sistemas Inteligentes ou como apoio em processos de tomada de decisão. Além disso, o conhecimento deve ser inovador e potencialmente útil.

Existem diversas abordagens para a divisão do processo de Mineração de Dados em etapas. Em Fayyad, Piatetsky-Shapiro, & Smyth (1996b) é proposta a divisão do processo em nove etapas – Compreensão do Domínio, Seleção e Preparação dos Dados, Mineração de Dados, Avaliação do Conhecimento e Utilização do Conhecimento, sendo que a etapa de Mineração de Dados é subdivida em Escolha das Atividades ou Funções, Escolha de Algoritmos, Preparação dos Dados, Extração de Conhecimento e Pós-Processamento. Em Weiss & Indurkha (1998), essa divisão é composta por apenas quatro etapas – Preparação dos Dados, Redução dos Dados, Modelagem e Predição dos Dados e, Análise dos Casos e Soluções. No entanto, neste trabalho é adotada a divisão utilizada em Rezende, Pugliesi, Melanda, & Paula (2003), que considera um ciclo com três grandes etapas – Pré-Processamento, Extração de Padrões e Pós-Processamento. Essas etapas, ilustradas na Figura 2.1, são precedidas pela fase de Identificação do Problema e sucedidas pela fase de Utilização do Conhecimento.

Figura 2.1: Etapas do processo de Mineração de Dados. Fonte: (Rezende, Pugliesi, Melanda, & Paula 2003)

2.1.1 Identificação do Problema

A compreensão do domínio da aplicação é um pré-requisito para extrair conhecimento útil dos dados.

Nessa fase é realizado um estudo do domínio da aplicação e são definidos os objetivos e metas a serem alcançados com o processo de Mineração de Dados. Também são identificados e selecionados os conjuntos de dados a serem utilizados para a extração de conhecimento.

Para que informações úteis possam ser obtidas, as pessoas envolvidas no processo de Mineração de Dados devem ter algum grau de entendimento sobre a área na qual será aplicado o processo.

2.1.2 Pré-Processamento

Geralmente os dados selecionados para o processo de Mineração de Dados não estão em um formato adequado para a extração de conhecimento. Durante o processo de coleta de dados podem ocorrer diversos problemas que devem ser tratados como erros de digitação, geração de dados incorretos ou inconsistentes por sensores, entre outros. Além disso, limitações de memória, tempo de processamento etc, podem impossibilitar a aplicação direta de alguns algoritmos de extração de padrões a todo o conjunto de dados. Todos esses problemas tornam necessária a utilização de métodos para tratamento, limpeza, redução do volume de dados, dentre outros, antes de realizar a etapa de Extração de Padrões.

É importante ressaltar que os objetivos do processo de extração, definidos na fase de Identificação do Problema, devem sempre guiar a execução das atividades de pré-processamento, de maneira que o conjunto de dados utilizado tenha as características necessárias para se atingir tais objetivos.

A seguir, são apresentadas diversas atividades que podem ser aplicadas a um conjunto de dados durante a etapa de Pré-Processamento.

Obtenção e unificação

As fontes de dados disponíveis para Mineração de Dados podem ser encontradas em diversos formatos – arquivos texto, arquivos no formato de planilhas, Base de Dados, *Data Warehouse*, entre outros. Assim, torna-se necessária a obtenção e unificação dos dados a partir dessas fontes para produzir uma única fonte de dados no formato atributo-valor (Monard & Baranauskas 2003a), como ilustrado na Tabela 2.1.

Na tabela é representado um conjunto de dados com n exemplos de m atributos. Uma linha i representa a i -ésima transação³ ($i = 1, 2, \dots, n$) e uma coluna j representa o j -ésimo item⁴ ($j = 1, 2, \dots, m$). Um valor $v_{i,j}$ corresponde ao valor do item j na transação i . A coluna Y representa um atributo especial, denominado classe ou atributo meta. Em Mineração de Dados Preditiva, o valor desse atributo é o valor a ser predito pelo modelo encontrado para novos exemplos. Já em Mineração de Dados Descritiva, como no caso de Regras de Associação, o atributo meta não é definido.

³Neste trabalho será utilizado o termo transação ao invés de exemplo porque no caso das Regras de Associação, geralmente, o conjunto de exemplos é composto por um conjunto de transações.

⁴Em Regras de Associação, geralmente é adotado o termo item ao invés de atributo.

Tabela 2.1: Conjunto de exemplos no formato atributo-valor. Fonte:(Monard & Baranauskas 2003a)

	a_1	a_2	$\cdots$	a_m	Y
t_1	v_{11}	v_{12}	$\cdots$	v_{1m}	y_1
t_2	v_{21}	v_{22}	$\cdots$	v_{2m}	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
t_n	v_{n1}	v_{n2}	$\cdots$	v_{nm}	y_n

Limpeza dos dados

Uma vez que o conhecimento extraído de Bases de Dados é geralmente utilizado em Sistemas Inteligentes ou como apoio em processos de tomada de decisão, a qualidade dos dados utilizados durante a extração de conhecimento é de grande relevância. Entretanto, é possível que a qualidade dos dados esteja comprometida em função de erros (por exemplo, erros de digitação ou de leitura dos dados por sensores) gerados no processo de coleta. Nesse caso, pode ser necessária a aplicação de técnicas de limpeza no conjunto de dados a fim de garantir a qualidade dos mesmos.

A limpeza dos dados pode ser realizada utilizando o conhecimento do domínio. Por exemplo, pode-se encontrar registros com valor inválido em algum atributo, granularidade incorreta ou exemplos errôneos. Pode-se também efetuar alguma limpeza independente do domínio, como decisão da estratégia de tratamento de atributos incompletos, remoção de ruído e tratamento de conjunto de exemplos não balanceados (Batista, Carvalho, & Monard 2000; Batista 2003).

Redução do volume de dados

Algumas situações que envolvem limitações de espaço em memória, tempo de processamento, entre outras, podem inviabilizar a utilização de alguns algoritmos de extração de padrões. Como solução para esse problema, pode ser necessária a aplicação de métodos para redução do volume de dados.

Segundo Weiss & Indurkha (1998), a redução do volume dos dados pode ser realizada de três maneiras:

1. **redução do número de exemplos:** deve ser realizada mantendo as características do conjunto de dados original, ou seja, deve-se gerar amostras representativas do conjunto de dados (Glymour, Madigan, Pregibon, & Smyth 1997). A principal

abordagem para redução do número de exemplos é a amostragem aleatória – um método que tende a produzir amostras representativas.

É importante ressaltar que se a amostra não for representativa, ou se a quantidade de exemplos for insuficiente para caracterizar os padrões contidos nos dados, os modelos encontrados podem não apresentar o comportamento dos dados originais;

2. **redução do número de atributos:** pode ser um modo de redução do espaço de busca pela solução. Ocorre por meio da seleção de um subconjunto dos atributos existentes, de maneira que isto não tenha grande impacto na qualidade da solução final. A aplicação desse método pode ser realizada com o apoio do especialista do domínio, uma vez que a remoção de um atributo potencialmente útil para o modelo pode diminuir consideravelmente a qualidade do conhecimento extraído;

3. **redução do número de valores de um atributo:** para essa tarefa são aplicados geralmente métodos de discretização ou de suavização de valores de um atributo.

Discretização de um atributo consiste na substituição de um atributo contínuo (inteiro ou real) por um atributo discreto, por meio do agrupamento de seus valores. Em essência, um algoritmo de discretização tem como entrada os valores de um atributo contínuo e gera como saída uma pequena lista de intervalos ordenados. Cada intervalo é representado na forma $[V_{inferior} : V_{superior}]$, de tal modo que $V_{inferior}$ e $V_{superior}$ são os limites inferior e superior do intervalo.

Na suavização de valores, o número de valores distintos de um atributo é reduzido sem discretizá-lo. Durante o processo, eles são agrupados de modo que cada grupo de valores é substituído por um valor numérico que o represente. Esse novo valor pode ser a média, a mediana ou mesmo os valores das extremidades de cada grupo (Weiss & Indurkha 1998).

As transformações descritas na etapa de Pré-Processamento devem ser realizadas criteriosamente e com o devido cuidado, uma vez que é fundamental garantir que as informações presentes nos dados originais continuem presentes nas amostras geradas, de modo que os modelos finais representem o comportamento dos dados originais.

Além disso, como o processo de Mineração de Dados é iterativo, algumas atividades de pré-processamento podem ser realizadas novamente, após a análise dos padrões encontrados na etapa de Extração de Padrões, buscando, assim, melhorar a qualidade do conhecimento extraído.

2.1.3 Extração de Padrões

A etapa de Extração de Padrões é direcionada a cumprir os objetivos definidos na fase de Identificação do Problema. Nessa etapa são realizadas a escolha, a configuração e a execução de um ou mais algoritmos para a extração de conhecimento.

Por se tratar de um processo iterativo, pode ser necessário que essa etapa seja realizada várias vezes para ajustar o seu conjunto de parâmetros, visando a obtenção de resultados mais adequados aos objetivos pré-estabelecidos. Os ajustes podem, por exemplo, melhorar a precisão ou a compreensibilidade do conhecimento extraído.

Segundo Fayyad, Piatetsky-Shapiro, & Smyth (1996b), a essa etapa compreendem as atividades de identificação da função de Mineração de Dados que será empregada, a escolha do algoritmo e a extração dos padrões. Sendo que neste trabalho será acrescida uma atividade de transformação de dados que irá adequá-los ao formato de entrada dos algoritmos de extração de padrões. As atividades da etapa de Extração de Padrões são descritas a seguir.

Escolha da função

A escolha da função é realizada de acordo com os objetivos desejáveis para a solução a ser encontrada. As funções possíveis na extração de padrões podem ser agrupadas em duas grandes categorias: atividades preditivas e atividades descritivas.

As atividades preditivas consistem na generalização de exemplos com seus respectivos atributos meta conhecidos em um modelo capaz de prever o atributo meta de um novo exemplo. Já as atividades descritivas consistem na identificação de padrões intrínsecos ao conjunto de dados, sendo que esses dados não possuem seus atributos meta especificados.

Escolha do algoritmo

Uma vez escolhida a função a ser empregada, existe uma grande variedade de algoritmos para executá-la. A definição do algoritmo de extração e a posterior configuração de seus parâmetros são realizadas por essa atividade. A escolha do algoritmo é feita de maneira subordinada à linguagem de representação dos padrões a serem encontrados. Pode-se, por exemplo, utilizar algoritmos indutores de árvores de decisão ou regras de produção se o objetivo é realizar uma classificação. Dentre os tipos mais frequentes de representação de padrões, destacam-se (Mitchell 1998): árvores de decisão, regras de produção, modelos lineares, modelos não lineares (Redes Neurais Artificiais), modelos baseados em exemplos

(KNN – *K-Nearest Neighbor*, Raciocínio Baseado em Casos) e modelos de dependência probabilística (Redes Bayesianas).

Transformação dos dados

Após a escolha da função e dos algoritmos de extração de padrões, os dados devem ser adequados para que possam ser utilizados pelos algoritmos. Algumas transformações comuns que podem ser aplicadas aos dados para adequá-los ao formato de entrada dos algoritmos, são: resumo, transformação de tipo, normalização e formatação.

Extração de padrões

Consiste na aplicação dos algoritmos selecionados para realizar a extração dos padrões contidos nos dados. É importante ressaltar que dependendo da função adotada (predição ou descrição) podem ser necessárias diversas execuções dos algoritmos de extração de padrões. Por exemplo, com o objetivo de se obter uma avaliação mais precisa da taxa de erro de um classificador, tem-se utilizado métodos de reamostragem (*resampling*).

A reamostragem consiste na realização de vários experimentos em que o classificador é treinado e testado com diferentes conjuntos de treinamento e testes disjuntos. Sendo que, com o resultado de cada um desses experimentos, é obtida uma taxa de erro do classificador para exemplos futuros. Um dos principais métodos de reamostragem é o *k-fold cross-validation*, que subdivide o conjunto de exemplos em k partições mutuamente exclusivas e, em cada iteração do método, $k - 1$ partições são fornecidas ao algoritmo de aprendizado e a partição restante é utilizada para calcular a taxa de erro da iteração. A média das taxas de erro de cada uma das k iterações é usada como a taxa de erro final, estimada pelo método.

No caso das Regras de Associação pode-se realizar apenas uma execução do algoritmo, sem comprometer a qualidade do conhecimento extraído. Segundo Zheng, Kohavi, & Mason (2001), os diversos algoritmos de Regras de Associação devem gerar sempre a mesma solução. Isso porque os parâmetros de entrada para avaliação dos resultados parciais (durante a geração das regras) são geralmente suporte e confiança, e a alteração desses valores irá apenas incluir ou excluir regras do conjunto de regras extraídas.

Após completar essa etapa, o conjunto de padrões obtido deve ser analisado e/ou processado na etapa de Pós-Processamento.

2.1.4 Pós-Processamento

O Pós-Processamento é uma etapa importante do processo de Mineração de Dados na qual o conhecimento extraído pode ser simplificado, avaliado, visualizado ou simplesmente documentado para o usuário final.

Essa etapa consiste de vários métodos e procedimentos que podem ser agrupados nas categorias apresentadas a seguir (Bruha & Famili 2000; Melanda 2002).

Filtragem do conhecimento

Pode ser realizada por meio de mecanismos de pós-poda para o caso de árvores de decisão ou de truncagem no caso de regras de decisão. Esses mecanismos são aplicados nas situações em que os algoritmos de extração de conhecimento geram árvores de decisão com muitas folhas ou regras de decisão muito específicas, cobrindo poucos exemplos (*overfitting*).

Interpretação e explanação

É geralmente aplicada quando o conhecimento obtido é utilizado por um usuário final como apoio em processos de tomada de decisão ou por um Sistema Inteligente. O conhecimento pode ser sumarizado e/ou combinado com o conhecimento prévio do domínio, documentado, visualizado ou modificado de maneira a tornar-se compreensível para o usuário.

Avaliação

Pode ser realizada por meio dos critérios: precisão, compreensibilidade, complexidade computacional, interessabilidade, entre outros. Em Lavrač, Flach, & Zupan (1999); Liu, Hsu, Chen, & Ma (2000) e Adamo (2001) são apresentadas algumas medidas que podem ser empregadas na avaliação do conhecimento obtido por Regras de Associação.

Integração do conhecimento

Os sistemas tradicionais de apoio à decisão são dependentes de uma única técnica, estratégia e modelo. Já os sistemas novos e sofisticados possibilitam combinar ou refinar os resultados de vários modelos de maneira a obter uma maior precisão e um melhor desempenho. Uma das técnicas que pode ser empregada é o uso da combinação de classificadores

(*ensembles*)(Breiman 2000).

Se o conhecimento não for de interesse do usuário final ou não cumprir com os objetivos propostos, o processo de extração pode ser repetido ajustando seus parâmetros ou melhorando o processo de escolha dos dados para a obtenção de resultados melhores numa próxima iteração.

2.1.5 Utilização do Conhecimento

Essa fase sucede o processo de Mineração de Dados que tem como objetivo automatizar a tarefa de extrair conhecimento útil a partir de grandes volumes de dados.

O conhecimento extraído com o uso do processo, depois de ser avaliado e validado na etapa de Pós-Processamento, é consolidado na fase de Utilização do Conhecimento, sendo incorporado a um Sistema Inteligente, utilizado diretamente pelo usuário final para apoio a algum processo de tomada de decisão ou, simplesmente, relatado às pessoas interessadas.

Após ser realizada a consolidação do conhecimento, o mesmo pode ser utilizado para resolver conflitos potenciais entre o conhecimento existente e o conhecimento obtido com o processo de Mineração de Dados (Fayyad, Piatetsky-Shapiro, & Smyth 1996b).

Na próxima seção é apresentada uma das técnicas que podem ser empregadas no Processo de Extração de Conhecimento – Regras de Associação. Serão abordados a definição da técnica, exemplificações e alguns algoritmos.

2.2 Regras de Associação

Na classificação usualmente empregada em Mineração de Dados, a técnica de Regras de Associação pode ser categorizada como uma *Atividade de Mineração de Dados Descritiva* (Weiss & Indurkha 1998; Rezende, Pugliesi, Melanda, & Paula 2003), como é ilustrado na Figura 2.2.

Uma Regra de Associação caracteriza o quanto a presença de um conjunto de itens nos registros de uma Base de Dados implica na presença de algum outro conjunto distinto de itens nos mesmos registros (Agrawal & Srikant 1994). Desse modo, o objetivo das Regras de Associação é encontrar tendências que possam ser usadas para entender e explorar padrões de comportamento dos dados. Por exemplo, observando os dados de vendas de um supermercado, sabe-se que 80% dos clientes que comprem o produto *Q* também adquirem, na mesma ocasião, o produto *W*. Nessa regra 80% corresponde à sua confiabilidade.

O formato de uma Regra de Associação pode ser representado como uma implicação na forma $LHS \Rightarrow RHS$, em que LHS e RHS são, respectivamente, o lado esquerdo (*Left Hand Side*) e o lado direito (*Right Hand Side*) da regra, definidos por conjuntos disjuntos de itens.

Figura 2.2: Contextualização de Regras de Associação em Mineração de Dados. Fonte: (Rezende, Pugliesi, Melanda, & Paula 2003)

2.2.1 Conceitos e Definições

Nesta seção são apresentadas algumas notações e definições baseadas nos trabalhos de Srikant & Agrawal (1997a) e Adamo (2001), que são necessárias à compreensão do processo de mineração de Regras de Associação.

Base de Dados para a Mineração de Regras de Associação

Diferentemente da Tabela 2.1, a Base de Dados utilizada pelos algoritmos de Regras de Associação é uma tabela booleana de itens-transações, como ilustrada na Tabela 2.2.

A Base de Dados utilizada pelos algoritmos de Regras de Associação pode também ser uma tabela de itens-transações, como ilustrado na Tabela 2.3. Na verdade, a tabela de itens-transações é uma simplificação da tabela booleana de itens-transações, na qual itens com valor 1 são mantidos nas transações da tabela de itens-transações e itens com

valor 0 são removidos das transações.

Tabela 2.2: Tabela booleana de itens-transações

	a_1	a_2	$\cdots$	a_m
t_1	1	1	$\cdots$	1
t_2	0	1	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
t_n	1	0	$\cdots$	1

Tabela 2.3: Tabela de itens-transações

t_1	a_1	a_2	$\cdots$	a_m
t_2		a_2	$\cdots$	a_m
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
t_n	a_1		$\cdots$	a_m

Quando a Base de Dados está em um formato distinto aos dois formatos apresentados, deve ser realizado um Pré-processamento dos dados.

Geração de *Itemsets* Frequentes

Tendo-se uma Base de Dados D no formato de uma tabela booleana itens-transações ou de uma tabela itens-transações é possível gerar todos os k -*itemsets*⁵ frequentes e, por consequência, as Regras de Associação. Um *itemset* é definido como um conjunto de itens ordenados lexicograficamente. Já um k -*itemset* é um conjunto de k itens ordenados lexicograficamente. Por fim, um k -*itemset* frequente é definido como um conjunto de k itens ordenados lexicograficamente que possui valor de suporte maior do que um valor de suporte mínimo definido pelo usuário. Uma exemplificação do processo de geração de *itemsets* frequentes é apresentada a seguir. Já a descrição do cálculo da medida suporte é apresentada na próxima seção.

No início do processo de mineração de Regras de Associação, cada *itemset* $X \subseteq A$ (A é um conjunto de itens distintos) é considerado potencialmente frequente. Em outras palavras, o espaço inicial de busca de *itemsets* frequentes consiste de todos os subconjuntos de A com exceção do conjunto vazio. Assim, mesmo os conjuntos com poucos itens tendem a possuir um espaço de busca grande. O espaço de busca para um conjunto de itens $A = \{a, b, c, d, e\}$ é ilustrado na Figura 2.3.

⁵Em Adamo (2001) é adotado o termo *cas* - *Canonical Attribute Sequences* (Seqüência Canônica de Atributos) ao invés de *itemset*.

Figura 2.3: Espaço de busca para o conjunto de itens $A = \{a, b, c, d, e\}$. Fonte: (Hipp, Güntzer, & Nakhaeizadeh 2002)

Para que não haja a necessidade de percorrer todo o espaço de busca à procura dos *itemsets* realmente freqüentes, algoritmos modernos para minerar Regras de Associação utilizam um método que gera e testa *itemsets* candidatos. Esses algoritmos geram conjuntos de *itemsets* potencialmente freqüentes e que são chamados de conjuntos de *itemsets* candidatos. Em seguida, utilizam-se da propriedade de linha de fronteira (*downward closure*) do suporte de um *itemset* (Agrawal & Srikant 1994) – a qual afirma que todo subconjunto de um *itemset* freqüente deve ser freqüente – para remover todos os *itemsets* que tenham pelo menos um subconjunto de itens não-freqüente. Calcula-se, então, o valor de suporte para cada *itemset* candidato (que não foi removido) utilizando a Base de Dados D , removendo, em seguida, os *itemsets* candidatos que possuem suporte inferior ao suporte mínimo definido pelo usuário. O algoritmo inicia uma nova iteração utilizando como entrada os *itemsets* freqüentes gerados na última iteração, encerrando a sua execução apenas quando não houver mais nenhum *itemset* potencialmente freqüente que possa ser considerado um *itemset* candidato.

Na Figura 2.4 a linha em negrito (que representa o suporte mínimo definido pelo usuário) separa os *itemsets* freqüentes (parte superior da linha), dos *itemsets* não-freqüentes (parte inferior da linha). A existência dessa linha é garantida pela propriedade de linha de fronteira do suporte de um *itemset*. Assim, ao invés de percorrer todo o espaço de busca à procura de *itemsets* freqüentes, percorre-se apenas o espaço de busca acima da

linha em negrito, o qual representa os *itemsets* freqüentes.

Figura 2.4: Separação de *itemsets* freqüentes e não-freqüentes no espaço de busca de $A = \{a, b, c, d, e\}$. Fonte: (Hipp, Güntzer, & Nakhaeizadeh 2002)

A obtenção de *itemsets* freqüentes para gerar Regras de Associação pode ser realizada utilizando diversos algoritmos, como: *AIS* (Agrawal, Imielinski, & Swami 1993), *SETM* (Houtsma & Swami 1995) APUD Adamo (2001), *Closet* (Pei, Han, & Mao 2000), *Direct Hashing and Pruning (DHP)* (Park, Chen, & Yu 1997) APUD Adamo (2001), *Charm* (Zaki & Hsiao 2002), *Opus* (Webb 1995), *Dynamic Set Counting (DIC)* (Brin, Motwani, Ullman, & Tsur 1997) APUD Adamo (2001), *Apriori* e *AprioriTid* (Agrawal & Srikant 1994). Embora existam vários algoritmos, teoricamente eles devem gerar sempre um mesmo resultado para um mesmo valor de suporte e confiança, e um mesmo conjunto de dados (Zheng, Kohavi, & Mason 2001). Por ser considerado, sob aspecto histórico, um dos mais importantes algoritmos para gerar *itemsets* freqüentes, o algoritmo *Apriori* é descrito na seção 2.2.2 na página 24.

Cálculo das Medidas Suporte e Confiança

Em Regras de Associação, as medidas mais empregadas são o suporte e a confiança, tanto no que se refere à etapa de Pós-processamento do conhecimento adquirido, como na etapa de seleção dos subconjuntos de itens durante o processo de geração das regras. Buscando facilitar a compreensão das medidas, as mesmas são definidas a seguir:

suporte – quantifica a incidência de um *itemset* X ou de uma regra no conjunto de dados, ou seja, indica a frequência com que X ou com que $LHS \cup RHS$ ocorre no conjunto de dados. Da maneira como foi definido, o suporte para um *itemset* X pode ser representado por:

$$sup(X) = \frac{n(X)}{N} \times 100, \quad (2.1)$$

em que $n(X)$ é o número de transações nas quais X ocorre e N é o número total de transações (exemplos) consideradas. Já o suporte de uma regra $LHS \Rightarrow RHS$ pode ser representado por:

$$sup(LHS \Rightarrow RHS) = sup(LHS \cup RHS) = \frac{n(LHS \cup RHS)}{N} \times 100, \quad (2.2)$$

em que $n(LHS \cup RHS)$ é o número de transações nas quais LHS e RHS ocorrem juntos e N é o número total de transações (exemplos) consideradas.

confiança – indica a frequência com que LHS e RHS ocorrem juntos em relação ao número total de transações em que LHS ocorre. Do modo como foi definida, a confiança de uma regra $LHS \Rightarrow RHS$ pode ser representada por:

$$conf(LHS \Rightarrow RHS) = \frac{sup(LHS \cup RHS)}{sup(LHS)} = \frac{n(LHS \cup RHS)}{n(LHS)} \times 100, \quad (2.3)$$

em que $n(LHS)$ é o número de transações nas quais LHS ocorre.

Formalização da Técnica de Regras de Associação

As Regras de Associação podem ser definidas da maneira descrita a seguir (Agrawal & Srikant 1994):

Seja D uma Base de Dados composta por um conjunto de itens $A = \{a_1, \dots, a_m\}$ ordenados lexicograficamente e por um conjunto de transações $T = \{t_1, \dots, t_n\}$, na qual cada transação $t_i \in T$ é composta por um conjunto de itens tal que $t_i \subseteq A$.

A Regra de Associação é uma implicação na forma $LHS \Rightarrow RHS$, em que $LHS \subset A$, $RHS \subset A$ e $LHS \cap RHS = \emptyset$. A regra $LHS \Rightarrow RHS$ ocorre no conjunto de transações T com confiança $conf$ se em $conf\%$ das transações de T em que ocorre LHS ocorre também RHS . A regra $LHS \Rightarrow RHS$ tem suporte sup se em $sup\%$ das transações em T ocorre $LHS \cup RHS$.

Assim pode-se decompor o problema de obtenção de Regras de Associação em dois passos:

1. Encontrar todos os k -*itemsets* que possuam suporte maior ou igual ao suporte mínimo especificado pelo usuário (**sup-min**). Os *itemsets* com suporte igual ou superior a **sup-min** são definidos como *itemsets* freqüentes, os demais conjuntos são denominados de *itemsets* não-freqüentes. Na seção 2.2.2 é apresentado um algoritmo para encontrar todos os k -*itemsets* freqüentes;
2. Utilizar todos os k -*itemsets* freqüentes, com $k \geq 2$, para gerar as Regras de Associação. Para cada *itemset* freqüente $l \subseteq A$, encontrar todos os subconjuntos $\tilde{a}$ de itens não vazios de l . Para cada subconjunto $\tilde{a} \subseteq l$, gerar uma regra na forma $\tilde{a} \Rightarrow (l - \tilde{a})$ se a razão de $sup(l)$ por $sup(\tilde{a})$ é maior ou igual a confiança mínima especificada pelo usuário (**conf-min**).

Com um conjunto de *itemsets* freqüentes $\{abcd\}$ e um subconjunto de *itemsets* freqüentes $\{ab\}$, por exemplo, pode-se gerar uma regra do tipo $ab \Rightarrow cd$, desde que $conf(ab \Rightarrow cd) \geq \mathbf{conf-min}$, em que, $conf(ab \Rightarrow cd) = sup(abcd)/sup(ab)$. Na seção 2.2.3 é apresentado um algoritmo para gerar regras a partir de *itemsets* freqüentes.

No Exemplo 1 é mostrado como se realiza a extração de Regras de Associação utilizando os 2 passos descritos.

Exemplo 1 *Seja D uma Base de Dados que contém um conjunto de itens $A = \{\text{bermuda, calça, camiseta, sandália, tênis}\}$ e um conjunto de transações $T = \{1, 2, 3, 4\}$, no qual a relação de itens comprados por cada transação t_i é apresentada na Tabela 2.4.*

Tabela 2.4: Relação de itens comprados por transação

Transações	Itens comprados
1	calça, camiseta, tênis
2	camiseta, tênis
3	bermuda, tênis
4	calça, sandália

*Considerando o valor de **sup-min** = 50% (2 transações) e **conf-min** = 50%, é possível obter todas as Regras de Associação contidas na Tabela 2.4 utilizando os dois passos descritos anteriormente.*

1. Encontrar todos os k -itemsets contidos na Tabela 2.4 e que possuam suporte maior ou igual a **sup-min** (itemsets freqüentes). Na Tabela 2.5 são apresentados todos os k -itemsets freqüentes.

Tabela 2.5: Itemsets freqüentes

Itemsets Freqüentes	Suporte
{tênis}	75%
{calça}	50%
{camiseta}	50%
{camiseta, tênis}	50%

2. Com os k -itemsets freqüentes, em que $k \geq 2$, gerar todas as Regras de Associação contidas na Tabela 2.4, que nesse exemplo são:

regra 1: $tênis \Rightarrow camiseta$,

- $suporte = suporte(\{tênis, camiseta\}) = 50\%$, que é igual a **sup-min**.
- $confiança = \frac{suporte(\{tênis, camiseta\})}{suporte(\{tênis\})} = \frac{50}{75} = 66,66\%$, que é maior do que **conf-min**.

regra 2: $camiseta \Rightarrow tênis$,

- $suporte = suporte(\{camiseta, tênis\}) = 50\%$, que é igual a **sup-min**.
- $confiança = \frac{suporte(\{camiseta, tênis\})}{suporte(\{camiseta\})} = \frac{50}{50} = 100\%$, que é maior do que **conf-min**.

Nas duas próximas seções são apresentados o algoritmo *Apriori* (considerado um dos mais importantes sob aspecto histórico) e um algoritmo simples, proposto por Agrawal & Srikant (1994), que gera Regras de Associação. O *Apriori* extrai todos os *itemsets* freqüentes que estão contidos em uma Base de Dados e o algoritmo simples utiliza os *itemsets* extraídos para gerar Regras de Associação.

2.2.2 O Algoritmo *Apriori*

O algoritmo *Apriori*, desenvolvido por Agrawal & Srikant (1994), é utilizado para encontrar todos os k -itemsets freqüentes contidos em uma Base de Dados. Esse algoritmo gera um conjunto de k -itemsets candidatos e então percorre a Base de Dados para determinar se os mesmos são freqüentes, identificando desse modo todos os k -itemsets freqüentes. O algoritmo *Apriori* é apresentado no Algoritmo 1. Nele é utilizada a notação

L_k para representar o conjunto de k -itemsets freqüentes e C_k para representar o conjunto de k -itemsets candidatos.

Algoritmo 1 *Apriori*. Fonte: (Agrawal & Srikant 1994)

Require: Uma Base de Dados D composta por um conjunto de itens $A = \{a_1, \dots, a_m\}$ ordenados lexicograficamente e por um conjunto de transações $T = \{t_1, \dots, t_n\}$, na qual cada transação $t_i \in T$ é composta por um conjunto de itens tal que $t_i \subseteq A$.

```

1:  $L_1 := \{1\text{-itemsets freqüente}\};$ 
2: for ( $k := 2; L_{k-1} \neq \emptyset; k++$ ) do
3:    $C_k := \text{apriori-gen}(L_{k-1});$  //Gera novos conjuntos candidatos
4:   for all (transações  $t \in T$ ) do
5:      $C_t := \text{subset}(C_k, t);$  //Conjuntos candidatos contidos em  $t$ 
6:     for all candidatos  $c \in C_t$  do
7:        $c.\text{count}++;$ 
8:     end for
9:   end for
10:   $L_k := \{c \in C_k \mid c.\text{count} \geq \text{sup-min}\};$ 
11: end for
12: Resposta  $:= \bigcup_k L_k;$ 

```

Inicialmente o algoritmo simplesmente conta a ocorrência de itens, determinando os 1 -itemsets freqüentes que são armazenados em L_1 . O passo seguinte, dito passo k , é dividido em duas etapas. Na primeira (linha 3 do Algoritmo 1) o conjunto de itemsets freqüentes L_{k-1} obtido no passo $(k-1)$ é utilizado para gerar o conjunto de k -itemsets candidatos C_k usando a função *apriori-gen*, descrita no Algoritmo 2. A seguir (linhas 4 a 9 do Algoritmo 1), a Base de Dados é percorrida para determinar o valor do suporte dos k -itemsets candidatos em C_k . Finalmente, são identificados os k -itemsets freqüentes de cada passo (linha 10). A solução final é dada pela união dos conjuntos L_k de k -itemsets freqüentes (Agrawal & Srikant 1994). Essa solução é utilizada como entrada para algum algoritmo que gera Regras de Associação, como o Algoritmo 3 descrito na página 30.

A seguir são apresentadas as funções *apriori-gen* e *subset* que fazem parte do algoritmo *Apriori*.

Função *apriori-gen*

A função *apriori-gen* é apresentada no Algoritmo 2. Essa função usa como argumento L_{k-1} e retorna um superconjunto do conjunto de todos os $(k-1)$ -itemsets. A função executa inicialmente um *join* dos elementos dos itemsets em L_{k-1} com o último elemento de outros itemsets, diferentes do primeiro, em L_{k-1} (linhas 1 a 4 do Algoritmo 2). Em seguida são removidos os k -itemsets que possuem algum subconjunto de tamanho $(k-1)$

não pertencente à L_{k-1} (linhas 5 a 11 do Algoritmo 2). Essa remoção (linhas 5 a 11 do Algoritmo 2) é garantida pela propriedade de linha de fronteira (*downward closure*) do suporte de um *itemset*, descrita anteriormente neste capítulo.

Algoritmo 2 Função apriori-gen. Fonte: (Agrawal & Srikant 1994)

Require: Um conjunto de *itemsets* freqüentes L_{k-1} .

```

1: insert into  $C_k$ 
2: select  $p.item_1, p.item_2, \dots, p.item_{k-1}, q.item_{k-1}$  from  $L_{k-1}p, L_{k-1}q$ 
3: where  $p.item_1 = q.item_1, \dots, p.item_{k-2} = q.item_{k-2},$ 
4:       $p.item_{k-1} < q.item_{k-1};$ 
   // A seguir, é realizada a etapa de poda, onde todos os itemsets  $c \in C_k$  são removidos se
   // algum  $(k-1)$ -subconjunto de  $c$  não pertencer a  $L_{k-1}$ .
5: for all itemset  $c \in C_k$  do
6:   for all  $(k-1)$ -subconjuntos  $s$  de  $c$  do
7:     if  $(s \notin L_{k-1})$  then
8:       delete  $c$  de  $C_k$ ;
9:     end if
10:  end for
11: end for

```

No Exemplo 2 é apresentado como a função apriori-gen pode ser utilizada para gerar os *itemsets* candidatos.

Exemplo 2 Suponha um conjunto qualquer de 3-*itemsets* freqüentes $L_3 = \{\{bermuda, calça, camiseta\}, \{bermuda, calça, sandália\}, \{bermuda, camiseta, tênis\}, \{bermuda, camiseta, sandália\}, \{calça, camiseta, sandália\}\}$. Aplicando a função apriori-gen no conjunto de 3-*itemsets* freqüentes, tem-se:

- As linhas 1 a 4 do Algoritmo 2 realizando um join dos itens de L_3 e gerando um conjunto de 4-*itemsets* candidatos $C_4 = \{\{bermuda, calça, camiseta, sandália\}, \{bermuda, camiseta, sandália, tênis\}\}$.
- Já as linhas de 5 a 11 irão remover de C_4 o conjunto $\{bermuda, camiseta, sandália, tênis\}$, uma vez que, o subconjunto $\{bermuda, sandália, tênis\}$ desse conjunto de itens não está contido em L_3 .

Assim a função apriori-gen retorna como resultado o conjunto de 4-*itemsets* candidatos $C_4 = \{\{bermuda, calça, camiseta, sandália\}\}$.

Função subset

A função subset retorna os k -itemsets candidatos que estão contidos em uma dada transação t_i . Para isso os itemsets candidatos são armazenados em uma árvore-hash. Cada nó da árvore pode conter uma lista de itemsets ou uma tabela hash (nó folha ou nó intermediário, respectivamente). Partindo do nó raiz, a função encontra todos os itemsets candidatos presentes na transação t_i . Se um nó folha é atingido e o itemset encontrado está contido na transação t_i , uma referência é adicionada ao conjunto de resposta. Se um nó intermediário é atingido a partir de um item a_j , pesquisa-se (hash) em cada item após a_j em t_i . Isso é possível porque os itens estão em ordem lexicográfica. No nó raiz, todos os itens a_j em t_i são pesquisados. No Exemplo 3 é ilustrado como a função subset é utilizada.

Exemplo 3 Dado um conjunto qualquer de 2-itemsets candidatos $C_2 = \{\{bermuda, calça\}, \{bermuda, camiseta\}, \{camiseta, meia\}, \{sandália, tênis\}\}$ armazenado em uma árvore-hash, conforme apresentado na figura 2.5 (neste exemplo a árvore possui apenas um nível) e uma transação $t_i = \{camiseta, meia, tênis\}$, a função subset irá encontrar todos os itemset contidos na transação verificando apenas as células correspondentes aos valores contidos na transação $t_i = \{camiseta, meia, tênis\}$, o que resultará no 2-itemset $\{camiseta, meia\}$.

Figura 2.5: Itemsets candidatos armazenados em uma árvore-hash

Para finalizar esta seção, no Exemplo 4 é apresentado o funcionamento do algoritmo *Apriori* aplicado a um pequeno conjunto de transações. O espaço de busca do conjunto de itens $A = \{bermuda, calça, camiseta, sandália, tênis\}$ utilizado no Exemplo 4 é ilustrado na Figura 2.6.

Figura 2.6: Espaço de busca do conjunto de itens $A = \{\text{bermuda}, \text{calça}, \text{camiseta}, \text{sandália}, \text{tênis}\}$

Exemplo 4 Seja D uma Base de Dados que contém um conjunto de itens $A = \{\text{bermuda, calça, camiseta, sandália, tênis}\}$ e um conjunto de transações $T = \{1, 2, 3, 4\}$, no qual a relação de itens comprados por cada transação t_i é apresentada na Tabela 2.4 do Exemplo 1. O valor do suporte mínimo (**sup-min**) é igual a 50%.

Ao aplicar o Apriori na Base de Dados D apresentada na Tabela 2.4, o algoritmo irá percorrer a Base de Dados gerando um conjunto C_1 de 1-itemsets candidatos e em seguida gerará um conjunto L_1 de 1-itemsets freqüentes (linha 1 do Algoritmo 1), como pode ser visto a seguir.

<p>Percorre a Base D</p> <p>$\Rightarrow$</p>	C_1		L_1																					
		<table> <tr> <th><i>Itemset</i></th> <th><i>Suporte</i></th> </tr> <tr> <td>{calça}</td> <td>50%</td> </tr> <tr> <td>{camiseta}</td> <td>50%</td> </tr> <tr> <td>{tênis}</td> <td>75%</td> </tr> <tr> <td>{bermuda}</td> <td>25%</td> </tr> <tr> <td>{sandália}</td> <td>25%</td> </tr> </table>	<i>Itemset</i>	<i>Suporte</i>	{calça}	50%	{camiseta}	50%	{tênis}	75%	{bermuda}	25%	{sandália}	25%		<table> <tr> <th><i>Itemset</i></th> <th><i>Suporte</i></th> </tr> <tr> <td>{calça}</td> <td>50%</td> </tr> <tr> <td>{camiseta}</td> <td>50%</td> </tr> <tr> <td>{tênis}</td> <td>75%</td> </tr> </table>	<i>Itemset</i>	<i>Suporte</i>	{calça}	50%	{camiseta}	50%	{tênis}	75%
	<i>Itemset</i>	<i>Suporte</i>																						
	{calça}	50%																						
	{camiseta}	50%																						
	{tênis}	75%																						
	{bermuda}	25%																						
{sandália}	25%																							
<i>Itemset</i>	<i>Suporte</i>																							
{calça}	50%																							
{camiseta}	50%																							
{tênis}	75%																							

Depois, o algoritmo executa a função apriori-gen (linha 3 do Algoritmo 1) gerando um conjunto C_2 de 2-itemsets candidatos. O próximo passo é verificar quais dos 2-itemsets candidatos em C_2 são freqüentes, que é realizado pelas linhas 4 a 11 do Algoritmo 1, gerando um conjunto L_2 de 2-itemsets freqüentes, como é apresentado a seguir.

C_2		C_2		L_2																
<table> <tr><th><i>Itemset</i></th></tr> <tr><td>{calça,camiseta}</td></tr> <tr><td>{calça,tênis}</td></tr> <tr><td>{camiseta,tênis}</td></tr> </table>	<i>Itemset</i>	{calça,camiseta}	{calça,tênis}	{camiseta,tênis}	<p><i>Percorre</i> <i>a Base D</i></p> <p>$\Rightarrow$</p>	<table> <tr><th><i>Itemset</i></th><th><i>Suporte</i></th></tr> <tr><td>{calça,camiseta}</td><td>25%</td></tr> <tr><td>{calça,tênis}</td><td>25%</td></tr> <tr><td>{camiseta,tênis}</td><td>50%</td></tr> </table>	<i>Itemset</i>	<i>Suporte</i>	{calça,camiseta}	25%	{calça,tênis}	25%	{camiseta,tênis}	50%		<table> <tr><th><i>Itemset</i></th><th><i>Suporte</i></th></tr> <tr><td>{camiseta,tênis}</td><td>50%</td></tr> </table>	<i>Itemset</i>	<i>Suporte</i>	{camiseta,tênis}	50%
<i>Itemset</i>																				
{calça,camiseta}																				
{calça,tênis}																				
{camiseta,tênis}																				
<i>Itemset</i>	<i>Suporte</i>																			
{calça,camiseta}	25%																			
{calça,tênis}	25%																			
{camiseta,tênis}	50%																			
<i>Itemset</i>	<i>Suporte</i>																			
{camiseta,tênis}	50%																			

Vários passos são realizados pelo algoritmo Apriori até não ser mais possível gerar um conjunto de itemsets freqüentes.

C_3		L_3				
<table> <tr><th><i>Itemset</i></th></tr> <tr><td>$\emptyset$</td></tr> </table>	<i>Itemset</i>	$\emptyset$	<i>Percorre a Base D</i> $\Rightarrow$	<table> <tr><th><i>Itemset</i></th></tr> <tr><td>$\emptyset$</td></tr> </table>	<i>Itemset</i>	$\emptyset$
<i>Itemset</i>						
$\emptyset$						
<i>Itemset</i>						
$\emptyset$						

Ao término de sua execução é gerado o conjunto *Resposta* (Tabela 2.6) pela união de todos os conjuntos L_k que contêm os k -itemsets freqüentes (linha 12 do Algoritmo 1).

Tabela 2.6: Conjunto *Resposta* contendo os *itemsets* freqüentes

<i>Itemsets</i> Freqüentes	Suporte
{tênis}	75%
{calça}	50%
{camiseta}	50%
{camiseta, tênis}	50%

O conjunto *Resposta* é utilizado como entrada para algum algoritmo que gera Regras de Associação, como o algoritmo apresentado na Seção 2.2.3.

Na próxima seção é apresentado um algoritmo que utiliza conjuntos de *itemsets* freqüentes (extraídos de Bases de Dados com o uso de algoritmos como o *Apriori*) para gerar Regras de Associação.

2.2.3 Algoritmo Simples para Gerar Regras de Associação

Existem diversos algoritmos que geram Regras de Associação a partir dos *itemsets* freqüentes obtidos de uma Base de Dados. Um dos algoritmos mais simples foi proposto por Agrawal & Srikant (1994) e é apresentado no Algoritmo 3.

Algoritmo 3 Gera Regras de Associação. Fonte: (Agrawal & Srikant 1994)

Require: Um conjunto *Resposta* contendo todos os k -itemsets freqüentes, com $k \geq 2$.

- 1: **for all** (k -itemset freqüente $l_k, k \geq 2$) **do**
- 2: **Call** genrules (l_k, l_k);
- 3: **end for**
- //O procedimento genrules gera todas as regras válidas $\tilde{a} \Rightarrow (l_k - \tilde{a})$, para todo $\tilde{a} \subset a_m$
- 4: **procedure** genrules (l_k : k -itemset freqüente, a_m : m -itemset freqüente)
- 5: $A := \{(m-1) - \text{itemsets } a_{m-1} \mid a_{m-1} \subset a_m\}$;
- 6: **for all** ($a_{m-1} \in A$) **do**
- 7: $conf := sup(l_k) / sup(a_{m-1})$;
- 8: **if** ($conf \geq \text{conf-min}$) **then**
- 9: **Output:** regra $a_{m-1} \Rightarrow (l_k - a_{m-1})$, com $confiança = conf$ e $suporte = sup(l_k)$;
- 10: **if** ($m-1 > 1$) **then**
- 11: **Call** genrules (l_k, a_{m-1}); //Gera regras com subconjuntos de a_{m-1} como antecedente
- 12: **end if**
- 13: **end if**
- 14: **end for**

O algoritmo deve ser executado para todos os k -itemsets freqüentes, com $k \geq 2$. Primeiramente, ele irá gerar subconjuntos não vazios de um itemset freqüente. Em seguida, os subconjuntos são utilizados para gerar Regras de Associação do tipo $LHS \Rightarrow RHS$, desde que a confiança da regra seja maior ou igual a confiança mínima especificada pelo usuário (**conf-min**).

No Exemplo 5, é ilustrado como uma Regra de Associação pode ser gerada a partir de um conjunto de itemsets freqüentes. Nesse exemplo será considerado um dos conjuntos de itemsets freqüentes contidos na Tabela 2.6 do Exemplo 4.

Exemplo 5 Seja $resposta_4 = \{camiseta, tênis\}$, o 2-itemset freqüente contido no conjunto Resposta apresentado na Tabela 2.6. Dado uma confiança mínima (**conf-min**) igual a 50%, o procedimento genrules primeiramente irá gerar todos os subconjuntos $\tilde{a}_j$ a partir de $resposta_4$ (linha 5 do Algoritmo 3):

$$\text{Subconjunto } \tilde{a}_1 = \{tênis\} \text{ e subconjunto } \tilde{a}_2 = \{camiseta\}.$$

O próximo passo do algoritmo consiste em calcular a confiança desses subconjuntos e gerar uma Regra de Associação para cada subconjunto que tenha confiança maior ou igual a **conf-min** (linhas 6 a 9 do Algoritmo 3), como é apresentado a seguir.

regra gerada pelo subconjunto $\tilde{a}_1$: $tênis \Rightarrow camiseta$,

- $suporte = suporte(\{tênis, camiseta\}) = 50\%$.
- $confiança = \frac{suporte(\{tênis, camiseta\})}{suporte(\{tênis\})} = \frac{50}{75} = 66,66\%$, que é maior do que **conf-min**.

regra gerada pelo subconjunto $\tilde{a}_2$: $camiseta \Rightarrow tênis$,

- $suporte = suporte(\{camiseta, tênis\}) = 50\%$.
- $confiança = \frac{suporte(\{camiseta, tênis\})}{suporte(\{camiseta\})} = \frac{50}{50} = 100\%$, que é maior do que **conf-min**.

Em seguida é verificado que $m - 1 = 1$ (m possui valor 2), finalizando a execução do algoritmo.

2.3 Considerações Finais

Como já citado, com o avanço da tecnologia de armazenamento a quantidade de dados disponíveis nas Bases de Dados das organizações tem crescido demasiadamente. Esse crescimento inviabilizou o uso de procedimentos manuais para análise de dados e obtenção de conhecimento.

Ao mesmo tempo, as organizações tem percebido que existe a possibilidade de utilizar seus dados para obter conhecimento sobre seus clientes, produtos e parceiros, e desse modo adquirir vantagens competitivas perante seus concorrentes.

Diante desse contexto, pesquisas vem sendo direcionadas para o estudo de processos de Extração de Conhecimento de Bases de Dados, visando automatizar a tarefa de extrair conhecimento útil a partir de grande volume de dados.

O objetivo deste capítulo foi o de apresentar o processo de extração de conhecimento conhecido por Mineração de Dados - um processo de natureza iterativa/iterativa e composto por etapas de Pré-Processamento, Extração de Padrões e Pós-Processamento. Em seguida, foi descrita a técnica de Mineração de Dados para a extração de conhecimento denominada Regras de Associação. Foram apresentadas a definição da técnica, exemplificações e alguns algoritmos.

No próximo capítulo são apresentados os conceitos relacionados ao uso de taxonomias em Regras de Associação. O uso de taxonomias será abordado com o intuito de resolver um dos maiores problemas dessa técnica que é a geração de um grande volume de conhecimento no formato de regras.

Uso de Taxonomias em Regras de Associação

O fato das Regras de Associação permitirem identificar associações entre itens e conjuntos de itens de uma Base de Dados faz com que os algoritmos produzam grandes quantidades de regras, muitas das quais não são interessantes para o usuário (Liu, Hsu, Chen, & Ma 2000).

Devido a essa grande quantidade de regras, a análise e a compreensão do conhecimento torna-se difícil para o usuário. A aplicação de taxonomias em Regras de Associação pode ser utilizada para reduzir o volume de regras extraídas e, por consequência, facilitar a análise e compreensão do conhecimento.

As taxonomias permitem realizar uma classificação hierárquica dos itens por meio de uma caracterização coletiva ou individual. Além disso, as informações das taxonomias possibilitam a definição de medidas que podem ser utilizadas para remover regras não interessantes segundo algum critério.

Neste capítulo são apresentados aspectos gerais do uso de taxonomias em Regras de Associação. Na Seção 3.1 é apresentada a mineração de Regras de Associação usando taxonomias. Na Seção 3.2 são apresentados alguns algoritmos para encontrar *itemsets* freqüentes aplicando taxonomias. Já na Seção 3.3 é descrita uma medida que permite

remover regras não interessantes utilizando as informações das taxonomias aplicadas à regra. Por fim, na Seção 3.4 são feitas algumas considerações finais sobre o capítulo.

3.1 Taxonomias e Regras de Associação

As taxonomias refletem uma caracterização coletiva ou individual de como os itens podem ser hierarquicamente classificados (Adamo 2001). Eventualmente, múltiplas taxonomias podem estar presentes simultaneamente, refletindo a existência de diversos pontos de vista ou a possibilidade de classificações distintas para o mesmo conjunto de itens. Na Figura 3.1 é apresentado um pequeno exemplo de uma taxonomia. Nesse exemplo pode-se verificar que: camiseta é uma roupa leve, bermuda é uma roupa leve, roupa leve é um tipo de roupa, sandália é um tipo de calçado, etc.

Figura 3.1: Exemplo de uma taxonomia para vestuário

Entretanto, existem algumas taxonomias cuja caracterização hierárquica dos itens não pode ser classificada. Um exemplo de uma taxonomia sem classificação é ilustrada na Figura 3.2, na qual pode-se verificar que não há uma classificação para *PRODUTOS1* e *PRODUTOS2*.

Do ponto de vista da lógica proposicional, a introdução de taxonomias é uma maneira de se complementar a capacidade descritiva das Regras de Associação com disjunções. Por exemplo, camisetas **ou** bermudas $\Rightarrow$ tênis **ou** sandálias pode ser uma representação

da regra roupas leves $\Rightarrow$ calçados. Observa-se que essa é uma maneira restritiva de introdução de disjunções, uma vez que os argumentos de disjunção não são automaticamente descobertos, como os argumentos de conjunção¹. Esses argumentos de disjunção devem ser introduzidos nas regras pelas taxonomias especificadas pelo usuário (Adamo 2001).

Figura 3.2: Exemplo de uma taxonomia sem classificação para vestuário

Uma das razões que torna interessante a aplicação de taxonomias em Regras de Associação refere-se à possibilidade de geração de regras que utilizem itens contidos em qualquer nível na taxonomia e não apenas os itens contidos no nível inferior da taxonomia, como ocorre em Regras de Associação que não se utilizam de taxonomias (Srikant & Agrawal 1997a). Por exemplo, usando a taxonomia apresentada na Figura 3.1, pode-se inferir uma regra em que pessoas que comprem Roupas Leves também comprem Sandálias, do fato de pessoas comprarem Camisetas com Sandálias e Bermudas com Sandálias, sem necessidade de utilizar as regras: pessoas que comprem Camisetas também comprem Sandálias e pessoas que comprem Bermudas também comprem Sandálias.

Outro motivo que torna o uso de taxonomias em Regras de Associação interessante é que as informações contidas nas taxonomias podem ser utilizadas na avaliação subjetiva do conhecimento (gerado no formato de Regras de Associação), em termos de compreensibilidade e interessabilidade/surpresa (Liu, Hsu, Chen, & Ma 2000). Srikant (2001) apresenta outras razões para o emprego de taxonomias em Regras de Associação:

- regras simples (cujos elementos são compostos apenas por itens terminais na taxonomia) podem não ter suporte suficiente para serem incluídas na solução, mas podem representar conhecimento interessante ao serem agrupadas segundo uma taxonomia;

¹As conjunções estão no antecedente e conseqüente de uma Regra de Associação de maneira implícita.

- regras muito específicas podem ser generalizadas. De maneira similar ao item anterior, mesmo considerando as regras simples com elevados níveis de suporte e confiança, essas podem ser agrupadas em regras mais gerais, melhorando a sua compreensibilidade e ainda podendo aumentar os valores de suporte e confiança;
- regras interessantes podem ser identificadas com o uso de informações contidas nas taxonomias. A interessabilidade de uma regra pode ser baseada em sua utilidade e inesperabilidade (Silberschatz & Tuzhilin 1995).

A seguir são apresentados alguns conceitos necessários para se ter uma melhor compreensão do uso de taxonomias em Regras de Associação.

3.1.1 Notações e Definições

Nesta seção são apresentadas algumas notações e definições baseadas nos trabalhos de Srikant & Agrawal (1997a) e Adamo (2001):

Generalização ou Ancestral Considerando X um *itemset*, em que $X \subseteq LHS$ ou $X \subseteq RHS$ e $LHS \Rightarrow RHS$ uma Regra de Associação, as notações $X \uparrow$ e $(LHS \Rightarrow RHS) \uparrow$ representam novos *itemsets* e regras que, respectivamente, derivam de X e $LHS \Rightarrow RHS$ pela substituição de um ou mais itens pelos seus ancestrais na taxonomia. Os novos *itemsets* e regras são ditas generalizações ou ancestrais de X e $LHS \Rightarrow RHS$.

Especialização ou Descendente Considerando X um *itemset*, em que $X \subseteq LHS$ ou $X \subseteq RHS$ e $LHS \Rightarrow RHS$ uma Regra de Associação, as notações $X \downarrow$ e $(LHS \Rightarrow RHS) \downarrow$ representam novos *itemsets* e regras que, respectivamente, derivam de X e $LHS \Rightarrow RHS$ pela substituição de um ou mais itens pelos seus descendentes na taxonomia. Os novos *itemsets* e regras são ditas especializações ou descendentes de X e $LHS \Rightarrow RHS$.

Pai Um *itemset* $X \uparrow$ é dito ser pai de X se não há nenhum *itemset* X' tal que X' é um ancestral de X e $X \uparrow$ é um ancestral de X' .

Filho Um *itemset* $X \downarrow$ é dito ser filho de X se não há nenhum *itemset* X' tal que X' é um descendente de X e $X \downarrow$ é um descendente de X' .

Generalização máxima Um *itemset* X é dito ser uma generalização máxima, se nenhum item em X pode ser substituído por um item ancestral na taxonomia. O mesmo é válido para uma regra $LHS \Rightarrow RHS$ e para um item a .

Especialização máxima Um *itemset* X é dito ser uma especialização máxima, se nenhum item em X pode ser substituído por um item descendente na taxonomia. O mesmo também é válido para uma regra $LHS \Rightarrow RHS$ e para um item a .

Uma Regra de Associação usando taxonomias pode ser definida como (Srikant & Agrawal 1997a):

Seja D uma Base de Dados composta por um conjunto de itens $A = \{a_1, \dots, a_m\}$ ordenados lexicograficamente e por um conjunto de transações $T = \{t_1, \dots, t_n\}$, na qual cada transação $t_i \in T$ é composta por um conjunto de itens tal que $t_i \subseteq A$. É dito que uma transação t_i suporta um item $a_j \in A$, se a_j está em t_i ou a_j é um ancestral de algum item em t_i . Seja $\mathcal{T}$ um grafo direcional e acíclico com os itens, representando um conjunto de taxonomias. Se há uma aresta em $\mathcal{T}$ de um item $a_p \in A$ para um item $a_c \in A$, a_p é dito ser pai de a_c e a_c é dito ser filho de a_p .

Uma Regra de Associação usando taxonomias é uma implicação na forma $LHS \Rightarrow RHS$, em que $LHS \subset A$, $RHS \subset A$, $LHS \cap RHS = \emptyset$ e nenhum item em RHS é um ancestral de qualquer item em LHS . A regra $LHS \Rightarrow RHS$ ocorre no conjunto de transações T com confiança $conf$ se em $conf\%$ das transações de T em que ocorre LHS ocorre também RHS . A regra $LHS \Rightarrow RHS$ tem suporte sup se em $sup\%$ das transações de T ocorre $LHS \cup RHS$.

Assim pode-se decompor o problema de obtenção de Regras de Associação aplicando taxonomias em três passos:

1. Encontrar todos os k -*itemsets* que possuam suporte maior ou igual ao suporte mínimo especificado pelo usuário (**sup-min**). Na seção 3.2 são apresentados alguns algoritmos para encontrar todos os k -*itemsets* freqüentes aplicando taxonomias.
2. Utilizar todos os k -*itemsets* freqüentes para gerar as Regras de Associação. O algoritmo apresentado na seção 2.2.3 na página 30 pode ser utilizado também para gerar Regras de Associação que usam taxonomias, desde que, os k -*itemsets* freqüentes utilizados como entrada pelo algoritmo tenham sido gerados com a aplicação de alguma taxonomia.
3. Remover todas as regras não interessantes do conjunto de regras. Na seção 3.3 é descrita uma medida e um algoritmo que permitem remover as regras não interessantes usando as informações contidas nas taxonomias utilizadas pela regra.

Na próxima seção é descrito como se realiza o cálculo do suporte de um *itemset* quando uma taxonomia é utilizada para gerar Regras de Associação.

3.1.2 Cálculo do Suporte Utilizando Taxonomias

Em Regras de Associação utilizando taxonomias, o suporte de um item terminal na taxonomia $a_j \in A$ é definido de maneira semelhante à apresentada na Seção 2.2.1 na página 21. O suporte para um item a_j não terminal na taxonomia, utilizando os conceitos apresentados em Adamo (2001), pode ser definido como

$$sup(a_j) = \frac{\# desc(a_j)}{N},$$

em que $desc(a_j)$ é o conjunto de descendentes mais próximos de a_j , sendo que:

- se a_j é um item terminal na taxonomia, $desc(a_j)$ é igual ao conjunto de transações que suportam a_j ;
- se a_j é um item não terminal na taxonomia, $desc(a_j) = \bigcup desc(a_z)$, tal que a_z seja um descendente mais próximos de a_j .

Na definição do suporte para um item não terminal, o símbolo “#” indica a cardinalidade de um conjunto e N é o número total de transações consideradas. No Exemplo 6 é descrito como é calculado o suporte de itens terminais e não terminais na taxonomia.

Exemplo 6 Considerando a taxonomia apresentada na Figura 3.1 e D uma Base de Dados que contém um conjunto de itens $A = \{bermudas, calças, camisetas, sandálias, tênis\}$ e um conjunto de transações $T = \{1, 2, 3, 4, 5, 6, 7\}$, no qual a relação de itens comprados por cada transação t_i é apresentada na Tabela 3.1.

Tabela 3.1: Relação de itens comprados por transação

Transações	Itens comprados
1	camisetas, calças, tênis
2	bermudas, tênis, sandálias
3	camisetas, bermudas, calças, tênis, sandálias
4	tênis, sandálias
5	camisetas, bermudas, calças
6	calças, tênis, sandálias
7	bermudas, calças, sandálias

A partir dessas informações são calculados os suportes dos itens terminais e não terminais da taxonomia:

- *Suporte dos Itens Terminais*

$$sup(camisetas) = n(camisetas)/N = 3/7 = 42\%$$

$$sup(bermudas) = n(bermudas)/N = 4/7 = 57\%$$

$$sup(calças) = n(calças)/N = 5/7 = 71\%$$

$$sup(tênis) = n(tênis)/N = 5/7 = 71\%$$

$$sup(sandálias) = n(sandálias)/N = 5/7 = 71\%$$

- *Suporte dos Itens Não Terminais*

$$\begin{aligned} sup(roupas leves) &= (\#desc(roupas leves))/N \\ &= (\#(desc(camisetas) \cup desc(bermudas)))/7 \\ &= (\#(\{1, 3, 5\} \cup \{2, 3, 5, 7\}))/7 \\ &= (\#\{1, 2, 3, 5, 7\})/7 \\ &= 5/7 = 71\% \end{aligned}$$

$$\begin{aligned} sup(calçados) &= (\#desc(calçados))/N \\ &= (\#(desc(tênis) \cup desc(sandálias)))/7 \\ &= (\#(\{1, 2, 3, 4, 6\} \cup \{2, 3, 4, 6, 7\}))/7 \\ &= (\#\{1, 2, 3, 4, 6, 7\})/7 \\ &= 6/7 = 85\% \end{aligned}$$

$$\begin{aligned} sup(roupas) &= (\#desc(roupas))/N \\ &= (\#(desc(roupas leves) \cup desc(calças)))/7 \\ &= (\#(desc(camisetas) \cup desc(bermudas) \cup desc(calças)))/7 \\ &= (\#(\{1, 3, 5\} \cup \{2, 3, 5, 7\} \cup \{1, 3, 5, 6, 7\}))/7 \\ &= (\#\{1, 2, 3, 5, 6, 7\})/7 \\ &= 6/7 = 85\% \end{aligned}$$

Nas seções 3.2 e 3.3 são apresentados, respectivamente, algoritmos para encontrar todos os k -*itemsets* freqüentes aplicando taxonomias e uma medida para remover regras não interessantes usando as informações contidas nas taxonomias aplicadas à regra.

3.2 Algoritmos para Gerar *Itemsets* Freqüentes com Taxonomias

Os algoritmos citados no Capítulo 2 geram Regras de Associação baseadas em conjuntos de itens nivelados (itens terminais na taxonomia). Nesta seção são apresentados alguns algoritmos para encontrar os k -*itemsets* freqüentes aplicando taxonomias. Esses *itemsets* são utilizados na geração de Regras de Associação compostas por itens contidos nos vários níveis das taxonomias.

3.2.1 Algoritmo Básico para Gerar *Itemsets* Freqüentes com Taxonomias

Em Regras de Associação que usam taxonomias, o problema de verificar se uma transação $t_i \in T$ suporta um *itemset* $X \subseteq A$, em que T e A são, respectivamente, conjunto de transações e itens, consiste em verificar se todo item $x_j \in X$ ou algum descendente de x_j está presente na transação t_i .

O algoritmo básico apresentado em Srikant & Agrawal (1997a) torna essa tarefa mais simples, uma vez que adiciona a T todos os ancestrais de cada item presente nas transações $t_i \in T$. Esse conjunto de transações estendidas é chamado de T' . Assim, T suporta X se, e somente se, T' é um superconjunto de X .

Dessa maneira, os *itemsets* freqüentes usando taxonomias são gerados pela aplicação de qualquer um dos algoritmos, citados no Capítulo 2, no conjunto de transações estendidas T' .

Embora o algoritmo básico seja simples, o mesmo apresenta o problema de gerar e avaliar muitos *itemsets* candidatos redundantes, que contêm itens juntos com seus ancestrais. O algoritmo *Cumulate*, apresentado a seguir, tenta resolver esse problema acrescentando algumas melhorias ao algoritmo básico.

3.2.2 Algoritmo *Cumulate*

O algoritmo *Cumulate* é uma versão melhorada do algoritmo básico (Srikant & Agrawal 1997a). O nome *Cumulate* indica que o suporte é calculado para um conjunto contendo todos os k -*itemsets* candidatos de um determinado tamanho k , percorrendo a Base de Dados apenas uma vez. As melhorias acrescidas ao algoritmo básico são:

Filtrar os ancestrais adicionados as transações Não há necessidade de adicionar a uma transação t_i todos os ancestrais dos itens contidos nessa transação. Pode-se apenas adicionar ancestrais que estão em um (ou mais) *itemsets* candidatos ao qual será calculado o suporte no passo corrente do algoritmo. Além disso, se o item original não está em qualquer um dos *itemsets* candidatos, ele pode ser removido da transação.

Por exemplo, assume-se uma taxonomia em que roupas é pai de roupas leves, que por sua vez é pai de camisetas. Se o *itemset* candidato ao qual será calculado o suporte for {roupas, tênis}, então qualquer transação que contenha camisetas irá substituí-la por roupas e não será necessário manter camisetas na transação e nem adicionar roupas leves a transação, uma vez que ambos os itens (camisetas e roupas leves) não estão contidos no *itemset* candidato {roupas, tênis}.

Pré-calcular os ancestrais Ao invés de calcular os ancestrais de cada item (percorrendo o grafo de taxonomias) para cada execução do algoritmo, pode-se pré-calcular os ancestrais de cada item e utilizá-los durante as execuções do algoritmo.

Remover *itemsets* que contenham apenas um item e seu ancestral A remoção desses *itemsets* é justificada pelos dois *lemas* apresentados a seguir.

Lema 1 *O suporte de um itemset X que contenha um item x e seu ancestral $x \uparrow$ será o mesmo de um itemset $X - x \uparrow$.*

Lema 2 *Se L_k , o conjunto de k -itemsets freqüentes, não inclui qualquer itemset que contenha um item e seu ancestral, o conjunto de itemsets candidatos C_{k+1} , gerado pela função *apriori-gen* da seção 2.2.2, não apresentará qualquer itemset candidato que contenha um item e seu ancestral.*

No primeiro *lema* é enunciado que não é necessário calcular o suporte de qualquer *itemset* que contenha um item e seu ancestral. Essa melhoria pode ser realizada removendo os 2-*itemsets* candidatos (*itemsets* candidatos de tamanho dois) constituídos de um item e seu ancestral. No segundo *lema* é mostrado que remover os

2-*itemsets* candidatos assegura que em passos subseqüentes do algoritmo nunca serão gerados *itemsets* candidatos que contenham um item e seu ancestral. As provas desses lemas são apresentadas em Srikant & Agrawal (1997b).

3.2.3 Algoritmo *Stratify*

Considerando uma ordenação parcial induzida por uma taxonomia sobre um conjunto de *itemsets*, tem-se que os *itemsets* sem pai possuem altura 0. Para os outros *itemsets*, a altura de cada *itemset* X é definida como $\max\{\text{altura}(X \uparrow) \mid X \uparrow \text{ é pai de } X\} + 1$.

O algoritmo *Stratify* apresentado em Srikant & Agrawal (1997a), primeiro calcula o suporte de todos os *itemsets* candidatos C_0 de altura 0 e em seguida remove os *itemsets* candidatos que são descendentes dos *itemsets* em C_0 que não possuem suporte mínimo, calculando em seguida o suporte dos *itemsets* restantes de altura 1. Depois, os *itemsets* candidatos que são descendentes dos *itemsets* em C_1 que não possuem suporte mínimo são removidos, sendo calculado em seguida o suporte dos *itemsets* restantes de altura 2, e assim sucessivamente até que não haja mais *itemsets* para calcular o suporte.

Por exemplo, com os *itemsets* candidatos {roupas, tênis}, {roupas leves, tênis} e {camisetas, tênis}, que são utilizados para o cálculo do suporte e considerando camisetas como filho de roupas leves e roupas leves como filho de roupas. Se {roupas, tênis} não possui suporte maior ou igual ao suporte mínimo, não é necessário calcular o suporte dos *itemsets* candidatos {roupas leves, tênis} e {camisetas, tênis}. Assim, ao invés de calcular o suporte de todos os *itemsets* candidatos de um determinado tamanho em um mesmo passo, como o algoritmo *Cumulate*, o algoritmo *Stratify* pode ser mais rápido calculando primeiro o suporte de {roupas, tênis} e só então calcular o suporte de {roupas leves, tênis} se {roupas, tênis} tem suporte maior ou igual ao suporte mínimo e finalmente calcular o suporte de {camisetas, tênis} se {roupas leves, tênis} também tem suporte maior ou igual ao suporte mínimo.

Porém, o custo extra de realizar vários passos sobre a Base de Dados durante o cálculo dos suportes pode ser maior do que os benefícios. Ao invés de esperar que os *itemsets* candidatos, cujos os itens estão posicionados nos níveis superiores da taxonomia, não tenham suporte mínimo resultando em não ter que calcular o suporte de seus *itemsets* candidatos descendentes, pode-se usar alguns exemplos da Base de Dados para estimar o suporte dos *itemsets* candidatos. Com o uso de exemplos para estimar o suporte, tanto os *itemsets* candidatos que são esperados ter suporte mínimo, quanto os *itemsets* candidatos que não são esperados ter suporte mínimo mas cujos pais têm ou é esperado que tenham suporte mínimo, são utilizados para o cálculo do suporte. Dessa maneira, é esperado que

não se tenha que calcular o suporte de nenhum dos descendentes dos *itemsets* candidatos que são esperados não ter suporte mínimo, mas se um *itemset* candidato que era esperado não ter suporte mínimo passa a ter, um passo extra deve ser realizado para calcular o suporte de seus descendentes. Os algoritmos *Estimate* e *EstMerge*, descritos em Srikant & Agrawal (1997a), utilizam-se de exemplos para estimar o valor do suporte.

Outros algoritmos que podem ser utilizados na geração de Regras de Associação com taxonomias são: Algoritmos da Família ML-T* (Han & Fu 1995) APUD Weber (1998), *Prutax* (Hipp, Myka, Wirth, & Güntzer 1998) e *Genex* (Weber 1998).

3.3 Uso de Taxonomias na Remoção de Regras Não Interessantes

A aplicação de taxonomias em Regras de Associação geralmente conduz a geração de conjuntos com maior número de regras. Por exemplo, pode-se observar que todas as regras na forma $LHS \Rightarrow RHS \uparrow$ serão válidas quando uma regra válida na forma $LHS \Rightarrow RHS$ for encontrada. Para a descoberta de regras com LHS e RHS suficientemente especializados deve-se utilizar os algoritmos de Regras de Associação com um baixo valor para o suporte mínimo (**sup-min**). Como consequência, um grande volume de regras é gerado, tornando a análise das regras por parte do usuário inviável.

Buscando resolver esse problema, nesta seção é apresentada uma solução proposta em Srikant & Agrawal (1997a), que consiste em remover as regras mais especializadas que não possuem comportamento significativamente diferente de suas generalizações.

3.3.1 Medida de Interesse

A aplicação de taxonomias em Regras de Associação permite derivar medidas que podem ser utilizadas para remover regras consideradas não interessantes perante a própria medida. A noção de “interesse”, usando taxonomias, é apresentada a seguir.

Considerando a regra roupas $\Rightarrow$ calçados (suporte=8%, confiança=70%), se roupa é pai de calças, e um quarto das vendas de roupa é de calças, poderia ser esperada a presença da regra calças $\Rightarrow$ calçados (suporte=2%, confiança=70%).

Se o suporte e a confiança da regra calças $\Rightarrow$ calçados são respectivamente, 2% e 70%, a regra poderia ser considerada redundante, uma vez que não apresenta qualquer informação adicional e é menos geral que a primeira regra. A partir dessa noção pode-

se dizer que uma regra é interessante se a mesma possui suporte ou confiança maior ou igual δ vezes o valor de um suporte ou de uma confiança esperada, para algum valor δ especificado pelo usuário.

Essa noção sobre interessabilidade de Regras de Associação aplicando taxonomias foi proposta por Srikant & Agrawal (1997a) e é definida em Adamo (2001) como:

Seja $A = \{a_1, a_2, \dots, a_m\}$ um conjunto de itens, r uma regra válida no formato $LHS \Rightarrow RHS$ e $r \downarrow$ uma especialização válida de r . A regra $r \downarrow$ é considerada como redundante em relação a r (podendo assim ser removida do conjunto de regras válidas) se $r \downarrow$ possui o mesmo comportamento que r . A sentença “ $r \downarrow$ possui o mesmo comportamento de r ” será representada pela medida *desvio*. O *desvio* de $r \downarrow$ com relação a r quando $r \downarrow$ difere de r em apenas um item $a_j \in A$ é definido como:

$$\begin{aligned} desvio(r \downarrow, r) = \max\{ & (sup(r \downarrow)/sup(r))/(sup(a_j \downarrow)/sup(a_j)), \\ & (conf(r \downarrow)/conf(r))/(sup(a_j \downarrow)/sup(a_j)) \}. \end{aligned} \quad (3.1)$$

De maneira intuitiva, a regra $r \downarrow$ possui um comportamento diferente da regra r quando o valor de suporte (*sup*) ou confiança (*conf*) varia de r para $r \downarrow$ mais do que é esperado com relação à variação do suporte de a_j para $a_j \downarrow$. Assim, $r \downarrow$ é considerada uma regra interessante com relação a r quando $desvio(r \downarrow, r) \geq \delta$, sendo que $\delta \geq 1$. O número δ especificado pelo usuário é chamado de interesse mínimo (Srikant & Agrawal 1997a) e especifica qual deve ser o desvio do comportamento da regra $r \downarrow$ para que a mesma possa ser considerada suficientemente diferente da regra r . A medida $desvio(r \downarrow, r)$ pode ser expressa pelas seguintes desigualdades:

$$sup(r \downarrow)/sup(r) \geq \delta \times (sup(a_j \downarrow)/sup(a_j)), \quad (3.2)$$

ou

$$conf(r \downarrow)/conf(r) \geq \delta \times (sup(a_j \downarrow)/sup(a_j)). \quad (3.3)$$

O *desvio* de $r \downarrow$ com relação a r pode ser generalizado para as situações em que $r \downarrow$ difere de r em vários itens, como:

$$desvio(r \downarrow, r) = \max\{sup_desvio(r \downarrow, r), conf_desvio(r \downarrow, r)\}, \quad (3.4)$$

em que

$$sup_desvio(r\downarrow, r) = (sup(r\downarrow)/sup(r)) / \prod_{j=1}^m (sup(a_j\downarrow)/sup(a_j)), \quad (3.5)$$

$$conf_desvio(r\downarrow, r) = (conf(r\downarrow)/conf(r)) / \prod_{j=1}^m (sup(a_j\downarrow)/sup(a_j)). \quad (3.6)$$

Nas situações em que $r\downarrow$ difere de r em vários itens, a medida $desvio(r\downarrow, r)$ pode ser expressa pelas seguintes desigualdades:

$$sup(r\downarrow)/sup(r) \geq \delta \times \prod_{j=1}^m (sup(a_j\downarrow)/sup(a_j)), \quad (3.7)$$

ou

$$conf(r\downarrow)/conf(r) \geq \delta \times \prod_{j=1}^m (sup(a_j\downarrow)/sup(a_j)). \quad (3.8)$$

No Exemplo 7 é mostrado como se utiliza a medida $desvio(r\downarrow, r)$ para encontrar Regras de Associação interessantes.

Exemplo 7 Seja τ a taxonomia apresentada na Figura 3.3 e D uma Base de Dados que contém um conjunto de itens $A = \{\text{calçados}, \text{calças}, \text{sandálias}, \text{tênis}\}$ e um conjunto de transações $T = \{1, 2, 3, 4, 5, 6\}$, no qual a relação de itens comprados por cada transação t_i é apresentada na Tabela 3.2.

Figura 3.3: Uma taxonomia τ para vestuário

Tabela 3.2: Relação de itens comprados por transação

Transações	Itens comprados
1	calçados, calças, sandálias
2	calçados, calças, tênis
3	calças
4	calçados, sandálias
5	calças
6	calçados, sandálias

Para demonstrar o uso da medida $\text{desvio}(r \downarrow, r)$, na tentativa de encontrar regras interessantes, foram calculados os seguintes valores de suporte e confiança a partir da Tabela 3.2:

$$\begin{aligned}
 \text{sup}(\text{calçados}) &= \frac{4}{6}, & \text{sup}(\text{tênis}) &= \frac{1}{6}, & \text{sup}(\text{sandálias}) &= \frac{3}{6}, \\
 \text{sup}(\text{calçados} \Rightarrow \text{calças}) &= \frac{2}{6}, & \text{sup}(\text{tênis} \Rightarrow \text{calças}) &= \frac{1}{6}, & \text{sup}(\text{sandálias} \Rightarrow \text{calças}) &= \frac{1}{6}, \\
 \text{conf}(\text{calçados} \Rightarrow \text{calças}) &= \frac{2}{4}, & \text{conf}(\text{tênis} \Rightarrow \text{calças}) &= 1, & \text{conf}(\text{sandálias} \Rightarrow \text{calças}) &= \frac{1}{3}.
 \end{aligned}$$

Com os valores de suporte e confiança calculados é possível verificar que

$$\begin{aligned}
 \left(\frac{\text{sup}(\text{tênis} \Rightarrow \text{calças})}{\text{sup}(\text{calçados} \Rightarrow \text{calças})} \right) &\geq \left(2 \times \left(\frac{\text{sup}(\text{tênis})}{\text{sup}(\text{calçados})} \right) \right) \Rightarrow \frac{1}{2} \geq \left(2 \times \frac{1}{4} \right) \Rightarrow \frac{1}{2} \geq \frac{1}{2}, \\
 \left(\frac{\text{conf}(\text{tênis} \Rightarrow \text{calças})}{\text{conf}(\text{calçados} \Rightarrow \text{calças})} \right) &\geq \left(2 \times \left(\frac{\text{sup}(\text{tênis})}{\text{sup}(\text{calçados})} \right) \right) \Rightarrow 2 \geq \left(2 \times \frac{1}{4} \right) \Rightarrow 2 \geq \frac{1}{2}, \\
 \left(\frac{\text{sup}(\text{sandálias} \Rightarrow \text{calças})}{\text{sup}(\text{calçados} \Rightarrow \text{calças})} \right) &< \left(\frac{\text{sup}(\text{sandálias})}{\text{sup}(\text{calçados})} \right) \Rightarrow \frac{1}{2} < \frac{3}{4}, \\
 \left(\frac{\text{conf}(\text{sandálias} \Rightarrow \text{calças})}{\text{conf}(\text{calçados} \Rightarrow \text{calças})} \right) &< \left(\frac{\text{sup}(\text{sandálias})}{\text{sup}(\text{calçados})} \right) \Rightarrow \frac{2}{3} < \frac{3}{4},
 \end{aligned}$$

e obter as seguintes conclusões:

- Nas duas primeiras linhas observa-se que a regra $\text{tênis} \Rightarrow \text{calças}$ é 2 vezes mais interessante (2-interessante) que a regra $\text{calçados} \Rightarrow \text{calças}$;
- Já nas duas últimas linhas não há um $\delta \geq 1$ na qual $\text{sandálias} \Rightarrow \text{calças}$ seria δ -interessante com relação a regra $\text{calçados} \Rightarrow \text{calças}$.

Além dessa medida, em Adamo (2001) é proposto um outro método para remover regras não interessantes, denominado Poda Baseada na Presença do Item, o qual afirma

que em certas circunstâncias é possível saber antecipadamente que a presença de certos itens em uma regra assegura que a mesma não pode ser δ -interessante, podendo assim, ser removida.

3.3.2 Algoritmo para Remover Regras Não Interessantes

No Algoritmo 4 é utilizada a medida de interesse *desvio*, apresentada na seção 3.3.1, que tenta encontrar regras interessantes. Esse algoritmo considera uma regra r e um interesse mínimo δ , sendo que r é dita ser δ -interessante:

1. se r é uma generalização máxima ou;
2. se para todas as generalizações $r \uparrow$ de r , $r \uparrow$ é δ -interessante, então, r é δ -interessante com relação a $r \uparrow$.

O algoritmo identifica todas as regras interessantes em um conjunto de regras válidas, começando pela generalização máxima. O conjunto de regras válidas é parcialmente ordenado pela relação de generalização, podendo ser representado por um grafo. Em seguida, o algoritmo é executado a partir do vértice de generalização máxima, verificando os próximos vértices até alcançar os vértices de especialização máxima.

Algoritmo 4 Remove Regras Não Interessantes. Fonte: (Adamo 2001)

Require: Um grafo representando um conjunto de regras parcialmente ordenado pela relação de generalização.

```

1: Inicialize regras_interessantes := {};
2: for all camada de generalização, iniciando a partir da mais geral do
3:   for all regra válida  $r$  presente na camada de generalização corrente do
4:     if  $r$  pertence a camada de generalização mais geral then
5:       regras_interessantes := regras_interessantes  $\cup$   $\{r\}$ ;
6:     else
7:       bool regra_interessante := true;
8:       while regra_interessante = true AND houver generalizações  $r \uparrow$  de  $r$ , tal que  $r \uparrow$  é
           $\delta$ -interessante do
9:         if desvio( $r, r \uparrow$ ) <  $\delta$  then
10:          regra_interessante := false;
11:        end if
12:      end while
13:    end if
14:    if regra_interessante = true then
15:      regras_interessantes := regras_interessantes  $\cup$   $\{r\}$ ;
16:    end if
17:  end for
18: end for

```

Em Adamo (2001) é proposto um algoritmo melhorado que considera um subconjunto limitado de generalizações de r , que é suficiente para concluir se uma regra é interessante ou não.

3.4 Considerações Finais

No processo de Extração de Conhecimento utilizando-se Regras de Associação, um dos problemas é o grande volume de conhecimento (no formato de regras) que é gerado, dificultando a análise e a compreensão do conhecimento pelo usuário.

Relacionado a esse problema, neste capítulo foi descrito como o uso de taxonomias em Regras de Associação pode reduzir o volume e/ou melhorar a compreensão das regras geradas. Entretanto, a técnica apresentada neste capítulo possui o inconveniente de gerar um volume de regras muito maior do que o volume de regras geradas pela técnica de Regras de Associação que não utiliza taxonomias, antes de reduzir esse volume de regras geradas.

A técnica descrita, primeiro gera todas as Regras de Associação possíveis (regras com e sem taxonomias). Depois de gerar as regras, a técnica busca reduzir o volume de regras geradas utilizando a medida de interesse (*desvio*), apresentada na Seção 3.3.1, para remover as regras que não são interessantes. Além disso, a escolha do valor da medida de interesse é realizada de maneira subjetiva (de acordo com a experiência do usuário), o que não garante que haja uma grande redução do volume de regras.

Em virtude dos problemas apresentados, no próximo capítulo é proposto um algoritmo que busca generalizar Regras de Associação na etapa de Pós-Processamento do conhecimento. As regras são mineradas com algoritmos de Regras de Associação que não utilizam taxonomias e, em um segundo momento, o volume de regras geradas é reduzido (generalizado) utilizando algumas taxonomias.

Algoritmo Proposto para Generalização de Regras de Associação

Como apresentado no Capítulo 3, o uso de taxonomias em Regras de Associação proposto por Srikant & Agrawal (1997a) possui o inconveniente de gerar um volume de regras muito maior do que o gerado pela técnica que não utiliza taxonomias. A técnica descrita por Srikant & Agrawal (1997a) inicialmente gera todas as Regras de Associação possíveis (com e sem taxonomias). Em seguida, na tentativa de reduzir o volume de regras gerado é utilizada a medida *desvio* (apresentada na Seção 3.3.1 do Capítulo 3) para remover as regras que não são interessantes. Como a escolha do valor da medida é realizada de maneira subjetiva (de acordo com o conhecimento do usuário), a mesma não garante que irá ocorrer uma grande redução do volume de regras geradas.

Devido aos problemas apresentados, neste capítulo é proposto um processo e, por consequência, um algoritmo que busca generalizar Regras de Associação sem aumentar o volume de regras gerado e sem a necessidade de utilizar uma medida subjetiva para remover as regras. Na Seção 4.1 é descrito o processo de generalização de Regras de Associação proposto. Na Seção 4.2 é apresentado o algoritmo *GART* que formaliza esse processo. Já na Seção 4.3 será apresentada a biblioteca de classes que implementa o algoritmo *GART*, sendo relatada a funcionalidade dos métodos implementados, os arquivos de entrada e de saída e a interface para uso do algoritmo. Por fim, na Seção 4.4 são feitas algumas

considerações finais do capítulo.

4.1 Proposta de um Processo para Generalizar Regras de Associação

No Capítulo 3 foi apresentado um método que utiliza taxonomias durante a mineração das regras buscando generalizá-las. Uma outra maneira de generalizar as Regras de Associação consiste no uso de taxonomias durante o seu Pós-processamento.

Analisando a estrutura das Regras de Associação, geradas por algoritmos que não utilizam taxonomias, é possível verificar que as mesmas podem ser generalizadas utilizando taxonomias. Esse fato é ilustrado nas Figuras 4.1 e 4.2.

Figura 4.1: Generalização de Regras de Associação usando uma taxonomia

Na Figura 4.1 é ilustrada a generalização das regras

$$\begin{aligned} &\text{camiseta} \Rightarrow \text{boné e} \\ &\text{bermuda} \Rightarrow \text{boné,} \end{aligned}$$

utilizando a taxonomia roupas leves (Taxonomia τ_1 na Figura 4.1) para generalizar os itens camiseta e bermuda e gerar a Regra de Associação generalizada

$$\text{roupas leves} \Rightarrow \text{boné.}$$

Durante a generalização, há a substituição dos itens camiseta e bermuda pelo item roupas leves gerando duas regras iguais $\text{roupas leves} \Rightarrow \text{boné}$. Em seguida, uma das regras é removida permanecendo apenas uma regra generalizada $\text{roupas leves} \Rightarrow \text{boné}$. De modo

semelhante, na Figura 4.2 é ilustrada a generalização de quatro Regras de Associação aplicando duas taxonomias.

Figura 4.2: Generalização de Regras de Associação usando duas taxonomias

Inicialmente, os itens camiseta e bermuda das regras

camiseta & chinelo $\Rightarrow$ boné,
 camiseta & sandália $\Rightarrow$ boné,
 bermuda & sandália $\Rightarrow$ boné e
 bermuda & chinelo $\Rightarrow$ boné,

são substituídos pelo item roupas leves (que representa uma generalização) gerando duas regras roupas leves & chinelo $\Rightarrow$ boné e duas regras roupas leves & sandália $\Rightarrow$ boné. Em seguida, as regras repetidas são removidas permanecendo apenas as regras

roupas leves & chinelo $\Rightarrow$ boné e
 roupas leves & sandália $\Rightarrow$ boné.

As duas regras resultantes do passo 1 (Figura 4.2), são novamente generalizadas, sendo os itens chinelo e sandália substituídos pelo item calçados abertos (que representa outra generalização), gerando duas regras roupas leves & calçados abertos $\Rightarrow$ boné. As regras repetidas são removidas permanecendo apenas a Regra de Associação generalizada

roupas leves & calçados abertos $\Rightarrow$ boné.

O comportamento ilustrado nas Figuras 4.1 e 4.2 foi identificado em diversos conjuntos de Regras de Associação analisados. A partir dessa análise foi proposto um processo para generalizar Regras de Associação que é apresentado na Figura 4.3.

Figura 4.3: Processo proposto para generalização de Regras de Associação

O processo proposto neste trabalho generaliza apenas um dos lados (LHS ou RHS) das Regras de Associação. Inicialmente as regras (representadas na sintaxe padrão definida por Melanda & Rezende (2003)) são agrupadas em subconjuntos que apresentam antecedente ou conseqüente semelhante. Se o processo for utilizado para generalizar o lado esquerdo das regras (LHS), os subconjuntos são gerados utilizando conseqüentes (RHS) semelhantes e se o processo for utilizado para generalizar o lado direito das regras (RHS), os subconjuntos são gerados utilizando antecedentes (LHS) semelhantes. Na Figura 4.3 é ilustrado o processo de generalização do lado esquerdo das regras, por conseqüência, os subconjuntos são agrupados utilizando semelhanças no lado direito das regras. Em seguida são utilizadas as taxonomias para generalizar cada subconjunto, e ao final da generalização, os itens de cada regra são ordenados lexicograficamente e a regra é armazenada em um conjunto de Regras de Associação generalizadas.

Após esse processo, deve-se calcular a Tabela de Contingência para cada Regra de Associação que foi generalizada, uma vez que a sintaxe padrão proposta por Melanda & Rezende (2003) inclui a regra acompanhada de sua Tabela de Contingência. A Tabela de Contingência de uma regra representa a sua cobertura com relação a Base de Dados que foi utilizada na mineração da regra. Além disso, os valores da tabela podem ser utilizados para calcular diversas medidas de avaliação de regras, entre as quais pode-se citar o *framework* de medidas de avaliação proposto por Lavrač, Flach, & Zupan (1999). O projeto *Discover* (descrito na Seção 5.1 do Capítulo 5), ao qual este projeto faz parte, possui métodos que calculam diversas medidas de avaliação de Regras de Associação desde que as regras estejam na sintaxe padrão proposta por Melanda & Rezende (2003).

Com o cálculo da Tabela de Contingência encerra-se o processo de generalização. Na próxima seção é apresentado o algoritmo $\mathcal{GART}$. Esse algoritmo é responsável pela generalização de Regras de Associação proposta nesta seção.

4.2 O Algoritmo $\mathcal{GART}$

Nesta seção é descrito o algoritmo $\mathcal{GART}$ (*Generalization of Association Rules using Taxonomies* – Generalização de Regras de Associação usando Taxonomias), que é responsável pelo processo de generalização de Regras de Associação proposto na Seção 4.1. O algoritmo $\mathcal{GART}$, ilustrado no Algoritmo 5, recebe como entrada um conjunto de Regras de Associação R , um conjunto de Taxonomias $\mathcal{T}$, uma Base de Dados D e a definição do *lado* da regra a ser generalizado – lado esquerdo (antecedente da regra) ou lado direito (conseqüente da regra). A notação $\overline{lado}$ define o lado das regras que não será generalizado.

Algoritmo 5 *GART*

Require: Um conjunto de Regras de Associação R , um conjunto de Taxonomias $\mathcal{T}$, uma Base de Dados D e a definição do *lado* da regra a ser generalizado – lado esquerdo (antecedente da regra) ou lado direito (consequente da regra).

```
1:  $Rg := \emptyset$ ;  
2:  $\hat{E} := \text{gera-subconjuntos}(R, \overline{\text{lado}})$ ;  
3: for all subconjunto  $E \subseteq \hat{E}$  do  
4:    $\text{generaliza-regras}(E, \mathcal{T}, \text{lado})$ ;  
5:    $\text{ordena-lexicograficamente}(E, \text{lado})$ ;  
6:    $Rg := Rg \cup E$ ;  
7: end for  
8: for all regra  $r \in Rg$  do  
9:   if  $r$  é uma regra generalizada then  
10:     $\text{calcula-TC}(r, \mathcal{T}, D)$ ;  
11:   end if  
12: end for  
13: return  $Rg$ ;
```

Inicialmente o algoritmo inicializa com “ $\emptyset$ ” (vazio) o conjunto Rg que irá armazenar as Regras de Associação generalizadas. Em seguida são gerados os subconjuntos de regras que possuem antecedente ou consequente semelhante. Se o algoritmo for utilizado para generalizar o lado esquerdo das regras, os subconjuntos são gerados utilizando consequentes semelhantes e se o algoritmo for utilizado para generalizar o lado direito das regras, os subconjuntos são gerados utilizando antecedentes semelhantes. A geração dos subconjuntos é realizada com a função gera-subconjuntos (linha 2 do Algoritmo 5). Nessa função, o parâmetro $\overline{\text{lado}}$ indica que os subconjuntos são gerados utilizando o lado das regras que não será generalizado. A função gera-subconjuntos é descrita detalhadamente em Domingues & Rezende (2004).

O próximo passo do algoritmo (linhas 3 a 7 do Algoritmo 5) consiste na generalização das Regras de Associação contidas em cada um dos subconjuntos de regras $E \subseteq \hat{E}$, na ordenação lexicográfica dos itens de cada regra pertencente aos subconjuntos de regras e no armazenamento das regras no conjunto Rg . A generalização é realizada utilizando a função generaliza-regras , descrita na Seção 4.2.1. A ordenação lexicográfica dos itens das regras é realizada utilizando a função $\text{ordena-lexicograficamente}$, que é descrita em Domingues & Rezende (2004).

Após a generalização é calculada a Tabela de Contingência para cada Regra de Associação generalizada contida no conjunto Rg (linhas 8 a 12 do Algoritmo 5). O cálculo da Tabela de Contingência é realizado utilizando a função calcula-TC , descrita na Seção 4.2.4. Por fim, o algoritmo (na linha 13) retorna o conjunto de Regras de Associação generalizadas Rg .

Nas próximas seções são descritas as principais funções utilizadas pelo algoritmo *GART*.

4.2.1 Função generaliza-regras

A função generaliza-regras é apresentada no Algoritmo 6. Essa função recebe como entrada um conjunto de Regras de Associação R (com antecedentes ou conseqüentes semelhantes – dependendo do lado a ser generalizado), um conjunto de Taxonomias $\mathcal{T}$ e a definição do *lado* da regra a ser generalizado – lado esquerdo (antecedente da regra) ou lado direito (conseqüente da regra).

Algoritmo 6 Função generaliza-regras

Require: Um conjunto de Regras de Associação R (com antecedentes ou conseqüentes semelhantes – dependendo do lado a ser generalizado), um conjunto de Taxonomias $\mathcal{T}$ e a definição do *lado* da regra a ser generalizado – lado esquerdo (antecedente da regra) ou lado direito (conseqüente da regra).

```

1: for all taxonomia  $\tau \in \mathcal{T}$  do
2:   for all regra  $r \in R$  do
3:     generaliza-uma-regra( $r, \tau, lado$ );
4:   end for
5:    $Rgt := \emptyset$ ;
6:    $\widehat{Et} := \text{gera-subconjuntos}(R, lado)$ ;
7:   for all subconjunto  $Et \subseteq \widehat{Et}$  do
8:     if número-regras( $Et$ ) = número-itens-especialização-máxima( $\tau$ ) then
9:       remove-regras-repetidas( $Et$ );
10:    else
11:      desfaz-última-generalização( $Et, \tau, lado$ );
12:    end if
13:     $Rgt := Rgt \cup Et$ ;
14:  end for
15:   $R := Rgt$ ;
16: end for
```

Na função generaliza-regras, a generalização das Regras de Associação é realizada considerando cada uma das taxonomias $\tau \in \mathcal{T}$. A generalização de cada regra $r \in R$ para uma determinada taxonomia $\tau \in \mathcal{T}$ (linhas 2 a 4 do Algoritmo 6) é realizada utilizando a função generaliza-uma-regra, descrita na Seção 4.2.2. Em seguida, a função generaliza-regras inicializa com “ $\emptyset$ ” (vazio) o conjunto Rgt , que irá armazenar temporariamente as Regras de Associação generalizadas (linha 5), e gera subconjuntos de regras que apresentem o *lado* da generalização (antecedente ou conseqüente) semelhantes (linha 6). O lado da regra que não está sendo generalizado ($\overline{lado}$) já apresenta itens semelhantes, devido a execução da função gera-subconjuntos no Algoritmo 5.

Um dos motivos para gerar os subconjuntos com regras generalizadas semelhantes é facilitar a remoção das regras que são repetidas. Outro motivo é a verificação da igualdade do número de regras em cada subconjunto $Et \subseteq \widehat{Et}$ e do número de itens que são especializações máximas na taxonomia. Essa verificação valida uma generalização e evita que a mesma induza um conhecimento que não existe. Por exemplo, na Figura 4.1 na página 50 é possível verificar que o número de itens que são especializações máximas na taxonomia (2 itens) é igual ao número de regras generalizadas (2 regras considerando as repetidas). Assim, a generalização é válida e as Regras de Associação generalizadas repetidas são removidas.

Uma generalização não é válida se o número de regras em um subconjunto for menor do que o número de itens que são especializações máximas na taxonomia. Por exemplo, se na Figura 4.1 na página 50 existisse apenas a regra **camiseta** $\Rightarrow$ **boné** para ser generalizada, e caso houvesse a generalização dessa regra, a mesma seria inválida pois haveria uma regra generalizada e dois itens que são considerados especializações máximas na taxonomia. Assim, a regra generalizada **roupas leves** $\Rightarrow$ **boné** estaria representando (pela presença do item **roupas leves**) a generalização das regras **camiseta** $\Rightarrow$ **boné** e **bermuda** $\Rightarrow$ **boné**, quando na verdade foi realizada apenas a generalização da regra **camiseta** $\Rightarrow$ **boné**, ocorrendo, dessa maneira, a indução de um conhecimento que não existe. Cabe ressaltar que em nenhum momento o número de regras armazenadas em um subconjunto poderá ser maior do que o número de itens que são especializações máximas em uma taxonomia, uma vez que isso implicaria na presença de Regras de Associação repetidas no subconjunto.

Na linha 8 do Algoritmo 6 é realizada a comparação do número de regras em cada subconjunto $Et \subseteq \widehat{Et}$ e do número de itens que são especializações máximas na taxonomia. Se o número de regras e o número de itens são iguais, as Regras de Associação que foram generalizadas são consideradas válidas e as regras repetidas são removidas (linha 9 do Algoritmo 6) utilizando a função *remove-regras-repetidas*, descrita em Domingues & Rezende (2004). Porém, se o número de regras e o número de itens não são iguais, as Regras de Associação que foram generalizadas são consideradas inválidas e a generalização é desfeita (linha 11 do Algoritmo 6) utilizando a função *desfaz-última-generalização*, descrita na Seção 4.2.3. Em seguida, as Regras de Associação que foram ou não generalizadas são armazenadas temporariamente no conjunto de regras *Rgt*. Por fim, o conjunto de regras *Rgt* é atribuído ao conjunto *R* (linha 15 do Algoritmo 6) para que o mesmo possa ser generalizado novamente utilizando uma outra taxonomia $\tau \in \mathcal{T}$.

4.2.2 Função generaliza-uma-regra

A função generaliza-uma-regra é apresentada no Algoritmo 7. Essa função recebe como entrada uma Regra de Associação r , uma Taxonomia $\mathcal{T}$ e a definição do *lado* da regra a ser generalizado – lado esquerdo (antecedente da regra) ou lado direito (conseqüente da regra).

Algoritmo 7 Função generaliza-uma-regra

Require: Uma Regra de Associação r , uma Taxonomia $\mathcal{T}$ e a definição do *lado* da regra a ser generalizado – lado esquerdo (antecedente da regra) ou lado direito (conseqüente da regra).

```
1: if  $lado = esquerda$  then  
2:   if apenas um item  $a$  do antecedente da regra  $r$  é generalizado por  $\mathcal{T}$  then  
3:      $r.última-generalização := a$ ;  
4:      $a := \mathcal{T}.item-generalização-máxima$ ;  
5:   end if  
6: else  
7:   if apenas um item  $a$  do conseqüente da regra  $r$  é generalizado por  $\mathcal{T}$  then  
8:      $r.última-generalização := a$ ;  
9:      $a := \mathcal{T}.item-generalização-máxima$ ;  
10:  end if  
11: end if
```

Inicialmente a função generaliza-uma-regra verifica qual é o lado da regra a ser generalizado. Se o lado da regra a ser generalizado for o esquerdo (antecedente da regra), as linhas 2 a 5 do Algoritmo 7 são executadas. Na linha 2 é verificado se apenas um item a da regra r é generalizado pela taxonomia $\mathcal{T}$, ou seja, se apenas um item a no lado da generalização é igual a um dos itens que são especializações máximas na taxonomia. Se a verificação for verdadeira, o item da regra é armazenado em $r.última-generalização$ (linha 3) para ser usado futuramente caso haja a necessidade de desfazer a generalização. Em seguida, o item da regra é substituído (generalizado) pelo item de generalização máxima da taxonomia (linha 4 do Algoritmo 7).

Caso o lado da regra a ser generalizado seja o lado direito (conseqüente da regra) são executadas as linhas 7 a 10 do Algoritmo 7. Nessas linhas são executadas as mesmas tarefas das linhas 2 a 5, porém, para o lado direito das regras.

É importante ressaltar que as verificações ocorridas nas linhas 2 e 7 são de grande importância para a generalização das Regras de Associação, pois as mesmas impedem que uma regra generalizada apresente, entre seus itens, duas ou mais generalizações iguais.

4.2.3 Função desfaz-última-generalização

No Algoritmo 8 é apresentada a função que desfaz a última generalização de um conjunto de regras R . A função desfaz-última-generalização recebe como entrada um conjunto de Regras de Associação R , uma taxonomia $\mathcal{T}$ e a definição do *lado* da regra a ser generalizado – esquerdo (antecedente da regra) ou direito (conseqüente da regra).

Algoritmo 8 Função desfaz-última-generalização

Require: Um conjunto de Regras de Associação R , uma taxonomia $\mathcal{T}$ e a definição do *lado* da regra a ser generalizado – lado esquerdo (antecedente da regra) ou lado direito (conseqüente da regra).

```
1: if lado = esquerdo then
2:   for all regra  $r \in R$  do
3:     if há algum item  $a$  no antecedente da regra  $r$  que foi generalizado por  $\mathcal{T}$  then
4:        $a := r.$ última-generalização;
5:     end if
6:   end for
7: else
8:   for all regra  $r \in R$  do
9:     if há algum item  $a$  no conseqüente da regra  $r$  que foi generalizado por  $\mathcal{T}$  then
10:       $a := r.$ última-generalização;
11:    end if
12:  end for
13: end if
```

A função desfaz-última-generalização é executada quando uma generalização é considerada inválida, ou seja, quando o número de regras em um subconjunto for menor do que o número de itens que são especializações máximas na taxonomia $\mathcal{T}$. A sua execução evita que o processo de generalização de Regras de Associação induza um conhecimento que não exista, como foi exemplificado na Seção 4.2.1.

4.2.4 Função calcula-TC

A Tabela de Contingência para uma Regra de Associação r , no formato $LHS \Rightarrow RHS$, representa a sua cobertura com relação a Base de Dados que foi utilizada para minerar essa regra. Na Tabela 4.1, LHS denota o antecedente da regra e $\overline{LHS}$ o complemento de LHS . O mesmo conceito se aplica para o conseqüente RHS e $\overline{RHS}$. $n(LHS)$ denota o número de transações no qual LHS é verdadeiro e $n(\overline{LHS})$ denota o número de transações no qual LHS é falso. $n(LHS \text{ } RHS)$ denota o número de transações no qual LHS e RHS são verdadeiros, $n(\overline{LHS} \text{ } RHS)$ denota o número de transações no qual $\overline{LHS}$ é falso e RHS é verdadeiro, de maneira similar para os demais elementos e N denota o número

total de transações.

Tabela 4.1: Tabela de Contingência. Fonte:(Lavrač, Flach, & Zupan 1999)

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

O cálculo da Tabela de Contingência para as Regras de Associação generalizadas (no processo proposto) é realizado pela função calcula-TC apresentada no Algoritmo 9. A função recebe como entrada uma Regra de Associação r , um conjunto de Taxonomias $\mathcal{T}$ e uma Base de Dados D .

Algoritmo 9 Função calcula-TC

Require: Uma Regra de Associação r , um conjunto de Taxonomias $\mathcal{T}$ e uma Base de Dados D .

```

1: Inicializa todos os valores da Tabela de Contingência com 0;
2: Recupera os itens de  $r$  que foram generalizados utilizando  $\mathcal{T}$ ;
3: for all transação da Base de Dados  $D$  do
4:   incrementa o número de transações  $N$ ;
5:   if todos os itens do antecedente da regra  $r$  são cobertos pela transação then
6:      $antecedente := \mathbf{true}$ ;
7:     incrementa o valor de  $n(LHS)$  na Tabela de Contingência;
8:   else
9:      $antecedente := \mathbf{false}$ ;
10:    incrementa o valor de  $n(\overline{LHS})$  na Tabela de Contingência;
11:   end if
12:   if todos os itens do conseqüente da regra  $r$  são cobertos pela transação then
13:      $consequente := \mathbf{true}$ ;
14:     incrementa o valor de  $n(RHS)$  na Tabela de Contingência;
15:   else
16:      $consequente := \mathbf{false}$ ;
17:     incrementa o valor de  $n(\overline{RHS})$  na Tabela de Contingência;
18:   end if
19:   if ( $antecedente = \mathbf{true}$ ) and ( $consequente = \mathbf{true}$ ) then
20:     incrementa o valor de  $n(LHS\ RHS)$  na Tabela de Contingência;
21:   else if ( $antecedente = \mathbf{true}$ ) and ( $consequente = \mathbf{false}$ ) then
22:     incrementa o valor de  $n(LHS\ \overline{RHS})$  na Tabela de Contingência;
23:   else if ( $antecedente = \mathbf{false}$ ) and ( $consequente = \mathbf{true}$ ) then
24:     incrementa o valor de  $n(\overline{LHS}\ RHS)$  na Tabela de Contingência;
25:   else
26:     incrementa o valor de  $n(\overline{LHS}\ \overline{RHS})$  na Tabela de Contingência;
27:   end if
28: end for

```

Os valores da Tabela de Contingência são inicializados com “0” (zero). Na linha 2 do

Algoritmo 9, os itens que foram generalizados na Regra de Associação são recuperados para que o cálculo dos valores da Tabela de Contingência possa ser realizado. A recuperação desses itens é necessária para a realização da comparação entre os itens da regra e os da Base de Dados, uma vez que os itens da base não estão generalizados.

No Algoritmo 9, o laço (*loop*) representado pelas linhas 3 a 28 permite verificar a cobertura de uma regra r para todas as transações pertencentes a uma Base de Dados D . Nas linhas 5 a 11 são verificadas a cobertura do antecedente (LHS) de uma Regra de Associação r e calculados os valores de $n(LHS)$ (se o antecedente da regra cobre a transação) e $n(\overline{LHS})$ (se o antecedente da regra não cobre a transação) para a Tabela de Contingência. Já nas linhas 12 a 18 são verificadas a cobertura do conseqüente (RHS) de uma Regra de Associação r e calculados os valores de $n(RHS)$ (se o conseqüente da regra cobre a transação) e $n(\overline{RHS})$ (se o conseqüente da regra não cobre a transação). Por fim, nas linhas 19 a 27 do Algoritmo 9 são calculados os valores $n(LHS\ RHS)$, $n(LHS\ \overline{RHS})$, $n(\overline{LHS}\ RHS)$ e $n(\overline{LHS}\ \overline{RHS})$. O valor do número de transações N é calculado na linha 4 do Algoritmo 9.

Cabe ressaltar que na implementação do algoritmo $\mathcal{GART}$, realizada neste trabalho, os valores da Tabela de Contingência são frequências relativas, ou seja, os valores da tabela são divididos por N .

No Exemplo 8, páginas 60 a 65, é ilustrado o funcionamento do algoritmo $\mathcal{GART}$ aplicado a um pequeno conjunto de Regras de Associação.

Exemplo 8 *Seja D uma Base de Dados que contém um conjunto de itens $A = \{\text{bermuda, boné, camiseta, chinelo, sandália, short, tênis}\}$ e um conjunto de transações $T = \{1, 2, 3, 4, 5, 6\}$, no qual a relação de itens comprados por cada transação t_i é apresentada na Tabela 4.2.*

Tabela 4.2: Exemplo 8 – Relação de itens comprados por transação

Transações	Itens comprados
1	camiseta, chinelo, boné, tênis
2	camiseta, sandália, boné, tênis
3	bermuda, chinelo, boné, tênis
4	bermuda, sandália, boné, tênis
5	sandália, chinelo, short, tênis
6	sandália, camiseta, short, tênis

A Base de Dados D foi submetida ao algoritmo Apriori¹ (descrito na Seção 2.2.2) com

¹Uma implementação gratuita do algoritmo Apriori pode ser obtida no site <http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html>

o valor do suporte mínimo (**sup-min**) igual a 10% e da confiança mínima (**conf-min**) igual a 50%. O algoritmo gerou 101 Regras de Associação, das quais foram selecionadas 8 regras para compor o conjunto R , apresentado na Tabela 4.3. Nessa tabela são apresentados apenas os valores de $n(LHS \rightarrow RHS)$, $n(LHS \rightarrow \overline{RHS})$, $n(\overline{LHS} \rightarrow RHS)$ e $n(\overline{LHS} \rightarrow \overline{RHS})$ da Tabela de Contingência de cada regra, uma vez que a partir desses valores é possível obter os demais valores da Tabela de Contingência.

Tabela 4.3: Exemplo 8 – Conjunto de Regras de Associação R

Regras	$n(LHS \rightarrow RHS)$	$n(LHS \rightarrow \overline{RHS})$	$n(\overline{LHS} \rightarrow RHS)$	$n(\overline{LHS} \rightarrow \overline{RHS})$
bermuda $\Rightarrow$ tênis	2	0	0	4
camiseta $\Rightarrow$ tênis	3	0	0	3
bermuda & chinelo $\Rightarrow$ boné	1	0	2	3
bermuda & sandália $\Rightarrow$ boné	1	0	2	3
camiseta & sandália $\Rightarrow$ boné	1	1	1	3
camiseta & chinelo $\Rightarrow$ boné	1	0	2	3
chinelo & sandália $\Rightarrow$ short	1	0	4	1
camiseta & sandália $\Rightarrow$ short	1	1	3	1

O conjunto de Taxonomias $\mathcal{T}$ que será fornecido ao algoritmo $\mathcal{GART}$ para generalizar o conjunto de Regras de Associação R é apresentado na Figura 4.4.

Figura 4.4: Exemplo 8 – Conjunto de taxonomias $\mathcal{T}$

Neste exemplo, o algoritmo irá tentar generalizar o antecedente das regras (lado = esquerdo). A sequência de passos ilustrando o seu funcionamento é descrito a seguir.

Inicialmente a função gera-subconjuntos do algoritmo $\mathcal{GART}$ (linha 2 do Algoritmo 5) é executada gerando os subconjuntos de regras que apresentam conseqüentes semelhantes. Os subconjuntos gerados são apresentados na Tabela 4.4. Em seguida, as linhas 3 a 7 do Algoritmo 5 são executadas na tentativa de generalizar cada subconjunto de regras.

Tabela 4.4: Exemplo 8 – Subconjuntos de Regras de Associação com conseqüentes semelhantes

Subconjuntos	Regras
Subconjunto E_1	bermuda $\Rightarrow$ tênis camiseta $\Rightarrow$ tênis
Subconjunto E_2	bermuda & chinelo $\Rightarrow$ boné bermuda & sandália $\Rightarrow$ boné camiseta & sandália $\Rightarrow$ boné camiseta & chinelo $\Rightarrow$ boné
Subconjunto E_3	chinelo & sandália $\Rightarrow$ short camiseta & sandália $\Rightarrow$ short

Tentativa de generalização do subconjunto E_1

Na tentativa de generalizar cada uma das regras armazenadas no subconjunto E_1 , a função *generaliza-regras* (linha 4 do Algoritmo 5) e, por conseqüência, a função *generaliza-uma-regra* (linha 3 do Algoritmo 6) são executadas. Inicialmente na função *generaliza-uma-regra* é verificado que ambas as regras **bermuda $\Rightarrow$ tênis** e **camiseta $\Rightarrow$ tênis** possuem apenas um item no antecedente que pode ser generalizado utilizando a taxonomia **roupas leves** – que generaliza os itens **camiseta** e **bermuda** e é representada pela Taxonomia τ_1 na Figura 4.4. Assim, é armazenado em *r.última-generalização* de cada regra o item a ser generalizado, sendo esse item substituído pelo de generalização máxima da Taxonomia τ_1 , gerando duas regras **roupas leves $\Rightarrow$ tênis**. Em seguida, dentro da função *generaliza-regras* é executada a função *gera-subconjuntos* (linha 6 do Algoritmo 6) que gera um subconjunto contendo as duas regras **roupas leves $\Rightarrow$ tênis**. Nas linhas 7 a 14 do Algoritmo 6 é verificado que o número de regras no subconjunto gerado (2 regras) é igual o número de itens que são especializações máximas na Taxonomia τ_1 (2 itens). A verificação valida a generalização e as regras repetidas são removidas (linhas 8 e 9 do Algoritmo 6).

A regra resultante **roupas leves $\Rightarrow$ tênis** é armazenada temporariamente no conjunto de regras generalizadas *Rgt* que é atribuído ao conjunto *R* na tentativa de generalizá-lo, novamente, utilizando a taxonomia **calçados abertos** – que generaliza os itens **sandália** e **chinelo** e é representada pela Taxonomia τ_2 na Figura 4.4. O processo descrito é executado novamente, sendo que, dessa vez, nenhum item da regra **roupas leves $\Rightarrow$ tênis** é generalizado pela Taxonomia τ_2 .

O conjunto de regras *R* contendo apenas a regra **roupas leves $\Rightarrow$ tênis** é retornado ao algoritmo *GART*, onde os itens da regra são ordenados lexicograficamente e a regra é armazenada no conjunto *Rg*. As linhas 3 a 7 do Algoritmo 5 são executadas novamente na tentativa de generalizar o subconjunto de regras E_2 .

Tentativa de generalização do subconjunto E_2

Na tentativa de generalizar cada uma das regras armazenadas no subconjunto E_2 , a função generaliza-regras (linha 4 do Algoritmo 5) e, por consequência, a função generaliza-uma-regra (linha 3 do Algoritmo 6) são executadas novamente. Inicialmente na função generaliza-uma-regra é verificado que cada uma das quatro regras:

bermuda & chinelo $\Rightarrow$ boné,
bermuda & sandália $\Rightarrow$ boné,
camiseta & sandália $\Rightarrow$ boné e
camiseta & chinelo $\Rightarrow$ boné

possui apenas um item no antecedente que pode ser generalizado pela Taxonomia τ_1 , ilustrada na Figura 4.4. Assim, é armazenado em $r.última-generalização$ de cada regra o item a ser generalizado, sendo esse item substituído pelo de generalização máxima da Taxonomia τ_1 . Desse modo, na função generaliza-uma-regra são geradas duas regras roupas leves & chinelo $\Rightarrow$ boné e duas regras roupas leves & sandália $\Rightarrow$ boné. Em seguida, é executada a função gera-subconjuntos (linha 6 do Algoritmo 6) que gera dois subconjuntos: um contendo as duas regras roupas leves & chinelo $\Rightarrow$ boné e o outro contendo as duas regras roupas leves & sandália $\Rightarrow$ boné. Nas linhas 7 a 14 do Algoritmo 6 é verificado que o número de regras em cada subconjunto (2 regras) é igual o número de itens que são especializações máximas na Taxonomia τ_1 (2 itens). A verificação valida a generalização e as regras repetidas em cada subconjunto são removidas (linhas 8 e 9 do Algoritmo 6).

As regras roupas leves & chinelo $\Rightarrow$ boné e roupas leves & sandália $\Rightarrow$ boné são armazenadas temporariamente no conjunto de regras generalizadas R_{gt} que é atribuído ao conjunto R na tentativa de generalizá-lo utilizando a Taxonomia τ_2 , ilustrada na Figura 4.4.

A função generaliza-uma-regra é novamente executada, permitindo verificar que as regras roupas leves & chinelo $\Rightarrow$ boné e roupas leves & sandália $\Rightarrow$ boné possuem apenas um item no antecedente que pode ser generalizado utilizando a Taxonomia τ_2 . Assim, é armazenado em $r.última-generalização$ de cada regra o item a ser generalizado, sendo esse item substituído pelo de generalização máxima da Taxonomia τ_2 . Dessa maneira, são geradas duas regras roupas leves & calçados abertos $\Rightarrow$ boné pela função generaliza-uma-regra. Em seguida, dentro da função generaliza-regras é executada a função gera-subconjuntos (linha 6 do Algoritmo 6) que gera um subconjunto contendo as duas regras roupas leves & calçados abertos $\Rightarrow$ boné. Nas linhas 7 a 14 do Algoritmo 6 é verificado que o número de regras do subconjunto gerado (2 regras) é igual o número de itens que são especializações máximas na Taxonomia τ_2 (2 itens). A verificação valida a genera-

lização e as regras repetidas são removidas (linhas 8 e 9 do Algoritmo 6). As regras são armazenadas temporariamente no conjunto de regras generalizadas R_{gt} que é atribuído ao conjunto R , o qual não pode mais ser generalizado pois não há mais taxonomias em $\mathcal{T}$ para generalizá-lo.

O conjunto de regras R contendo a regra roupas leves & calçados abertos $\Rightarrow$ boné é retornado ao algoritmo $\mathcal{GART}$, onde os itens da regra são ordenados lexicograficamente e a regra é armazenada no conjunto R_g . As linhas 3 a 7 do Algoritmo 5 são executadas novamente na tentativa de generalizar o subconjunto de regras E_3 .

Tentativa de generalização do subconjunto E_3

A função *generaliza-regras* (linha 4 do Algoritmo 5) e, por consequência, a função *generaliza-uma-regra* (linha 3 do Algoritmo 6) são executadas novamente na tentativa de generalizar cada uma das regras armazenadas no subconjunto E_3 . Inicialmente na função *generaliza-uma-regra* é verificado que para a Taxonomia τ_1 (Figura 4.4), a regra camiseta & sandália $\Rightarrow$ short possui apenas um item no antecedente que pode ser generalizado. Assim, é armazenado em $r.última-generalização$ da regra o item a ser generalizado, sendo esse item substituído pelo de generalização máxima da Taxonomia τ_1 . Já a regra chinelo & sandália $\Rightarrow$ short não é generalizada, pois a mesma não apresenta itens que possam ser generalizados utilizando a Taxonomia τ_1 . Desse modo, na função *generaliza-uma-regra* são geradas as regras chinelo & sandália $\Rightarrow$ short e roupas leves & sandália $\Rightarrow$ short. Em seguida, dentro da função *generaliza-regras* é executada a função *gera-subconjuntos* (linha 6 do Algoritmo 6) que gera dois subconjuntos: um contendo a regra chinelo & sandália $\Rightarrow$ short e o outro contendo a regra roupas leves & sandália $\Rightarrow$ short. Nas linhas 7 a 14 do Algoritmo 6 é verificado que o número de regras em cada subconjunto (1 regra) é diferente do número de itens que são especializações máximas na Taxonomia τ_1 (2 itens). A verificação invalida o processo de generalização e as generalizações são desfeitas (linha 11 do Algoritmo 6). As regras são armazenadas temporariamente no conjunto de regras generalizadas R_{gt} que é atribuído ao conjunto R na tentativa de generalizá-lo utilizando a Taxonomia τ_2 , ilustrada na Figura 4.4.

Na função *generaliza-uma-regra* é novamente verificado que, para a Taxonomia τ_2 , a regra camiseta & sandália $\Rightarrow$ short possui apenas um item no antecedente que pode ser generalizado. Assim, é armazenado em $r.última-generalização$ da regra o item a ser generalizado, sendo esse item substituído pelo de generalização máxima da Taxonomia τ_2 . Já a regra chinelo & sandália $\Rightarrow$ short não pode ser generalizada pois a mesma apresenta dois itens que podem ser generalizados pela mesma taxonomia: chinelo e sandália. Dessa maneira, na

função *generaliza-uma-regra* são geradas duas regras: **camiseta & calçados abertos $\Rightarrow$ short** e **chinelo & sandália $\Rightarrow$ short**. Em seguida, é executada a função *gera-subconjuntos* (linha 6 do Algoritmo 6) que gera um subconjunto contendo a regra **chinelo & sandália $\Rightarrow$ short** e outro contendo a regra **camiseta & calçados abertos $\Rightarrow$ short**. Nas linhas 7 a 14 do Algoritmo 6 é verificado que o número de regras em cada subconjunto (1 regra) é diferente do número de itens que são especializações máximas na Taxonomia τ_2 (2 itens). A verificação invalida o processo de generalização e as generalizações são desfeitas (linha 11 do Algoritmo 6). As regras são armazenadas temporariamente no conjunto de regras generalizadas R_{gt} que é atribuído ao conjunto R , o qual não pode ser mais generalizado pois não há mais taxonomias em $\mathcal{T}$ para generalizá-lo.

O conjunto de regras R da função *generaliza-regras* é retornado ao algoritmo $\mathcal{GART}$, onde os itens de cada regra são ordenados lexicograficamente e as regras são armazenadas no conjunto R_g . Como não há mais subconjuntos para serem generalizados, as linhas 8 a 12 do algoritmo $\mathcal{GART}$ são executadas para realizar o cálculo da Tabela de Contingência das regras generalizadas armazenadas em R_g . O conjunto de Regras de Associação generalizadas resultante com parte das devidas Tabelas de Contingência é apresentado na Tabela 4.5.

Tabela 4.5: Exemplo 8 – Conjunto de Regras de Associação generalizadas R_g

Regras	$n(LHS \text{ } RHS)$	$n(LHS \overline{RHS})$	$n(\overline{LHS} \text{ } RHS)$	$n(\overline{LHS} \overline{RHS})$
roupas leves $\Rightarrow$ tênis	5	0	0	1
roupas leves & calçados abertos $\Rightarrow$ boné	4	1	1	0
chinelo & sandália $\Rightarrow$ short	1	0	4	1
camiseta & sandália $\Rightarrow$ short	1	1	3	1

No conjunto de Regras de Associação generalizadas R_g , apresentado na Tabela 4.5, os valores da Tabela de Contingência das regras que não foram generalizadas (**chinelo & sandália $\Rightarrow$ short** e **camiseta & sandália $\Rightarrow$ short**) são iguais aos valores, apresentados na Tabela 4.3, para essas mesmas regras. Já os valores da Tabela de Contingência das Regras de Associação generalizadas (**roupas leves $\Rightarrow$ tênis** e **roupas leves & calçados abertos $\Rightarrow$ boné**) não possuem nenhuma relação com os valores da Tabela de Contingência das Regras de Associação não generalizadas apresentadas na Tabela 4.3.

Na próxima seção é descrita a implementação do algoritmo $\mathcal{GART}$ realizada neste trabalho.

4.3 Implementação do Algoritmo $\mathcal{GART}$

O objetivo desta seção é descrever a biblioteca de classes desenvolvida neste trabalho e que implementa o algoritmo $\mathcal{GART}$, sendo relatada a funcionalidade dos métodos implementados e apresentados os arquivos de entrada, os arquivos de saída e a interface do usuário com o algoritmo.

4.3.1 Biblioteca de Classes do Algoritmo $\mathcal{GART}$

A implementação do algoritmo $\mathcal{GART}$, realizada neste trabalho, é composta de três classes: **ARigart**, **ARtaxonomy** e **ARrule**. As classes foram implementadas utilizando a linguagem de programação *Perl*² (Till 1996), que é a linguagem utilizada no desenvolvimento do projeto *Discover*, ao qual este projeto faz parte.

Classe **ARigart**

A classe **ARigart** implementa o algoritmo $\mathcal{GART}$, que utiliza um conjunto de taxonomias, definido por um usuário, para generalizar Regras de Associação. Os métodos dessa classe são descritos na Tabela 4.6.

Tabela 4.6: Descrição dos métodos da classe **ARigart**

Método	Funcionalidade
<i>new</i>	método construtor da classe. Pode ser utilizado em qualquer aplicação
<i>newWeb</i>	método construtor da classe utilizado apenas no Módulo Computacional para Regras de Associação – <i>RuleE-GAR</i> (descrito na Seção 5.3 do Capítulo 5)
<i>alg_igart</i>	método que implementa o algoritmo $\mathcal{GART}$
<i>generalizeRules</i>	generaliza um conjunto de Regras de Associação
<i>groupAssocRules</i>	agrupa as Regras de Associação comparando o antecedente ou o conseqüente das regras. O método facilita a geração de subconjuntos de regras que apresentam antecedentes ou conseqüentes semelhantes
<i>sortTaxonomies</i>	método auxiliar que realiza a ordenação decrescente das taxonomias comparando o número de itens que são especializações máximas em cada taxonomia
<i>makeReports</i>	gera os relatórios de saída do algoritmo $\mathcal{GART}$
<i>getNotAnalyzed</i>	método auxiliar que retorna a primeira posição que possui valor 0 em um <i>array</i> contendo apenas 0s (zeros) e 1s (uns)

²Practical Extraction and Report Language.

Classe **ARtaxonomy**

A classe **ARtaxonomy** é responsável pelo armazenamento das taxonomias utilizadas no processo de generalização das Regras de Associação e por prover métodos que permitam acessar essas taxonomias. A descrição da funcionalidade dos métodos dessa classe é apresenta na Tabela 4.7.

Tabela 4.7: Descrição dos métodos da classe **ARtaxonomy**

Método	Funcionalidade
<i>new</i>	método construtor da classe utilizado para armazenar uma nova taxonomia
<i>init</i>	relaciona o item de generalização máxima com os itens de especialização máxima para cada taxonomia
<i>getFatherItem</i>	retorna o item de generalização máxima na taxonomia
<i>getChildrenItems</i>	retorna os itens que são especializações máximas na taxonomia
<i>getOneChild</i>	retorna um dos itens que são especializações máximas na taxonomia
<i>getNumberChildrenItems</i>	retorna o número de itens que são especializações máximas na taxonomia

Classe **ARrule**

A classe **ARrule** é responsável pelo armazenamento das regras que serão generalizadas e das Regras de Associação já generalizadas. Isso é possível porque as Regras de Associação antes e depois da generalização são representadas na sintaxe padrão definida em Melanda & Rezende (2003). A classe **ARrule** também provê métodos que permitem acessar as regras armazenadas. A descrição da funcionalidade dos métodos dessa classe é apresenta na Tabela 4.8.

Tabela 4.8: Descrição dos métodos da classe **ARrule**

Método	Funcionalidade
<i>new</i>	método construtor da classe utilizado para armazenar uma Regra de Associação
<i>init</i>	configura o lado da regra a ser generalizado (<i>LHS</i> ou <i>RHS</i>)
<i>generalizeOneRule</i>	generaliza uma Regra de Associação, se essa possui apenas um item que pode ser generalizado
<i>calcTC</i>	calcula os valores das frequências relativas $f(LHS\ RHS)$, $f(LHS\ \overline{RHS})$, $f(\overline{LHS}\ RHS)$, $f(\overline{LHS}\ \overline{RHS})$ e o valor de N da Tabela de Contingência de uma Regra de Associação
<i>setRuleID</i>	configura o número de identificação de uma regra

Tabela 4.8: Descrição dos métodos da classe **ARrule** (continuação)

<i>setSpecRuleIDSet</i>	armazena o número de identificação das regras que foram generalizadas
<i>undoGeneralization</i>	desfaz a generalização de uma Regra de Associação
<i>increaseQtyItemsGen</i>	incrementa o número de itens generalizados em uma regra
<i>getPosItemGen</i>	retorna a posição do último item que foi generalizado na regra
<i>resetRule</i>	configura uma Regra de Associação generalizada como sendo uma regra não generalizada
<i>getSideGen</i>	retorna o lado da regra (<i>LHS</i> ou <i>RHS</i>) que será generalizado
<i>getSpecRuleIDSet</i>	retorna os números de identificação das regras que foram generalizadas

4.3.2 Arquivos de Entrada e Saída

Nesta seção são descritos os arquivos de entrada utilizados pelo algoritmo *GART*, e os arquivos de saída gerados por esse algoritmo.

Arquivos de Entrada

O algoritmo *GART* utiliza 3 arquivos de entrada no formato texto. O primeiro arquivo, que possui a extensão *.apr.data*, contém o conjunto de dados transacionais que foram utilizados na mineração das Regras de Associação que serão generalizadas. O arquivo segue o formato padrão definido em Domingues & Rezende (2004), na qual cada transação é representada por uma linha do arquivo e os valores (itens) em cada transação são separados por um espaço em branco. Na Figura 4.5 é apresentado um arquivo de dados contendo 8 transações.

```
camiseta chinelo bone tenis
camiseta sandalia bone tenis
bermuda camiseta
bermuda chinelo bone tenis
bermuda sandalia bone tenis
bone tenis camiseta
sandalia chinelo short tenis
sandalia camiseta short tenis
```

Figura 4.5: Exemplo de um arquivo de dados transacionais *.apr.data*

O segundo arquivo, que possui a extensão *.apr.dcar*, contém o conjunto de Regras de Associação que será generalizado. O conjunto de regras está representado na sintaxe padrão para Regras de Associação definida em Melanda & Rezende (2003). Assim, cada linha do arquivo representa uma Regra de Associação que contém um identificador (*ID*),

o lado esquerdo da regra (LHS), o lado direito da regra (RHS) e a frequência relativa dos valores $f(LHS \text{ } RHS)$, $f(LHS \overline{RHS})$, $f(\overline{LHS} \text{ } RHS)$ e $f(\overline{LHS} \overline{RHS})$ da Tabela de Contingência, além do número total de transações consideradas N . Os valores da Tabela de Contingência são delimitados por “[]” (colchetes). Cada regra possui as informações apresentadas separadas por “,” (vírgula):

$$ID, LHS, RHS, [f(LHS \text{ } RHS), f(LHS \overline{RHS}), f(\overline{LHS} \text{ } RHS), f(\overline{LHS} \overline{RHS}), N]$$

Em Melanda & Rezende (2003) também é apresentada uma biblioteca de classes que converte as Regras de Associação geradas pelo algoritmo *Apriori* e pelos *softwares Mineset*, *MagnumOpus* e *Weka* para o formato padrão descrito. Na Figura 4.6 é apresentado um exemplo de um arquivo de Regras de Associação contendo 8 regras que fazem parte do conjunto de regras geradas a partir da Base de Dados apresentada na Figura 4.5.

```
[R0001],bermuda,tenis,[0.333333,0.000000,0.000000,0.666667,6]
[R0002],camiseta,tenis,[0.500000,0.000000,0.000000,0.500000,6]
[R0003],bermuda & chinelo,bone,[0.166667,0.000000,0.333333,0.500000,6]
[R0004],bermuda & sandalia,bone,[0.166667,0.000000,0.333333,0.500000,6]
[R0005],chinelo & camiseta,bone,[0.166667,0.000000,0.333333,0.500000,6]
[R0006],camiseta & sandalia,bone,[0.166667,0.166667,0.500000,0.166667,6]
[R0007],chinelo & sandalia,short,[0.166667,0.000000,0.666667,0.166667,6]
[R0008],camiseta & sandalia,short,[0.166667,0.166667,0.500000,0.166667,6]
```

Figura 4.6: Exemplo de um arquivo de regras .apr.dcar

O terceiro e último arquivo utilizado pelo algoritmo *GART* possui a extensão .tax e contém o conjunto de taxonomias que será utilizado para generalizar as Regras de Associação. Cada linha do arquivo representa uma taxonomia que contém um item representando a generalização máxima na taxonomia e, um ou mais itens representando especializações máximas na taxonomia. Os itens que representam especializações máximas são delimitados por “()” (parênteses) e separados por “,” (vírgula). Na Figura 4.7 é apresentado um arquivo de taxonomias contendo as 4 taxonomias ilustradas na Figura 4.8.

```
roupas_leves(camiseta,bermuda)
calcados_abertos(sandalia,chinelo)
roupas_sociais(calca,camiseta,sapato)
calcados_sociais(sapato)
```

Figura 4.7: Exemplo do arquivo de taxonomias .tax com as taxonomias da Figura 4.8

Figura 4.8: Um conjunto de taxonomias $\mathcal{T}$ com 4 taxonomias

Caso o usuário deseje utilizar taxonomias que contenham mais do que 2 níveis, como ilustrado na Figura 4.9, o mesmo deve iniciar a escrita do arquivo de taxonomias (.tax) pelos níveis inferiores da taxonomia e terminar pelo nível superior, como é ilustrado na Figura 4.10.

Figura 4.9: Um conjunto $\mathcal{T}$ contendo uma taxonomia de 3 níveis

```

calcados_abertos(sandalia,chinelo)
roupas_leves(camiseta,bermuda)
roupas_esportivas(calcados_abertos,roupas_leves)
  
```

Figura 4.10: Exemplo do arquivo .tax contendo a taxonomia de 3 níveis da Figura 4.9

Arquivos de Saída

O algoritmo *IGART* pode gerar 2 tipos de relatório como saída: um contendo 2 arquivos no formato texto e outro contendo apenas 1 arquivo no formato texto. O primeiro tipo de relatório separa as informações, geradas durante o processo de generalização das Regras de Associação, das regras generalizadas. O arquivo que apresenta as informações do processo de generalização possui a extensão *.gar.report* e é composto por um cabeçalho, nome dos arquivos de entrada utilizados pelo algoritmo, nome dos arquivos de saída gerados pelo algoritmo, número de regras antes e depois do processo de generalização e uma relação dos identificadores das regras generalizadas com os identificadores das que deram origem a essas regras. Já o arquivo que apresenta as Regras de Associação generalizadas possui a extensão *.gar.dcar*. O arquivo de regras generalizadas está representado na sintaxe padrão para Regras de Associação definida em Melanda & Rezende (2003), sendo que, os itens das regras que são generalizações estão delimitados por “()” (parênteses). Exemplos dos arquivos *.gar.report* e *.gar.dcar* são apresentados nas Figuras 4.11 e 4.12.

```
Association Rules Generalization using IGART Algorithm          Copyright (c) Marcos Aurélio Domingues
Date: Fri Dec  5 12:52:29 2003

ALGORITHM INPUT FILE:
DataBase: files_teste_dis/teste.apr.data
Taxonomies Set: files_teste_dis/teste.tax
Rules Set: files_teste_dis/teste.apr.dcar

ALGORITHM OUTPUT FILE:
Generalized Rules: files_teste_dis/teste.gar.dcar
Report File: files_teste_dis/teste.gar.report

# Rules Set: 8
# Generalized Rules: 4

Generalized Rule ID <=> Source Rule IDs
[R0001] <=> [R0002;R0001]
[R0002] <=> [R0006;R0004;R0003;R0005]
[R0003] <=> [R0008]
[R0004] <=> [R0007]
```

Figura 4.11: Exemplo de um arquivo de relatório *.gar.report*

```
[R0001],(roupas_leves),tenis,[0.833333,0.000000,0.000000,0.166667,6]
[R0002],(calcados_abertos) & (roupas_leves),bone,[0.666667,0.166667,0.166667,0.000000,6]
[R0003],camiseta & sandalia,short,[0.166667,0.166667,0.500000,0.166667,6]
[R0004],chinelo & sandalia,short,[0.166667,0.000000,0.666667,0.166667,6]
```

Figura 4.12: Exemplo de um arquivo de regras generalizadas *.gar.dcar*

O segundo tipo de relatório coloca as informações geradas durante o processo de ge-

neralização das Regras de Associação e as próprias regras generalizadas em um único arquivo, como é apresentado na Figura 4.13.

```

Association Rules Generalization using IGART Algorithm          Copyright (c) Marcos Aurélio Domingues
Date: Fri Dec 5 12:52:33 2003

ALGORITHM INPUT FILE:
DataBase: files_teste_dis/teste.apr.data
Taxonomies Set: files_teste_dis/teste.tax
Rules Set: files_teste_dis/teste.apr.dcar

ALGORITHM OUTPUT FILE:
Report File: files_teste_dis/teste.gar.dcar.report

# Rules Set: 8
# Generalized Rules: 4

[Generalized Rule ID], LHS, RHS, [Contingency Table (LHSRHS,LHSnotRHS,notLHSnotRHS,notLHSRHS,N)], [Source Rule IDs]

[R0001],(roupas_leves),tenis,[0.833333,0.000000,0.000000,0.166667,6],[R0001;R0002]
[R0002],(calcados_abertos) & (roupas_leves),bone,[0.666667,0.166667,0.166667,0.000000,6],[R0003;R0005;R0004;R0006]
[R0003],chinelo & sandalia,short,[0.166667,0.000000,0.666667,0.166667,6],[R0007]
[R0004],camiseta & sandalia,short,[0.166667,0.166667,0.500000,0.166667,6],[R0008]

```

Figura 4.13: Exemplo do relatório .gar.dcar.report

O arquivo possui a extensão `.gar.dcar.report` e é composto por um cabeçalho, nome dos arquivos de entrada utilizados pelo algoritmo, nome dos arquivos de saída gerados pelo algoritmo, número de regras antes e depois do processo de generalização e as Regras de Associação generalizadas. As regras generalizadas estão representadas na sintaxe padrão definida em Melanda & Rezende (2003), com as exceções de que os itens das regras que são generalizações estão delimitados por “()” (parênteses) e que ao final de cada regra é colocado entre “[]” (colchetes), os identificadores das regras que originaram a regra generalizada. Esses identificadores são separados por “;” (ponto e vírgula).

4.3.3 Interface com o Usuário

Para interagir com o algoritmo *GART*, e conseqüentemente utilizar as funcionalidades disponíveis, o usuário precisa invocar os métodos implementados nas classes que compõem o algoritmo. Os métodos podem ser invocados de três maneiras. A primeira maneira consiste em invocar os métodos dentro do próprio programa do usuário. Nesse caso, o usuário precisa invocar primeiramente o construtor da classe (por meio do método *new*) e em seguida invocar os métodos que desejar.

A segunda maneira consiste em utilizar um *script* (denominado **generalize.pl**) que atua como uma interface entre o algoritmo e o usuário. Esse *script*, executado por linha de comando do sistema operacional, oculta do usuário detalhes da implementação do algoritmo *GART*. A linha de comando é executada da maneira descrita a seguir:

```
>generalize.pl <stem> <side> [<parâmetros adicionais>]
```

O primeiro parâmetro utilizado na execução é sempre o *stem*, ou seja, o radical dos nomes do conjunto de dados, conjunto de Regras de Associação e conjunto de taxonomias (não considerando a extensão, todos os arquivos devem ter o mesmo nome). O segundo parâmetro utilizado é o *side*. Esse parâmetro, que aceita os valores *left* (lado esquerdo da regra) e *right* (lado direito da regra), define o lado da regra que será generalizado. Já o terceiro parâmetro é a opção do usuário (gerar apenas um relatório de saída e ordenar as taxonomias antes de generalizar as Regras de Associação). As possíveis opções utilizadas para executar o *script* **generalize.pl**, juntamente com suas respectivas linhas de comando são descritas a seguir:

-onlyreport: o padrão do algoritmo é gerar dois arquivos de relatórios, um arquivo contendo as informações do processo de generalização das Regras de Associação (Figura 4.11) e outro arquivo contendo as regras generalizadas (Figura 4.12). A opção **-onlyreport** gera apenas um arquivo de relatório contendo as informações do processo de generalização e as Regras de Associação generalizadas juntas, como ilustrado na Figura 4.13.

>generalize.pl <stem> <side> -onlyreport

-sort: essa opção ordena (em ordem decrescente) o conjunto de taxonomias antes de generalizar o conjunto de Regras de Associação. A ordenação decrescente das taxonomias é realizada utilizando o número de especializações máximas das taxonomias como valores de comparação. Essa ordenação possibilita uma maior generalização das regras, uma vez que as taxonomias que apresentam os maiores números de especializações máximas serão consideradas primeiro durante o processo de generalização. Cabe ressaltar que o uso dessa opção pode ocasionar uma maior generalização, apenas se o arquivo *.tax* possuir duas ou mais taxonomias sobre o mesmo conjunto de itens. Por exemplo, considerando os itens short, bermuda e camiseta, a opção **-sort** pode ocasionar uma maior generalização das regras se o arquivo *.tax* possuir taxonomias como roupa_esporte(short,bermuda) e roupa_esporte(short,bermuda,camiseta), ou seja, duas ou mais taxonomias sobre o mesmo conjunto de itens.

>generalize.pl <stem> <side> -sort

A terceira maneira de utilizar o algoritmo *GART* é por meio do Módulo Computacional para Regras de Associação – *RulEE-GAR*, descrito no Capítulo 5.

4.4 Considerações Finais

Como já citado em outros capítulos, o maior problema da técnica de Regras de Associação é o grande volume de conhecimento (no formato de regras) que é gerado, dificultando a análise e o uso desse conhecimento.

Buscando resolver esse problema, Srikant & Agrawal (1997a) propõe utilizar taxonomias durante o processo de mineração de Regras de Associação, buscando assim reduzir o volume de regras geradas. Embora a técnica apresentada por Srikant & Agrawal (1997a) seja interessante, a mesma possui o inconveniente de gerar um volume de regras muito maior do que o gerado pela técnica que não utiliza taxonomias, uma vez que essa técnica primeiro gera todas as Regras de Associação possíveis (com e sem taxonomias) e, em um segundo momento, busca reduzir esse volume de regras utilizando uma medida para remover as regras que não são interessantes. Como a escolha do valor da medida é realizada de maneira subjetiva (de acordo com o conhecimento do usuário), a mesma não garante que irá ocorrer uma grande redução do volume de regras geradas.

Neste capítulo foi apresentado o algoritmo, denominado *GART* (*Generalization of Association Rules using Taxonomies*), proposto para generalizar as Regras de Associação na etapa de Pós-processamento do conhecimento extraído. O algoritmo proposto busca generalizar as regras sem aumentar o seu volume e sem precisar de uma medida para reduzir o volume de regras geradas.

No próximo capítulo é apresentado o Módulo Computacional para Regras de Associação – *RuleE-GAR*. Esse módulo facilita o uso do algoritmo *GART* e fornece funcionalidades para analisar e avaliar as Regras de Associação generalizadas.

Módulo Computacional para Regras de Associação *RuleE-GAR*

O algoritmo *GART*, descrito no capítulo anterior, foi proposto com o intuito de reduzir o volume de regras geradas pelos algoritmos de Regras de Associação de modo que não haja uma grande perda de conhecimento. A redução do volume de regras facilita a sua análise e o seu uso em um processo de tomada de decisão ou em um Sistema Inteligente.

Neste capítulo é apresentando o módulo computacional *RuleE-GAR* que facilita o uso do algoritmo *GART* e amplia as funcionalidades do *RuleE* (Paula 2003) – um ambiente para análise e exploração de regras de Classificação, Regressão e Associação – viabilizando a análise de Regras de Associação generalizadas. O Ambiente *RuleE* faz parte do projeto *Discover* (Baranauskas & Batista 2000; Rezende, Monard, Batista, Prati, Pugliesi, & Melanda 2004), que vem sendo desenvolvido no LABIC (Laboratório de Inteligência Computacional - ICMC/USP) e que tem por objetivo fornecer um ambiente integrado para apoiar as etapas do processo de descoberta de conhecimento. O capítulo possui a seguinte organização: na Seção 5.1 é apresentado o projeto *Discover*, ao qual este trabalho faz parte. Já na Seção 5.2 é descrito o Ambiente para Exploração de Regras *RuleE*. Na Seção 5.3 é apresentado o módulo computacional *RuleE-GAR*, que provê a generalização de Regras de Associação e a análise das regras generalizadas. Por fim, na Seção 5.4 são

apresentadas algumas considerações referentes a este capítulo.

5.1 O Projeto *Discover*

O projeto *Discover* foi proposto com o objetivo de fornecer um ambiente integrado para apoiar as etapas do processo de Extração de Conhecimento de Dados e Textos (Baranauskas & Batista 2000; Rezende, Monard, Batista, Prati, Pugliesi, & Melanda 2004). Nesse ambiente são utilizados algoritmos de Aprendizado de Máquina implementados pela comunidade, bem como módulos com finalidades específicas desenvolvidos pelos pesquisadores do LABIC. Entre as funcionalidades desses módulos estão: Pré-processamento de dados, Pré-processamento de textos e Pós-processamento de conhecimento. As principais funcionalidades do projeto *Discover* são ilustradas na Figura 5.1.

Figura 5.1: Funcionalidades do projeto *Discover*

No início, o projeto *Discover* consistia apenas de um repositório de *scripts*¹, que seriam implementados na linguagem *Perl* (Till 1996) – *Practical Extraction and Report Language* – e posteriormente reutilizados de modo a facilitar a execução de experimentos. Algumas vantagens dessa linguagem são:

- permite desenvolver códigos de maneira bastante rápida;

¹Scripts são pequenos programas que realizam tarefas atômicas.

- fornece um poderoso mecanismo para trabalhar com expressões regulares, muito utilizadas nessas implementações;
- permite manipular grande quantidade de dados de maneira eficiente e é bastante flexível com relação aos tipos de dados;
- facilita a escrita de protótipos e *scripts*, bem como funções genéricas e pacotes;
- além dessas vantagens, o CPAN², um site que provê diversos recursos para auxiliar o usuário no uso da linguagem *Perl*, fornece um excelente repositório de código reutilizável dessa linguagem.

Portanto, *a priori*, o projeto *Discover* consistia de um conjunto de ferramentas independentes que poderiam ser desenvolvidas conforme as necessidades dos usuários, e reunidas em um repositório compartilhado entre esses usuários. Assim, os usuários poderiam combinar essas ferramentas para executar experimentos mais complexos.

Posteriormente, surgiu a proposta de se criar um ambiente integrado, em que os *scripts* seriam substituídos por bibliotecas de classes. As classes seriam empacotadas como componentes que seriam desenvolvidos por meio de uma interface gráfica. Dessa maneira, os programas seriam desenvolvidos como componentes e o ambiente *Discover* possibilitaria a construção de experimentos por meio da composição desses componentes.

No entanto, para tornar o *Discover* um ambiente integrado, sua implementação tornou-se bem mais complexa. Surgiu a necessidade de uma maior comunicação entre os participantes e de interação entre os diferentes componentes. Com relação à arquitetura do ambiente, em Prati (2003) é proposto um *framework* para a integração dos componentes do Ambiente *Discover* baseado em *software patterns*, no qual os componentes são integrados por meio de uma linguagem baseada em XML. Além disso, a interface gráfica responsável pela composição dos componentes deveria ser implementada de acordo com a arquitetura proposta para o ambiente. Essa interface é apresentada em Geromini (2002).

De um modo geral, o projeto *Discover* pode ser entendido como um conjunto de métodos aplicados sobre os dados ou sobre o conhecimento extraído a partir dos dados. Dessa maneira, é muito importante que o projeto *Discover* ofereça uma maneira consistente para manipular dados e conhecimento. Isso é obtido com o uso de sintaxes padrão para representar os dados e o conhecimento, e também por bibliotecas que oferecem um conjunto de funcionalidades básicas para manipula-los. A sintaxe padrão para representação de dados, denominada DSX (*Discover Dataset Syntax*), é apresentada em Batista (2003). Com

²CPAN – <http://www.cpan.org>.

relação às sintaxes padrão para representação de conhecimento, destacam-se as sintaxes para Regras de Decisão (Prati, Baranauskas, & Monard 2001), para Regras de Regressão (Pugliesi, Dosualdo, & Rezende 2003) e para Regras de Associação (Melanda & Rezende 2003). Entre as diversas ferramentas e sistemas implementados para o projeto *Discover* podem ser citados:

- a biblioteca de classes *DOL* (*Discover Object Library*), que implementa a manipulação de dados na sintaxe padrão (Batista & Monard 2003; Batista 2003). Essa biblioteca apresenta métodos para conversão de dados na sintaxe padrão para os formatos de entrada dos sistemas de aprendizado: *C4.5*, *C4.5rules*, *C5.0/See5*, *Cubist*, *CN2*, *Ripper*, *Trepan*, *Newid*, *RT*, *M5*, *RETIS*, *CART*, *Weka*, *SNNS*, *SVM Torch*, *Apriori*, *MagnumOpus* e *Mineset*. Além disso, a biblioteca fornece vários métodos para pré-processar os dados, entre os quais, podem ser citados: cálculo de estatísticas descritivas básicas, remoção, troca de posição e criação de novos atributos a partir de expressões, embaralhamento, amostragem aleatória e complementar, entre outros;
- a ferramenta **PreText** (Matsubara, Martins, & Monard 2003; Martins 2003), que realiza o Pré-processamento de uma coleção de documentos utilizando a abordagem *bag-of-words*. Entre as principais funcionalidades da ferramenta, podem ser citadas: transformação de palavras escritas em português, espanhol ou inglês, em *stems*; criação da tabela atributo-valor no formato padrão do *Discover* (Batista & Monard 2003; Batista 2003); redução da dimensionalidade do conjunto de atributos usando a lei de Zipf e os cortes de Luhn; utilização de indução construtiva na criação de novos atributos na tabela atributo-valor;
- uma biblioteca que transforma os Classificadores Simbólicos (induzidos pelos algoritmos *ID3*, *C4.5*, *C4.5rules*, *C5.0/See5*, *CN2*, *OC1*, *Ripper*, *T2* e *MC4*) para o formato padrão de Regras de Decisão do *Discover* (Prati, Baranauskas, & Monard 2001);
- uma biblioteca que transforma os Regressores Simbólicos (induzidos pelos algoritmos *M5*, *RT*, *Cubist*, *CART* e *RETIS*) para o formato padrão de Regras de Regressão do *Discover* (Pugliesi, Dosualdo, & Rezende 2003; Dosualdo 2003);
- uma biblioteca que transforma as Regras de Associação (geradas pelo algoritmo *Apriori* e pelos softwares *Weka*, *MagnumOpus* e *Mineset*) para o formato padrão de Regras de Associação do *Discover* (Melanda & Rezende 2003);

- o sistema **XRULER** (Baranauskas 2001), no qual um conjunto de exemplos é dividido em amostras de treinamento e teste. As amostras de treinamento são submetidas a diferentes indutores e os classificadores obtidos são convertidos para o formato padrão de Regras de Decisão. Então, a partir de um subconjunto dessas regras que cobrem os exemplos obtém-se um classificador simbólico final;
- o ambiente computacional **DiPER** (Dosualdo 2003; Dosualdo & Rezende 2003), que auxilia na avaliação de um modelo de regressão simbólico. Entre as principais funcionalidades do ambiente **DiPER** podem ser citadas: conversão dos regressores gerados pelos algoritmos *Cubist*, *RT*, *RETIS/RT*, *CART/RT*, *M5 Model* e *M5 Regression* para o formato padrão de Regras de Regressão definido para o *Discover* (Pugliesi, Dosualdo, & Rezende 2003); aplicação de tratamento semântico ao conjunto de regras no formato padrão; cálculo de medidas de precisão e compreensibilidade para as Regras de Regressão; construção de uma Tabela de Contingência para Regras de Regressão (Pugliesi & Rezende 2003) e cálculo de uma série de medidas derivadas dessa tabela (Lavrač, Flach, & Zupan 1999); e por fim, realização de um teste de hipóteses para comparar os regressores induzidos por diferentes algoritmos de regressão sobre um mesmo conjunto de dados;
- o ambiente computacional **SNIFFER** (Batista & Monard 2003; Batista 2003), que gerencia avaliações e comparações experimentais de algoritmos de Aprendizado de Máquina. Entre suas principais funcionalidades podem ser citadas: gerenciamento de sintaxes dos sistemas de aprendizado, aplicação de métodos de *resampling* para estimar a taxa de erro verdadeira, recuperação das taxas de erro, cálculo e comparação de medidas de desempenho;
- o Ambiente para Exploração de Regras *RuleE* (Paula 2003), que viabiliza tanto a análise quanto a disponibilização de regras de Classificação, Regressão e Associação. Por estar relacionado a implementação do módulo computacional *RuleE-GAR*, esse ambiente é descrito na Seção 5.2.

Além dos trabalhos já implementados descritos aqui, cabe ressaltar que diversos outros trabalhos desenvolvidos ou em desenvolvimento pelos pesquisadores do LABIC, com funcionalidades para processamento de dados e textos, aprendizado de máquina e processamento de padrões fazem parte do projeto *Discover*, entre os quais podem ser citados: Prati (2003), Bernardini (2002), Gonçalves (2002), Imamura (2001), Nagai (2000), Horst (1999), Lee (2000), Pugliesi (2001), Melanda (2002), Geromini (2002), Matsubara (2003), entre outros.

Na próxima seção é descrito o ambiente *RuleE*, que com o desenvolvimento do módulo computacional *RuleE-GAR* (descrito na Seção 5.3) passou a fornecer funcionalidades para generalização de Regras de Associação e também para análise e exploração das regras generalizadas.

5.2 O Ambiente para Exploração de Regras *RuleE*

O Ambiente para Exploração de Regras *RuleE* (Paula 2003) foi desenvolvido com o objetivo de viabilizar tanto a análise quanto a disponibilização de regras de Classificação (Mitchell 1998; Monard & Baranauskas 2003a; Monard & Baranauskas 2003b), Regressão (Weiss & Indurkha 1998) e Associação (Agrawal & Srikant 1994; Adamo 2001). Nesse ambiente, o processo de análise e disponibilização das regras está baseado na idéia de se utilizar um Repositório de Regras e Medidas, que é a entidade principal do ambiente. No repositório são armazenados os conjuntos de regras, os valores das medidas e as consultas realizadas pelos usuários. A arquitetura geral do ambiente *RuleE* é ilustrada na Figura 5.2.

Figura 5.2: Arquitetura geral do ambiente *RuleE*. Fonte: (Paula 2003)

Na Figura 5.2, os elementos que estão no retângulo pontilhado correspondem ao *RuleE*. O ambiente tem como entrada arquivos no formato texto contendo conjuntos de regras de Classificação, Regressão ou Associação representados em suas respectivas sintaxes padrões (Prati, Baranauskas, & Monard 2001; Pugliesi, Dosualdo, & Rezende 2003; Melanda & Rezende 2003). Por utilizar sintaxes padrões de representação de regras e pelo projeto *Discover* possuir bibliotecas de conversão dos diversos tipos de regras para essas sintaxes

padrões, podem ser inseridas no ambiente as saídas de uma grande variedade de sistemas e algoritmos de Extração de Conhecimento de Dados, entre os quais podem ser citados: *C4.5*, *C4.5rules*, *C5.0/See5*, *Cubist*, *CN2*, *Ripper*, *Trepan*, *Newid*, *RT*, *M5*, *RETIS*, *CART*, *Weka*, *SNNS*, *SVM Torch*, *Apriori*, *MagnumOpus* e *Mineset*.

Para avaliar as regras, o *RuleE* apresenta um conjunto de medidas de avaliação. Após um conjunto de regras ter sido inserido no ambiente, os valores das diversas medidas são calculados e ficam disponíveis para os usuários. O ambiente provê fácil acesso às regras e aos valores dessas medidas, e permite ainda que várias medidas sejam utilizadas de forma combinada para a análise (pela aplicação de restrições ou filtros) e ordenação das regras.

Além disso, com o intuito de auxiliar a criação de novas medidas e a realização de experimentos, nesse ambiente é possível definir de maneira simples novas medidas cujo valor pode ser obtido a partir de uma expressão que utiliza o valor de outras medidas. O cálculo do valor de novas medidas é realizado por procedimentos desenvolvidos pelos próprios usuários, bastando ao ambiente invocar esses procedimentos e buscar o valor desejado no arquivo contendo o resultado da execução do procedimento. A implementação do ambiente *RuleE* permite que essas configurações possam ser realizadas facilmente, sem necessidade de alterar o código fonte.

Uma descrição mais detalhada dos elementos representados na Figura 5.2 é fornecida a seguir:

Módulos para Processamento São os programas desenvolvidos para processamento de dados, regras e informações dos usuários. Os módulos para processamento encontrados no ambiente são:

- **Módulo de Gerenciamento:** responsável pelo gerenciamento das regras disponibilizadas no ambiente. Esse módulo recebe requisições da Interface do ambiente, busca as regras correspondentes no Repositório de Regras, invoca os módulos de Pós-Processamento ou recupera algum conjunto de dados no Repositório de Dados, para então retornar o resultado da requisição para a Interface.

O Módulo de Gerenciamento é responsável pela inserção dos conjuntos de regras de Classificação, Regressão e Associação (em seus respectivos formatos padrão) no Repositório de Regras e Medidas e pela obtenção ou cálculo dos valores de medidas para as regras.

- **Módulos de Pós-Processamento:** são os módulos responsáveis pelo cálculo das medidas de avaliação das regras disponibilizadas. Esses módulos contêm procedimentos para cálculo das medidas e apresentam métodos distintos

para tratamento de regras de Classificação, Regressão e Associação. É importante destacar que no ambiente *RuleE* são utilizados os módulos de pós-processamento implementados no projeto *Discover*, sendo que para isso foi definida uma interface para invocar os métodos de pós-processamento quando necessário.

Interface Por meio da Interface, o usuário pode acessar todas as informações disponibilizadas pelo ambiente *RuleE*. A Interface é baseada na *www*, a fim de aproveitar algumas de suas vantagens, como independência de plataforma e, especialmente, facilidade de utilização pelos usuários do processo independente de sua localização física. Pela Interface do ambiente, o usuário interativamente seleciona as regras que julgar mais interessantes, utilizando as medidas disponíveis, para então interpretá-las e utilizá-las. Uma outra função da Interface é coletar, armazenar e disponibilizar informações fornecidas pelos usuários. Uma das telas da interface do *RuleE* é apresentada na Figura 5.3.

Figura 5.3: Interface do ambiente *RuleE*

Cabe ressaltar que Paula (2003), em seu trabalho, não se preocupou com questões de usabilidade da Interface e que essas questões serão abordadas em trabalhos futuros.

Repositórios Correspondem à Base de Dados, desenvolvida utilizando o gerenciador *MySQL* (Axmark, Widenius, & DuBois 2002) e os arquivos nos quais são armaze-

nadas as informações disponibilizadas. No ambiente, são encontrados três tipos de repositórios:

Repositório de Informações de Usuários Conjunto de tabelas nas quais são armazenadas as informações relacionadas aos usuários do ambiente. Algumas informações armazenadas são: cadastro de usuários, comentários e votações em regras.

Repositório de Regras e Medidas Conjunto de tabelas para armazenamento das regras e dos valores das medidas a serem disponibilizadas no ambiente. Um aspecto importante foi a definição do modelo desse repositório, de forma a facilitar o armazenamento e recuperação das regras, o cadastro e inserção de novas medidas para avaliação, e o cálculo e recuperação dos valores das medidas para as regras.

Repositório de Dados Arquivos nos quais são armazenados os dados referentes aos conjuntos de regras disponibilizados. Esses dados podem ser utilizados para auxiliar o usuário na compreensão das regras, assim como podem ser utilizados por algum módulo de Pós-Processamento.

Entre as principais características ou funcionalidades implementadas no ambiente *RuleE* podem ser citadas:

- o cálculo das medidas objetivas para avaliação de regras descritas no *framework* apresentado em Lavrač, Flach, & Zupan (1999) e da medida *RI* (Piatetsky-Shapiro 1991) para a análise das regras;
- a implementação de consultas para recuperação dos conjuntos de regras e dos respectivos valores de medidas;
- a Interface simples e interativa baseada na *www*;
- a possibilidade de análise dos conjuntos de regras utilizando consultas de forma iterativa;
- o armazenamento de valores inseridos por usuários como indicativos da interessabilidade, compreensibilidade, utilidade e novidade das regras, além do armazenamento de comentários anexados às regras dos conjuntos de regras e das consultas.

Além das características citadas, com o desenvolvimento do módulo computacional *RuleE-GAR*, o ambiente *RuleE* passou a fornecer funcionalidades para a generalização

de Regras de Associação e para a análise e exploração das regras generalizadas. O ambiente *RuleE* é detalhadamente descrito em Paula (2003). Já o módulo *RuleE-GAR* é apresentado na próxima seção.

5.3 O Módulo Computacional *RuleE-GAR*

O módulo computacional *RuleE-GAR* foi desenvolvido com dois objetivos. O primeiro é facilitar o uso do algoritmo *GART* (descrito no Capítulo 4) no processo de generalização de Regras de Associação. O segundo objetivo consiste na ampliação das funcionalidades do Ambiente de Exploração de Regras *RuleE*, para que o mesmo possa permitir a análise e a exploração de Regras de Associação generalizadas. A contextualização do módulo *RuleE-GAR* no ambiente *RuleE*, bem como as suas funcionalidades são ilustradas na Figura 5.4.

Figura 5.4: Contextualização e funcionalidades do módulo computacional *RuleE-GAR* no ambiente *RuleE*

O módulo computacional *RuleE-GAR* foi desenvolvido como um Módulo de Pós-Processamento do ambiente *RuleE*. Assim, o módulo *RuleE-GAR* pode utilizar a Base de Dados e os métodos implementados para o *RuleE*, ou seja, pode utilizar toda a estrutura de repositórios e módulos implementada, ilustrada na Figura 5.2 e descrita na Seção 5.2.

Embora o ambiente *RuleE* possua uma estrutura que facilita a inclusão de novos módulos, a mesma teve que ser alterada para a inclusão do módulo *RuleE-GAR*. As principais alterações realizadas são descritas a seguir.

Alterações na Base de Dados

Para armazenar Regras de Associação generalizadas, as tabelas da Base de Dados destinadas ao armazenamento dos conjuntos de regras foram alteradas para que as mesmas pudessem armazenar conjuntos de regras generalizadas (tabelas **RULESET** e **RULE**). Além disso, foram criadas três novas tabelas: **GENERALIZATION_ITEM** (que armazena as taxonomias utilizadas na generalização das regras), **RULE_GENERALIZATION** (que armazena a relação entre as regras generalizadas e as regras originais) e **RULE_GENERALIZATION_ITEM** (que armazena a relação entre as regras generalizadas e as taxonomias que foram utilizadas na sua generalização). O diagrama físico da Base de Dados do ambiente *RuleE* é apresentado na Figura 5.5. As alterações realizadas e as novas tabelas criadas, para permitir o armazenamento de Regras de Associação generalizadas no ambiente *RuleE*, estão destacadas por retângulos em negrito.

Alterações no Módulo de Gerenciamento

Para o ambiente *RuleE* armazenar e acessar as Regras de Associação generalizadas foram implementados vários métodos para as diversas classes que compõem o seu Módulo de Gerenciamento. Uma descrição sucinta dos métodos implementados é apresentada na Tabela 5.1. Já as informações detalhadas sobre as alterações realizadas no ambiente *RuleE*, para permitir que o mesmo trabalhe com Regras de Associação generalizadas, estão disponíveis em Domingues & Rezende (2004).

Na próxima seção é descrita a interface do módulo *RuleE-GAR* e apresentados exemplos de uso da interface na generalização de Regras de Associação e na análise das regras generalizadas.

Figura 5.5: Diagrama físico da Base de Dados do ambiente *RuleEE*

Tabela 5.1: Descrição dos métodos implementados para a manipulação de Regras de Associação generalizadas

Método	Funcionalidade
<i>NewGeneralizationRuleset</i>	insere um conjunto de Regras de Associação generalizadas no ambiente <i>RuleE</i>
<i>GetGeneralizedRulesets</i>	retorna os conjuntos de Regras de Associação generalizadas relacionados a um determinado projeto
<i>GetOriginalRuleset</i>	retorna o conjunto de regras original utilizado na generalização
<i>GetOriginalRules</i>	retorna as regras originais de uma dada regra generalizada
<i>IsGeneralized</i> (para um conjunto de regras)	verifica se um conjunto de regras é ou não generalizado
<i>IsGeneralized</i> (para uma regra)	verifica se uma regra é ou não generalizada
<i>GetGeneralizedBy</i> (para um conjunto de regras)	retorna os conjuntos de regras que são generalizações de um dado conjunto de regras
<i>GetGeneralizedBy</i> (para uma regra)	retorna as regras que são generalizações de uma dada regra
<i>Remove</i> (para um conjunto de regras)	remove um conjunto de regras. Esse método já existia no ambiente e foi alterado para também remover conjuntos de Regras de Associação generalizadas
<i>Remove</i> (para uma regra)	remove uma regra. Esse método já existia no ambiente e foi alterado para também remover uma Regra de Associação generalizada
<i>GetGeneralizationItems</i> (para um conjunto de regras)	retorna as taxonomias que foram utilizadas na generalização de um dado conjunto de regras
<i>GetGeneralizationItems</i> (para uma regra)	retorna as taxonomias que foram utilizadas na generalização de uma regra
<i>GetGeneralization</i>	retorna um conjunto de taxonomias armazenado no ambiente <i>RuleE</i>
<i>GetGeneralizationFileName</i>	retorna o nome de um arquivo no formato texto que contém um conjunto de taxonomias já inserido no ambiente <i>RuleE</i>

5.3.1 Descrição da Interface do Módulo *RuleE-GAR*

A interface do módulo é carregada utilizando o ambiente *RuleE* (disponível no endereço <http://143.107.231.137/rulee/index.html>), ilustrado nas Figuras 5.6 e 5.7. É importante destacar que, para utilizar a interface do módulo *RuleE-GAR* é necessário que o *browser* esteja com a opção *Java Script* habilitada, caso contrário, muitas de suas funcionalidades não estarão acessíveis.

Figura 5.6: *Login* no ambiente *RuleE*

Figura 5.7: *Menu* principal do ambiente *RuleE*

Após a autenticação do usuário no ambiente *RuleE*, pode-se acessar o módulo *RuleE-GAR*, ilustrado nas Figuras 5.8 e 5.9, por meio do *link* Generalização de Regras (Figura 5.7).

Figura 5.8: Interface para generalização de Regras de Associação

Figura 5.9: Interface para análise das Regras de Associação generalizadas

Na Figura 5.8 é ilustrada a interface do módulo *RuleE-GAR* que utiliza o algoritmo *GART* (descrito no Capítulo 4) para generalizar os conjuntos de Regras de Associação armazenados no ambiente *RuleE*. Já na Figura 5.9 é ilustrada a interface que permite analisar e explorar os conjuntos de Regras de Associação generalizadas. A navegação pelo módulo *RuleE-GAR* é realizada utilizando o *menu* disponibilizado no canto superior direito da interface, como pode ser visualizado na Figura 5.9. O *menu* principal do módulo fornece ao usuário as seguintes opções:

Link Generalize Permite o acesso a interface de generalização de Regras de Associação. Essa interface é ilustrada na Figura 5.8 e descrita na Seção 5.3.2.

Link Analyze Fornece acesso a interface de análise e exploração da Regras de Associação generalizadas. A interface é ilustrada na Figura 5.9 e descrita na Seção 5.3.3.

Link Help Permite o acesso ao sistema de ajuda do módulo. Esse sistema é ilustrado na Figura 5.10.

Link Exit Finaliza a execução do módulo e fecha o *browser*.

Figura 5.10: Sistema de ajuda (*Help*) do módulo *RuleEE-GAR*

Na próxima seção é exemplificado o uso do módulo computacional *RuleEE-GAR* na generalização de um conjunto de Regras de Associação.

5.3.2 Generalização de Regras de Associação Utilizando o Módulo *RuleEE-GAR*

O link *Generalize*, descrito anteriormente, dá acesso a tela da interface de generalização de Regras de Associação ilustrada na Figura 5.8. Nessa tela há alguns campos para entrada de dados que permitem ao usuário selecionar um conjunto de regras para ser generalizado. Os campos são descritos a seguir:

Rule Base Nesse campo o usuário seleciona a base de regras da qual será selecionado um conjunto de regras para ser generalizado. Visando facilitar a identificação da base de regras, cada elemento desse campo apresenta as seguintes informações: tipo da base de regras, identificador, data de extração das regras, algoritmo utilizado e o usuário responsável pela inserção da base no ambiente *RuleE*.

Rules with Items Nesse campo devem ser inserido os itens (separados por um espaço em branco) que devem estar presente nas regras selecionadas. Se nenhum item for especificado ou se for inserido o item “*all*”, a seleção irá considerar que qualquer item pode estar presente nas Regras de Associação a serem generalizadas.

linked by Nesse campo o usuário define a relação entre os itens, definidos no campo *Rules with Items*, que devem estar presente na regra. Existem duas opções de relação: a opção “*OR*” que irá selecionar regras que tenham pelo menos um dos itens especificados no campo *Rules with Items* e a opção “*AND*” que irá selecionar apenas as regras que tenham todos os itens especificados.

Rule Side Nesse campo é definido o lado, das Regras de Associação, que deve ser generalizado. Como apresentado no Capítulo 4, o algoritmo *GART* generaliza apenas um lado da regra – lado esquerdo (*LHS*) ou lado direito (*RHS*). Além disso, a escolha do lado da regra a ser generalizado define:

- o lado na qual as restrições dos campos *Rules with Items* e *linked by* deverão ser aplicadas, uma vez que, no módulo implementado, essas restrições são aplicadas apenas no lado das regras que será generalizado;
- o modo como as regras selecionadas deverão ser ordenadas antes de serem apresentadas para o usuário. Se o usuário escolher generalizar o lado esquerdo das regras, as mesmas serão ordenadas primeiro pelo lado direito e em seguida pelo lado esquerdo, caso contrário, as regras serão ordenadas primeiro pelo lado esquerdo e em seguida pelo lado direito. A ordenação do conjunto de regras, antes de sua visualização, facilita a identificação de taxonomias pelo usuário.

Para visualizar o conjunto de Regras de Associação selecionado deve-se utilizar a opção *List Rules*. Para cada regra do conjunto é apresentado o seu número de identificação (*Rule ID*), o seu lado esquerdo (*Left Hand Side*) e o seu lado direito (*Right Hand Side*). Na Figura 5.11 é ilustrada a seleção de um pequeno conjunto de regras para ser generalizado.

Após a seleção do conjunto de regras a ser generalizado, deve-se definir as taxonomias que serão utilizadas na generalização das regras selecionando a opção *Make Taxonomies*. A tela da interface que permite o usuário definir as taxonomias que serão utilizadas na generalização das regras é ilustrada na Figura 5.12.

Figura 5.11: Seleção de um conjunto de Regras de Associação para ser generalizado

Figura 5.12: Definição das taxonomias para a generalização das regras

Na tela apresentada na Figura 5.12, o usuário possui duas maneiras para montar as taxonomias que serão utilizadas na generalização das regras:

Montando as taxonomias manualmente Na montagem manual, o usuário utiliza o

campo *Generalizations* para inserir o nome de cada generalização, ou seja, para inserir o item de generalização máxima na taxonomia. No campo *Specific Items*, o usuário informa os itens específicos que devem ser generalizados, ou seja, os itens que são especializações máximas na taxonomia. Assim, os itens específicos podem ser inseridos na taxonomia utilizando o campo *Specific Items* ou clicando nos itens das regras que serão generalizadas. Como pode ser visualizado na Figura 5.12, os itens das Regras de Associação se tornam *links* para facilitar a montagem das taxonomias.

Carregando um arquivo de taxonomias Com essa opção pode ser utilizado um arquivo de taxonomias (.tax), igual ao definido na Seção 4.3.2 do Capítulo 4, para generalizar as regras. Além das duas maneiras para montar as taxonomias, o usuário pode também salvá-las em um arquivo .tax.

Definidas as taxonomias a serem utilizadas, o usuário pode executar o algoritmo *GART* para generalizar as regras. A execução do algoritmo é realizada na opção *Generalize the Rules*. Ao término da execução do algoritmo é apresentada ao usuário uma mensagem informando que o processo de generalização foi realizado com sucesso, como é ilustrado na Figura 5.13.

Figura 5.13: Término do processo de generalização das Regras de Associação

Após a mensagem, o módulo retorna a interface ilustrada na Figura 5.12, onde o usuário pode realizar uma outra generalização ou pode selecionar o *link Analyze* para

analisar as regras generalizadas. O uso do módulo *RulEE-GAR* na análise de Regras de Associação generalizadas é apresentado na próxima seção.

5.3.3 Análise de Regras de Associação Generalizadas Utilizando o Módulo *RulEE-GAR*

O *link Analyze*, apresentado na Seção 5.3.1, dá acesso a tela da interface de análise de Regras de Associação generalizadas ilustrada na Figura 5.9. Nessa tela há alguns campos para entrada de dados que permitem ao usuário selecionar um conjunto de regras generalizadas para ser analisado. Os campos são descritos a seguir:

Columns Nesse campo devem ser inseridas as colunas a serem visualizadas nos dados finais referentes à consulta. Dentre as colunas que podem ser utilizadas nesse campo, encontram-se *RULE*, referenciando a regra completa, *RULE_NUMBER*, referenciando o número da regra dentro do conjunto de regras, e os nomes das medidas disponíveis (que foram cadastradas no ambiente *RulEE*), por exemplo, *Acc* referenciando a precisão da regra. Uma lista contendo as colunas disponíveis para utilização se encontra ao lado do campo *Columns*.

Rule Base Nesse campo deve ser selecionada a base de regras da qual serão selecionadas as regras a serem apresentadas. Visando facilitar a identificação da base de regras, cada elemento desse campo apresenta as seguintes informações: tipo da base de regras, identificador da base, data da generalização das regras, descrição de que a base foi generalizada e o usuário responsável pela generalização da base de regras. O botão *Remove*, ao lado do campo *Rule Base*, permite o usuário remover do ambiente *RulEE* uma base de Regras de Associação generalizadas.

Restrictions Nesse campo devem ser inseridas as restrições a serem aplicadas às regras visando verificar quais regras devem ser selecionadas pela consulta. No campo podem ser utilizadas restrições sobre quaisquer medidas disponíveis para o conjunto de Regras de Associação generalizadas. Abaixo do campo *Restrictions* existem alguns campos para auxiliar o usuário na elaboração das restrições. Por esses campos, o usuário pode selecionar a coluna sobre a qual deseja aplicar a restrição, o operador a ser utilizado e o valor de teste.

Sorting Nesse campo deve ser especificada a lista de colunas para ordenação das regras, podendo ser utilizadas quaisquer colunas disponíveis para a base de Regras de Associação generalizadas.

Para visualizar o conjunto de Regras de Associação generalizadas deve-se selecionar a opção *List Rules*. Na Figura 5.14 é ilustrada a seleção de um pequeno conjunto de regras generalizadas.

Figura 5.14: Resultado de uma consulta a uma base de Regras de Associação generalizadas

Na Figura 5.14 é possível verificar que além das colunas selecionadas pelo usuário, a consulta retorna outras informações como, por exemplo, o número total de regras da base generalizada (*Rules after generalization*) e da base original (*Rules before generalization*). Junto com cada consulta realizada, a interface disponibiliza quatro *links* na seção *Downloads* para o usuário visualizar ou fazer o *download* dos arquivos contendo, respectivamente, o conjunto de dados transacionais (*link Data Set*), o conjunto de regras originais (*link Rule Set*), o conjunto de regras generalizadas (*link Generalized Rule Set*) e o conjunto de taxonomias utilizadas (*link Taxonomy Set*). Com exceção do arquivo contendo o conjunto de regras generalizadas, os demais arquivos estão no formato descrito na Seção 4.3.2 do Capítulo 4. Já o arquivo que contém o conjunto de regras generalizadas, embora esteja representado na sintaxe padrão definida por Melanda & Rezende (2003), o mesmo apresenta os itens generalizados das regras delimitados por “()” (parênteses). Além disso, ao final de cada regra são listados os identificadores das regras originais (que são disponibilizadas pelo *link Rule Set*) que geraram a regra generalizada. Esses identificadores são separados por “;” (ponto e vírgula) e delimitados por “[]” (colchetes). Um exemplo de cada arquivo

disponível, para visualização ou *download*, é ilustrado nas Figuras 5.15, 5.16, 5.17 e 5.18.

Figura 5.15: Visualização do arquivo de dados transacionais (*link Data Set*)

Figura 5.16: Visualização do arquivo de regras originais (*link Rule Set*)

Figura 5.17: Visualização do arquivo de regras generalizadas (*link Generalized Rule Set*)

Figura 5.18: Visualização do arquivo de taxonomias (*link Taxonomy Set*)

Além dos *links* para visualização ou *download* dos arquivos, cada Regra de Associação generalizada, listada pela consulta, apresenta alguns *links* que permitem explorar informações relacionadas a sua generalização. Os *links* que são descritos a seguir, estão posicionados a esquerda das regras (Figura 5.14).

Link Expanded Rule - E

O *link Expanded Rule*, representado na interface pela letra “E” (Figura 5.14), permite a visualização da regra generalizada de modo expandido, ou seja, os itens generalizados em uma regra (que são generalizações máximas na taxonomia) são substituídos pelos respectivos itens específicos (que são especializações máximas na taxonomia). Na Figura 5.19 é ilustrada a visualização de uma Regra de Associação generalizada expandida.

Figura 5.19: Tela de visualização de uma Regra de Associação generalizada expandida

Link Source Rules - S

O *link Source Rules*, representado na interface pela letra “S” (Figura 5.14), permite ao usuário visualizar as regras originais que geraram uma Regra de Associação generalizada, como é ilustrado na Figura 5.20.

Figura 5.20: Tela de visualização das Regras de Associação que geraram uma regra generalizada

Link Measures - M

O *link Measures*, representado pela letra “M” (Figura 5.14), é visualizado na interface apenas se o usuário seleciona as medidas suporte (*Sup*) e/ou confiança (*Cov*) em sua consulta.

Como o ambiente *RuleE* possibilita que o usuário, ao inserir um conjunto de Regras de Associação, possa também inserir o valores de suporte e confiança mínimos utilizados na mineração desse conjunto, o *Link Measures* permite ao usuário comparar os valores de

suporte e confiança mínimos de um conjunto de regras não generalizadas com os valores de suporte e confiança de uma regra generalizada obtida a partir desse conjunto. Essa análise é importante, pois embora as definições apresentadas nas Seções 2.2.1 e 3.1.2 garantam que o suporte de uma Regra de Associação generalizada é sempre igual ou maior do que o valor do suporte mínimo definido na mineração das regras não generalizadas, a mesma garantia não ocorre para a medida confiança, ou seja, uma Regra de Associação generalizada pode ter confiança menor do que a confiança mínima definida na mineração das regras. Na Figura 5.21 é ilustrada a tela que permite ao usuário comparar os valores das medidas suporte e confiança.

Figura 5.21: Tela de visualização dos valores das medidas suporte e confiança de uma regra generalizada

Na Figura 5.14 também é possível verificar que os itens generalizados em uma regra (que estão delimitados por parênteses) são apresentados como *links*, para que o usuário possa visualizar os itens originais que foram generalizados e geraram esse item. A tela que permite a visualização de itens originais a partir de um item generalizado é ilustrada na Figura 5.22.

Figura 5.22: Tela de visualização de itens originais a partir de um item generalizado

Por fim, o usuário também tem a opção de salvar em um arquivo no formato texto, as informações listadas na consulta que ele considerar importantes. Para isso basta marcar as informações consideradas importantes (utilizando os *checkboxes*) e selecionar a opção *Save Information*.

5.4 Considerações Finais

O projeto *Discover* tem como objetivo fornecer um ambiente integrado para apoiar as etapas do processo de Mineração de Dados e Texto. O *Discover* é desenvolvido integrando os diversos projetos que foram e vêm sendo realizados no LABIC (Laboratório de Inteligência Computacional - ICMC/USP).

O módulo computacional *RuleE-GAR*, descrito neste capítulo, faz parte do projeto *Discover* e foi desenvolvido com o objetivo de facilitar o uso do algoritmo *GART* (*Generalization of Association Rules using Taxonomies*) na generalização de Regras de Associação, além de possibilitar a análise das regras generalizadas. O módulo foi integrado ao Ambiente para Exploração de Regras *RuleE*, que viabiliza tanto a análise quanto a disponibilização de regras de Classificação, Regressão e Associação.

Cabe ressaltar que durante a realização deste trabalho, foi verificada a existência de apenas uma ferramenta que permite o uso de taxonomias na análise e exploração de Regras de Associação. Essa ferramenta, denominada *Association Interestingness Analysis System* (AIAS), permite a análise subjetiva da interessabilidade de Regras de Associação e possibilita de maneira bem simples o uso de taxonomias como filtros, identificando/filtrando as regras que possuem uma maior conformidade com as taxonomias (Liu, Hsu, Chen, & Ma 2000).

No próximo capítulo são apresentados alguns experimentos utilizando o algoritmo *GART*. Tais experimentos foram realizados para mostrar que o uso de taxonomias na generalização de um conjunto de Regras de Associação pode reduzir o volume desse conjunto.

Experimentos Realizados

No Capítulo 4 foi proposto um algoritmo que utiliza taxonomias para generalizar Regras de Associação. A generalização pode reduzir o volume de um conjunto de Regras de Associação facilitando a análise e o uso dessas regras.

Neste capítulo são descritos alguns experimentos realizados com o objetivo de mostrar como a generalização de regras pode reduzir o volume de um conjunto de Regras de Associação. O capítulo possui a seguinte organização: na Seção 6.1 é descrita a Base de Dados utilizada nos experimentos. As atividades de Pré-processamento realizadas na Base de Dados são descritas na Seção 6.2. Na Seção 6.3 é descrito como a Base de Dados foi utilizada para minerar Regras de Associação. Já na Seção 6.4 são descritos os experimentos realizados na etapa de Pós-processamento das regras, que consistem no uso de taxonomias para generalizar Regras de Associação. Nos conjuntos de regras também é possível verificar a presença de algumas taxonomias sem classificação e, por esse motivo, foram realizados alguns experimentos utilizando esse tipo de taxonomia. Por fim, foram realizados alguns experimentos que combinam taxonomias com e sem classificação. Esses experimentos também são apresentados na Seção 6.4. Na Seção 6.5 é apresentada a análise dos resultados obtidos com os experimentos. Já na Seção 6.6 são apresentadas algumas considerações sobre este capítulo.

6.1 Descrição da Base de Dados Utilizada

Os experimentos foram realizados utilizando uma Base de Dados que contém dados sobre as operações de vendas de um supermercado. Essa base possui um arquivo, denominado DBPROD.DBF, onde são armazenados os dados sobre os diversos produtos que são vendidos no supermercado, como: nome do produto, tipo do produto, preço, impostos, data de vencimento.

O restante da base é composta por diversos arquivos no formato texto, sendo que cada arquivo armazena os dados referentes a uma venda. O nome desses arquivos é determinado pelo número da nota fiscal e pelo número da máquina onde foi processada a venda, separados pelo símbolo “.” (ponto), por exemplo, M004870.004, significa número da nota fiscal igual a M004870 e número da máquina igual a 004. Cada arquivo possui várias linhas que contêm números representando os dados de uma venda. Os dois primeiros números de cada linha identificam o tipo da linha no arquivo e podem assumir os seguintes valores: “01”, “02”, “03”, “04”, “05”, “06”, “07”, “08”, “09”, “10” e “36”. As linhas iniciadas pelos números “03”, “02” e “01” são ilustradas na Figura 6.1 e correspondem, respectivamente, aos produtos vendidos, ao subtotal da venda e ao valor total da venda. O significado das demais linhas é confidencial.

036043753960006084655111027090300000.....	(Produto vendido)
031512589880004004861111022090300000.....	(Produto vendido)
026043754120006084655111027090300000.....	(Subtotal das vendas)
016043754120006084655111027090300000.....	(Total das vendas)

Figura 6.1: Exemplo das linhas “03”, “02” e “01” em um arquivo de vendas

Para a realização dos experimentos foram utilizados os dados armazenados nas linhas “01”, “02” e “03”, descritas a seguir. Os dados, nos três tipos de linhas, são armazenados em campos de largura fixa.

Os campos de dados da linha “01”

Os dados armazenados na linha “01” são referentes ao valor total de uma venda. O nome e a descrição de cada campo que compõe a linha “01”, bem como o seu tamanho e a sua posição nessa linha são apresentados na Tabela 6.1.

Tabela 6.1: Descrição dos campos de dados da linha “01”

Nome do Campo	Tamanho (em dígitos)	Posição na Linha (em dígitos)	Descrição
Espec_Oper	2	1 a 2	Identifica o tipo da linha.
Desc1	9	3 a 11	Campo confidencial.
ECP	4	12 a 15	Número da máquina onde foi processada a venda.
CCO	6	16 a 21	Número da nota fiscal.
Desc2	4	22 a 25	Campo confidencial.
Data	6	26 a 31	Data da venda.
Desc3	10	32 a 41	Campo confidencial.
Hora_Inicio	4	42 a 45	Hora do início do processamento da venda.
Soma_Quantidade	9	46 a 54	Este campo soma a quantidade total dos produtos vendidos. Os produtos que são vendidos por peso tem a sua soma (em gramas) armazenada nos três últimos dígitos desse campo.
Total_Venda	12	55 a 66	Valor total da venda.
Desc4	255	67 a 321	Campo confidencial.
Hora_Fim	4	322 a 325	Hora do fim do processamento da venda.
Desc5	67	326 a 392	Campo confidencial.

Os campos de dados da linha “02”

Na linha “02”, os dados armazenados são referentes ao valor subtotal de uma venda. O nome e a descrição dos campos de dados da linha “02”, bem como o seu tamanho e a sua posição nessa linha são apresentados na Tabela 6.2.

Tabela 6.2: Descrição dos campos de dados da linha “02”

Nome do Campo	Tamanho (em dígitos)	Posição na Linha (em dígitos)	Descrição
Espec_Oper	2	1 a 2	Identifica o tipo da linha.
Desc1	9	3 a 11	Campo confidencial.
ECP	4	12 a 15	Número da máquina onde foi processada a venda.
CCO	6	16 a 21	Número da nota fiscal.
Desc2	4	22 a 25	Campo confidencial.
Data	6	26 a 31	Data da venda.
Desc3	10	32 a 41	Campo confidencial.
Tipo_Pgto	3	42 a 44	Forma de pagamento da venda.
Desc4	6	45 a 50	Campo confidencial.
Din_Recebido	12	51 a 62	Valor em dinheiro recebido para pagar a venda.
Troco_Dev	12	63 a 74	Valor do troco devolvido em razão do campo Din_Recebido.

Tabela 6.2: Descrição dos campos de dados da linha “02” (continuação)

P02e07	26	75 a 101	Campo confidencial.
Telefone	16	102 a 117	Telefone do cliente.
Pgto_02_03_05_07	79	118 a 196	Campo confidencial.
Hora_Inicio	4	197 a 200	Hora do início do processamento da venda.
Desc5	141	201 a 341	Campo confidencial.
Aut_Cartão	9	342 a 350	Informação referente a cartão de crédito.
Aut_Cartão2	12	351 a 362	Informação referente a cartão de crédito.
Desc6	11	363 a 374	Campo confidencial.

Os campos de dados da linha “03”

Os dados armazenados na linha “03” são referentes a um produto adquirido em uma compra realizada no supermercado. O nome dos campos de dados da linha “03”, bem como descrição, tamanho e posição dos campos na linha são apresentados na Tabela 6.3.

Tabela 6.3: Descrição dos campos de dados da linha “03”

Nome do Campo	Tamanho (em dígitos)	Posição na Linha (em dígitos)	Descrição
Espec_Oper	2	1 a 2	Identifica o tipo da linha.
Desc1	9	3 a 11	Campo confidencial.
ECP	4	12 a 15	Número da máquina onde foi processada a venda.
CCO	6	16 a 21	Número da nota fiscal.
Desc2	4	22 a 25	Campo confidencial.
Data	6	26 a 31	Data da venda.
Cod_Bar	10	32 a 41	Código de identificação do produto.
Setor	3	42 a 44	Local onde o produto se encontra na loja.
Desc3	19	45 a 63	Campo confidencial.
Quantidade	12	64 a 75	Armazena a quantidade de um determinado produto vendido. Os produtos que são vendidos por peso tem a sua quantidade (em gramas) armazenada nos três últimos dígitos desse campo.
Preco_Total_Produtos	12	76 a 87	Valor total dos produtos iguais.
Desc4	28	88 a 115	Campo confidencial.
Hora_Inicio	4	116 a 119	Hora do início do processamento da venda.
Desc5	49	120 a 168	Campo confidencial.
ICMS	3	169 a 171	Porcentagem de imposto de um determinado produto.
Desc6	87	172 a 258	Campo confidencial.

Cabe ressaltar que mensalmente são gerados cerca de 50000 arquivos no formato texto,

ou seja, são realizadas cerca de 50000 compras no supermercado, gerando em torno de 250 MB (*Mega Bytes*) de dados.

6.2 Pré-Processamento da Base de Dados

Para facilitar a análise dos dados, armazenados nas linhas “01”, “02” e “03” (descritas na Seção 6.1), bem como o seu uso na realização de experimentos, os mesmos foram inseridos em um Sistema de Gerenciamento de Banco de Dados (SGBD). Pela sua simplicidade e facilidade de uso, o *Microsoft Access* foi o SGBD escolhido.

Inicialmente, o arquivo DBPROD.DBF foi importado para o SGBD. Em seguida, foram criadas três tabelas para armazenar alguns campos de dados das linhas “01”, “02” e “03” (descritas na Seção 6.1). A descrição das três tabelas e os campos que compõem cada tabela são apresentados a seguir:

tab_01 Armazena os dados referentes a linha “01”. Essa tabela é composta pelos campos ECP, CCO, Data, Hora_Inicio, Total_Venda e Hora_Fim (apresentados na Tabela 6.1). Além disso, para essa tabela foram criados os seguintes campos:

- Data_Mod – campo Data modificado para o formato DD/MM/AA, em que DD = dia, MM = mês e AA = ano;
- Hora_Inicio_Mod – campo Hora_Inicio modificado para o formato HH:MM:SS, em que HH = horas, MM = minutos e SS = segundos;
- Hora_Fim_Mod – campo Hora_Fim modificado para o formato HH:MM:SS, em que HH = horas, MM = minutos e SS = segundos;
- Duracao – armazena o valor da subtração: Hora_Fim_Mod – Hora_Inicio_Mod;
- Periodo – classificação conforme hora da compra (manhã/tarde/noite).

tab_02 Armazena os dados referentes a linha “02”. Essa tabela é composta pelos campos ECP, CCO, Tipo_Pgto e Din_Recebido (apresentados na Tabela 6.2).

tab_03 Armazena os dados referentes a linha “03”. Os campos que compõem essa tabela no SGBD são: ECP, CCO, Cod_Bar, Setor, Quantidade e Preco_Total_Produtos (apresentados na Tabela 6.3).

O diagrama físico das tabelas criadas no SGBD é ilustrado na Figura 6.2. Devido ao grande número de campos da tabela DBPROD.DBF, no diagrama foi apresentado apenas o campo CODIPROD, que representa o código do produto e é chave primária da tabela DBPROD.DBF.

Figura 6.2: Diagrama físico das tabelas criadas no SGBD

Para a realização dos experimentos, apresentados neste capítulo, foram removidos 158 produtos da tabela DBPROD.DBF que apresentavam códigos de identificação iguais e poderiam influenciar os resultados dos experimentos. Com a remoção, a tabela passou a ter 12548 produtos diferentes. Além disso, os nomes dos produtos na tabela DBPROD.DBF foram alterados para que os mesmos descrevessem apenas o tipo e a marca dos produtos. Assim, dados como tipo e tamanho da embalagem foram removidos dos nomes dos produtos. Essa alteração foi realizada para diminuir a diversidade de produtos na Base de Dados. Durante a realização dos experimentos, a Base de Dados apresentava cerca de 12548 produtos diferentes e apenas dados referentes a três meses de vendas. A grande diversidade de produtos (itens) associada a um pequeno volume de vendas (transações) pode fazer com que uma Base Dados apresente transações muito diversificadas, o que dificulta a mineração de grandes conjuntos de Regras de Associação. A mineração das regras é descrita na próxima seção.

6.3 Mineração das Regras de Associação

Como apresentado na seção anterior, o volume de dados armazenados na base é referente as vendas realizadas no período de três meses. Assim, além de utilizar a base considerando as vendas dos três meses, também foram realizadas quatro partições da mesma para a realização dos experimentos descritos na próxima seção. As partições foram realizadas considerando um dia, sete dias, catorze dias e um mês de vendas. Para a mineração das regras foram utilizados, das transações dos clientes, apenas os nomes do produtos vendidos. As informações sobre os cinco conjuntos de dados utilizados na mineração das Regras de Associação são apresentadas na Tabela 6.4.

Tabela 6.4: Descrição dos conjuntos de dados utilizados nos experimentos

Nome do Conjunto de Dados	# Transações	# Itens	# Itens Diferentes	Descrição
DataSet_1dia	1716	18427	1939	Dados do dia 05/09/2003
DataSet_7dias	10076	108799	2912	Dados dos dias 05/09/2003 a 12/09/2003
DataSet_14dias	20727	219119	3299	Dados dos dias 05/09/2003 a 19/09/2003
DataSet_1mes	37311	425241	3625	Dados do mês 09 de 2003
DataSet_3meses	97854	1129242	4098	Dados dos meses 09, 11 e 12 de 2003

Na mineração das Regras de Associação foi utilizada a implementação do algoritmo *Apriori* realizada por Chistian Borgelt e que se encontra disponível para *download* no site <http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html>. Para a mineração das regras foi utilizado um valor de suporte mínimo igual a 0.5, confiança mínima igual a 0.5 e número máximo de 5 itens por regra. Os conjuntos de regras obtidos são descritos a seguir:

- RuleSet_1dia – com as 32.668 regras geradas a partir do DataSet_1dia;
- RuleSet_7dias – com as 19.166 regras geradas a partir do DataSet_7dias;
- RuleSet_14dias – com as 16.053 regras geradas a partir do DataSet_14dias;
- RuleSet_1mes – com as 21.505 regras geradas a partir do DataSet_1mes;
- RuleSet_3meses – com as 19.936 regras geradas a partir do DataSet_3meses.

Na próxima seção é descrito o Pós-processamento realizado para generalizar as Regras de Associação.

6.4 Pós-Processamento das Regras de Associação

No Pós-processamento das Regras de Associação foram realizados alguns experimentos utilizando o algoritmo proposto neste trabalho – *GART* – para generalizar os conjuntos de Regras de Associação descritos na seção anterior.

Inicialmente foram realizados alguns experimentos para mostrar o uso de taxonomias na generalização de Regras de Associação. Nos conjuntos de regras também é possível verificar a presença de algumas taxonomias sem classificação e, por esse motivo, foram realizados alguns experimentos utilizando esse tipo de taxonomia. Por fim, foram realizados alguns experimentos que combinam taxonomias com e sem classificação na generalização das Regras de Associação.

Cabe ressaltar que os conjuntos de taxonomias, com e sem classificação, foram construídos manualmente para a realização dos experimentos, a partir da análise da Base de Dados e dos conjuntos de Regras de Associação. Além disso, nos experimentos apresentados nesta seção, o algoritmo *GART* foi executado sem a opção **-sort** (Seção 4.3.3 do Capítulo 4).

6.4.1 Pós-Processamento Utilizando Taxonomias

Os experimentos, apresentados nesta seção, foram realizados para mostrar que o uso de taxonomias na generalização de um conjunto de Regras de Associação pode reduzir o volume desse conjunto.

Para facilitar a compreensão dos experimentos, a seguir são apresentados os conjuntos de taxonomias utilizados na generalização das Regras de Associação. Logo após são apresentados os experimentos realizados e a análise dos resultados obtidos.

Conjuntos de Taxonomias Utilizados

Nesta seção são apresentados os conjuntos de taxonomias construídos manualmente, a partir da análise da Base de Dados e dos conjuntos de Regras de Associação, para a realização dos experimentos. Os conjuntos de taxonomias, ilustrados nas Figuras 6.3 a 6.15, foram denominados de “Tax” seguido de um número de identificação, como: Tax01. Além disso, cabe ressaltar que o conjunto de taxonomias Tax13 foi construído a partir da união dos conjuntos Tax01 a Tax12.

Os conjuntos estão no formato do arquivo de taxonomias definido para o algoritmo de generalização de Regras de Associação *GART*, Seção 4.3.2 do Capítulo 4. Nesse formato, cada linha de um arquivo descreve uma taxonomia que contém um item representando a generalização máxima na taxonomia e, um ou mais itens representando especializações máximas na taxonomia. Os itens que representam especializações máximas são delimitados por “()” (parênteses) e separados por “,” (vírgula). Assim, pode-se verificar, por exemplo, que no conjunto de taxonomias ilustrado na Figura 6.3 há uma taxonomia na qual CHOCOLATE generaliza CHOCNESTLE e CHOCLACTA, uma taxonomia na qual CREME_DENTAL generaliza CDSORRISO e CDCOLGATE, uma taxonomia na qual REFRIGERANTE generaliza GUARANA_ANTARTICA, FANTA e COCA_COLA, e assim sucessivamente para as demais taxonomias.

```
CHOCOLATE(CHOCNESTLE,CHOCLACTA)
CREME_DENTAL(CDSORRISO,CDCOLGATE)
REFRIGERANTE(GUARANA_ANTARTICA,FANTA,COCA_COLA)
BISCOITO(BISCNESTLE,BISCNABISCO,BISCPIRAQUE,BISCMARILAN)
DETERGENTE(DETERGYPE,DETERGLIMPOL)
PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
MACARRAO(NISSIM_L_AMEM,MACRENATA,MACGALO)
LEITE(LEITE_PARMALAT,LEITE_NILZA)
FARINHA_TRIGO(FARTRIGO_RENATA,FARTRIGO_DONA_BENTA)
SABONETE(SABVINOLIA,SABLUX)
AMACIANTE(AMACBABY_SOFT,AMACCOMFORT)
MOLHO_TOMATE(EXTRATO_ELEFANTE_KNORRICA,MOLHO_TOMATE_CIRIO)
FEIJAO(FEIJAO_GRAO_DO_CAMPO,FEIJAO_TORRESAN,FEIJAO_BROTO_LEGAL)
CALDOS(CALDO_KNORR,CALDO_MAGGI)
PAES_WICKBOLD(PAO_WICKBOLD_LIGHT,PAO_WICKBOLD_FORMA)
IORGUTE(IOGBATAVO,IOGNESTLE,IOGCORPUS)
ACHOCOLATADO(ACHOCTODDY,ACHOCNESCAU)
SUCOS(TANG,SUCO_DEL_VALLE)
GUARDANAPO_PAPEL(GUARDSANTEPEL,GUARDMASCOT)
ABSORVENTE(ABSCAREFREE,ABSSLIVRE)
CATCHUP(CATCHUP_PIC,CATCHUP_HELLMANN_S)
TOALHA_PAPEL(TOALHA_PAPEL_MASCOT,TOALHA_PAPEL_SNOB)
MANTEIGA(MARGQUALY,MANTEIGA_SAO_CARLOS,MARGDORIANA)
ELMA_CHIPS(BATATA_RUFFLES,BATATA_PALHA_ELMA_CHIPS)
ACUCAR(ACUCAR_DA_BARRA,ACUCAR_CRISTALCUCAR,ACUCAR_UNIAO)
```

Figura 6.3: Conjunto de taxonomias Tax01

PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
 REFRIGERANTE(FANTA,COCA_COLA)
 ACUCAR(ACUCAR_DA_BARRA,ACUCAR_UNIAO)
 BISCOITO(BISCNESTLE,BISCNABISCO)
 AGUA_SANITARIA(AGUA_SANITVAREK,AGUA_SANITCANDURA)
 CREME_DENTAL(CDSORRISO,CDCCLOSEUP)
 MACARRAO(NISSIM_L_AMEM,MACRENATA)
 CALDOS(CALDO_KNORR,CALDO_MAGGI)
 CHOCOLATE(CHOCCNESTLE,CHOCLACTA)
 FEIJAO(FEIJAO_GRAO_DO_CAMPO,FEIJAO_TORRESAN,FEIJAO_BROTO_LEGAL)
 DETERGENTE(DETERGYPE,DETERGLIMPOL)
 MANTEIGA(MARGDORIANA,MARGBECEL)
 FARINHA_TRIGO(FARTRIGO_RENATA,FARTRIGO_DONA_BENTA)
 LEITE(LEITE_LACTUS,LEITE_PARMALAT)
 IOGURTE(IOGNESTLE,IOGCORPUS)
 SABONETE(SABVINOLIA,SABLUX)
 CAFE(CAFE_OURO_BRASILEIRO,CAFE_SERRA_DA_GRAMA)
 ACHOCOLATADO(ACHOCNESCAU,ACHOCTODDY)
 SABAO_PO(SPO_OMO,SPO_MINERVA)
 OLEO(OLEO_SOJA_LIZA,OLEO_SOJA_SOYA)
 AMACIANTE(AMACMON_BIJOU,AMACCOMFORT)
 ABSORVENTE(ABSCAREFREE,ABSSLIVRE)
 SUCOS(SUCO_DEL_VALLE,SUCO_MAGUARY,TANG)
 PAES_WICKBOLD(PAO_WICKBOLD_LIGHT,PAO_WICKBOLD_FORMA,PAO_WICKBOLD_SCASCA)
 ELMA_CHIPS(BATATA_RUFFLES,BATATA_PALHA_ELMA_CHIPS)
 REQUEIJAO(REQCATUPIRY,REQNILZA)
 PRODUTOS_PANCO(BISNAGUINHA_PANCO,PAO_PANCO_FORMA)

Figura 6.4: Conjunto de taxonomias Tax02

ADES(ADES_ABACAXI_ALIMLIQDE_SOJA,ADES_PESSEGO,ADES_UVA)
 DETERGENTE(DETERGYPE,DETERGLIMPOL)
 PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
 REFRIGERANTE(FANTA,SPRITE,COCA_COLA)
 MACARRAO(NISSIM_L_AMEM,MACRENATA)
 CREME_DENTAL(CDSORRISO,CDCCLOSEUP)
 AGUA_SANITARIA(AGUA_SANITCANDURA,AGUA_SANITVAREK)
 LEITE(LEITE_NILZA,LEITE_PARMALAT,LEITE_LACTUS)
 BISCOITO(BISCNESTLE,BISCNABISCO,BISCTOSTINES)
 IOGURTE(IOGCORPUS,IOGNESTLE,IOGMOLICO,IOGBATAVO)
 CHOCOLATE(CHOCCNESTLE,CHOCLACTA)
 MOLHO_TOMATE(EXTRATO_ELEFANTE_KNORRCICA,MOLHO_TOMATE_CIRIO)
 FARINHA_TRIGO(FARTRIGO_DONA_BENTA,FARTRIGO_RENATA)
 OLEO(OLEO_SOJA_SOYA,OLEO_SOJA_LIZA)
 FEIJAO(FEIJAO_TORRESAN,FEIJAO_BROTO_LEGAL,FEIJAO_GRAO_DO_CAMPO)
 SABONETE(SABVINOLIA,SABLUX)
 ACUCAR(ACUCAR_CRISTALCUCAR,ACUCAR_UNIAO)
 SUCOS(SUCO_DEL_VALLE,SUCO_MAGUARY)
 TOALHA_PAPEL(TOALHA_PAPEL_SNOB,TOALHA_PAPEL_MASCOT)
 AMACIANTE(AMACBABY_SOFT,AMACCOMFORT)
 ACHOCOLATADO(ACHOCTODDY,ACHOCNESCAU)
 CALDOS(CALDO_KNORR,CALDO_MAGGI)
 PAES_WICKBOLD(PAO_WICKBOLD_FORMA,PAO_WICKBOLD_LIGHT)
 CEREAL(TRIO,NESTON_BARRA)
 ABSORVENTE(ABSCAREFREE,ABSSLIVRE)
 MANTEIGA(MANTEIGA_SAO_CARLOS,MARGDORIANA)
 ELMA_CHIPS(BATATA_RUFFLES,BATATA_PALHA_ELMA_CHIPS)

Figura 6.5: Conjunto de taxonomias Tax03

PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
 REFRIGERANTE(FANTA,COCA_COLA)
 ACUCAR(ACUCAR_DA_BARRA,ACUCAR_UNIAO)
 BISCOITO(BISCNESTLE,BISCNABISCO)
 AGUA_SANITARIA(AGUA_SANITVAREK,AGUA_SANITCANDURA)
 CREME_DENTAL(CDSORRISO,CDCLOUSEUP)
 MACARRAO(NISSIM_L_AMEM,MACRENATA)
 CALDOS(CALDO_KNORR,CALDO_MAGGI)
 CHOCOLATE(CHOENESTLE,CHOCLACTA)
 FEIJAO(FEIJAO_GRAO_DO_CAMPO,FEIJAO_TORRESAN,FEIJAO_BROTO_LEGAL)
 DETERGENTE(ETERGYPE,ETERGLIMPOL)
 MANTEIGA(MARGDORIANA,MARGBECEL)
 FARINHA_TRIGO(FARTRIGO_RENATA,FARTRIGO_DONA_BENTA)
 LEITE(LEITE_LACTUS,LEITE_PARMALAT)
 IOGURTE(IOGNESTLE,IOGCRPUS)
 SABONETE(SABVINOLIA,SABLUX)

Figura 6.6: Conjunto de taxonomias Tax04

ADES(ADES_ABACAXI_ALIMLIQDE_SOJA,ADES_PESSEGO,ADES_UVA)
 DETERGENTE(ETERGYPE,ETERGLIMPOL)
 PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
 REFRIGERANTE(FANTA,SPRITE,COCA_COLA)
 MACARRAO(NISSIM_L_AMEM,MACRENATA)
 CREME_DENTAL(CDSORRISO,CDCLOUSEUP,CDCLGATE)
 AGUA_SANITARIA(AGUA_SANITCANDURA,AGUA_SANITVAREK)
 LEITE(LEITE_NILZA,LEITE_PARMALAT,LEITE_LACTUS)
 BISCOITO(BISCNESTLE,BISCNABISCO,BISCTOSTINES)
 IOGURTE(IOGCRPUS,IOGNESTLE,IOGMOLICO,IOGBATAVO)
 CHOCOLATE(CHOENESTLE,CHOCLACTA)
 MOLHO_TOMATE(EXTRATO_ELEFANTE_KNORRCICA,MOLHO_TOMATE_CIRIO)
 FARINHA_TRIGO(FARTRIGO_DONA_BENTA,FARTRIGO_RENATA)
 OLEO(OLEO_SOJA_SOYA,OLEO_SOJA_LIZA)
 FEIJAO(FEIJAO_TORRESAN,FEIJAO_BROTO_LEGAL,FEIJAO_GRAO_DO_CAMPO)
 SABONETE(SABVINOLIA,SABLUX)
 ACUCAR(ACUCAR_CRISTALCUCAR,ACUCAR_UNIAO)
 SUCOS(SUCO_DEL_VALLE,SUCO_MAGUARY)
 TOALHA_PAPEL(TOALHA_PAPEL_SNOB,TOALHA_PAPEL_MASCOT)
 AMACIANTE(AMACBABY_SOFT,AMACCOMFORT)
 ACHOCOLATADO(ACHOCTODDY,ACHOCNESCAU)
 CALDOS(CALDO_KNORR,CALDO_MAGGI)
 PAES_WICKBOLD(PAO_WICKBOLD_FORMA,PAO_WICKBOLD_LIGHT)
 BARRA_CEREAL(TRIO,NESTON_BARRA)
 ABSORVENTE(ABSCAREFREE,ABSSLIVRE)
 MANTEIGA(MANTEIGA_SAO_CARLOS,MARGDORIANA)
 ELMA_CHIPS(BATATA_RUFFLES,BATATA_PALHA_ELMA_CHIPS)
 CATCHUP(CATCHUP_PIC,CATCHUP_HELLMANN_S)
 FOSFORO(FOSFORO_FIAT_LUX,FOSFORO_GUARANY)
 REQUEIJAO(REQNILZA,REQPOCOS_DE_CALDAS)
 CAFE(CAFE_MELITTA,CAFE_CABOCLO,CAFE_RIBEIRAO_BONITO,CAFE_SERRA_DA_GRAMA)
 ARROZ(ARROZ_SANO,ARROZ_ALBARUSKA,ARROZ_PRATO_FINO)
 CEREAL(SUCRILHOS_KELLOGG_S,CEREAL_NESTLE)
 SABAO_PO(SPO_CAMPEIRO,SPO_MINERVA,SPO_OMO)
 SOBREMESA_CREMOSA_CHOCOLATE(CHANDELLE,DANETTE)
 OVOS(OVOS_BRANCOS_KAMIMURA,OVOS_ESTRELA)
 QUEIJO_RALADO(QUEIJO_RALADO_TEIXEIRA_4_FORMAGGI,QUEIJO_RALADO_TEIXEIRA_PARMESAO)

Figura 6.7: Conjunto de taxonomias Tax05

PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
 REFRIGERANTE(FANTA,COCA_COLA)
 ACUCAR(ACUCAR_DA_BARRA,ACUCAR_UNIAO)
 BISCOITO(BISCNESTLE,BISCNABISCO)
 AGUA_SANITARIA(AGUA_SANITVAREK,AGUA_SANITCANDURA)
 CREME_DENTAL(CDSORRISO,CDCLOSEUP)
 MACARRAO(NISSIM_L_AMEM,MACRENATA,MACGALO,MACMAGGI,MACBASILAR)
 CALDOS(CALDO_KNORR,CALDO_MAGGI)
 CHOCOLATE(CHOENESTLE,CHOCLACTA)
 FEIJAO(FEIJAO_GRAO_DO_CAMPO,FEIJAO_TORRESAN,FEIJAO_BROTO_LEGAL)
 DETERGENTE(DETERGYPE,DETERGLIMPOL)
 MANTEIGA(MARGDORIANA,MARGBECEL)
 FARINHA_TRIGO(FARTRIGO_RENATA,FARTRIGO_DONA_BENTA)
 LEITE(LEITE_LACTUS,LEITE_PARMALAT)
 IOGURTE(IOGNESTLE,IOGCORPUS)
 SABONETE(SABVINOLIA,SABLUX,SABPROTEX,SABFRANCIS,SABPHEBO,SABPOM_POM)
 CAFE(CAFE_OURO_BRASILEIRO,CAFE_SERRA_DA_GRAMA)
 ACHOCOLATADO(ACHOCNESCAU,ACHOCTODDY)
 SABAO_PO(SPO_OMO,SPO_MINERVA)
 OLEO(OLEO_SOJA_LIZA,OLEO_SOJA_SOYA)
 AMACIANTE(AMACMON_BIJOU,AMACCOMFORT)
 ABSORVENTE(ABSCAREFREE,ABSSLIVRE)
 SUCOS(SUCO_DEL_VALLE,SUCO_MAGUARY,TANG)
 PAES_WICKBOLD(PAO_WICKBOLD_LIGHT,PAO_WICKBOLD_FORMA,PAO_WICKBOLD_SCASCA)
 ELMA_CHIPS(BATATA_RUFFLES,BATATA_PALHA_ELMA_CHIPS)
 REQUEIJAO(REQCATUPIRY,REQNILZA)
 PRODUTOS_PANCO(BISNAGUINHA_PANCO,PAO_PANCO_FORMA)
 TOALHA_PAPEL(TOALHA_PAPEL_MASCOT,TOALHA_PAPEL_SNOB)
 OVOS(OVOS_BRANCOS_KAMIMURA,OVOS_ESTRELA)
 MOLHO_TOMATE(POMAROLA_CICA,SALSARETTI,MOLHO_TOMATE_CIRIO)
 PALMITO(PALMITO_INDAIA,PALMITO_DOIDAO_ACAI)
 ARROZ(ARROZ_PRATO_FINO,ARROZ_ALBARUSKA)
 DESODORANTE(DESREXONA,DESAXE)
 BARRA_CEREAL(TRIO,NESTON_BARRA)
 CEREAL(SUCRILHOS_KELLOGG_S,CEREAL_NESTLE)
 SOBREMESA_CREMOSA_CHOCOLATE(CHANDELLE,DANETTE)

Figura 6.8: Conjunto de taxonomias Tax06

ADES(ADES_ABACAXI_ALIMLIQDE_SOJA,ADES_PESSEGO,ADES_UVA)
 DETERGENTE(DETERGYPE,DETERGLIMPOL)
 PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
 REFRIGERANTE(FANTA,SPRITE,COCA_COLA)
 MACARRAO(NISSIM_L_AMEM,MACRENATA)
 CREME_DENTAL(CDSORRISO,CDCLOSEUP)
 AGUA_SANITARIA(AGUA_SANITCANDURA,AGUA_SANITVAREK)
 LEITE(LEITE_NILZA,LEITE_PARMALAT)
 BISCOITO(BISCNESTLE,BISCNABISCO,BISCTOSTINES)
 IOGURTE(IOGCORPUS,IOGNESTLE)
 CHOCOLATE(CHOENESTLE,CHOCLACTA)
 MOLHO_TOMATE(EXTRATO_ELEFANTE_KNORRCICA,MOLHO_TOMATE_CIRIO)
 FARINHA_TRIGO(FARTRIGO_DONA_BENTA,FARTRIGO_RENATA)
 OLEO(OLEO_SOJA_SOYA,OLEO_SOJA_LIZA)

Figura 6.9: Conjunto de taxonomias Tax07

ADES(ADES_ABACAXI_ALIMLIQDE_SOJA,ADES_PESSEGO,ADES_UVA)
 BISCOITO(BISCSAO_CARLOS,BISCBAUDUCCO,BISCPARMALAT,BISCPIRAQUE,BISCTOSTINES,BISCNESTLE,BISCMARILAN)
 IOGURTE(IOGMOLICO,IOGCRPUS,IOGBATAVO,IOGNESTLE)
 GUARANAS(GUARANA_KUAT,GUARANA_ANTARTICA)
 REFRIGERANTE(COCA_COLA,FANTA,GUARANAS)
 LEITE(LEITE_NILZA,LEITE_PARMALAT,LEITE_LACTUS,LEITE_BATAVO)
 FEIJAO(FEIJAO_TORRESAN,FEIJAO_GRAO_DO_CAMPO,FEIJAO_BROTO_LEGAL)
 ACHOCOLATADO(ACHOCTODDY,ACHOCNESCAU)
 CEREAL(SUCRILHOS_KELLOGG_S,CEREAL_NESTLE)
 SUCOS(SUCO_MAGUARY,SUCO_DEL_VALLE,TANG)
 CREME_DENTAL(CDSORRISO,CDCLOSEUP,CDCOLGATE)
 CHOCOLATE(CHOCLACTA,CHOCNESTLE)
 ACUCAR(ACUCAR_UNIAO,ACUCAR_DA_BARRA)
 PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
 DETERGENTE(DETERGLIMPOL,DETERGYPE)
 MACARRAO(NISSIM_L_AMEM,MACRENATA,MACGALO,MACMAGGI)
 AGUA_SANITARIA(AGUA_SANITVAREK,AGUA_SANITCANDURA)
 FARINHA_TRIGO(FARTRIGO_RENATA,FARTRIGO_DONA_BENTA)
 SABONETE(SABLUX,SABVINOLIA,SABPROTEX,SABDOVE)
 AMACIANTE(AMACBABY_SOFT,AMACCOMFORT)
 POMAROLA(POMAROLA_CICA,POMAROLA_KNORRCICA)
 MOLHO_TOMATE(EXTRATO_ELEFANTE_KNORRCICA,MOLHO_TOMATE_CIRIO)
 MANTEIGA(MANTEIGA_SAO_CARLOS,MARGDORIANA,MARGQUALY)
 PAES_WICKBOLD(PAO_WICKBOLD_LIGHT,PAO_WICKBOLD_FORMA)
 PAES(PAES_WICKBOLD,BISNAGUINHA_PANCO)
 ELMA_CHIPS(BATATA_PALHA_ELMA_CHIPS,BATATA_RUFFLES)
 ESPONJA(ESPONJA_BOMBRIL,ESPONJA_SCOTCH_BRITE)
 ABSORVENTE(ABSSLIVRE,ABSCAREFREE)
 GUARDANAPO(GUARDSANTEPEL,GUARDMASCOT)
 CAFE(CAFE_SERRA_DA_GRAMA,CAFE_PILAO,CAFE_OURO_BRASILEIRO,CAFE_RIBEIRAO_BONITO,CAFE_CABOCLO,
 NESCAFE_TRADICAO)
 OLEO(OLEO_SOJA_SOYA,OLEO_SOJA_LIZA)
 SABAO_PO(SPO_CAMPEIRO,SPO_OMO,SPO_MINERVA)
 REQUEIJAO(REQPAULISTA,REQPOCOS_DE_CALDAS)
 OVOS(OVOS_BRANCOS_KAMIMURA,OVOS_ESTRELA)
 CREME_DE_LEITE(CREME_DE_LEITE_BATAVO,CREME_DE_LEITE_NESTLE)
 FOSFORO(FOSFORO_GUARANY,FOSFORO_FIAT_LUX)
 ALCOOL(ALCOOL_CANDURA,ALCOOL_MULTIALCOOL)
 SOBREMESA_CREMOSA_CHOCOLATE(CHANDELLE,DANETTE)
 ARROZ(ARROZ_ALBARUSKA,ARROZ_SANO,ARROZ_PRATO_FINO)
 GELATINA(GELATINA_OETKER,GELATINA_ROYAL)

Figura 6.10: Conjunto de taxonomias Tax08

CHOCOLATE(CHOCNESTLE,CHOCLACTA)
 REFRIGERANTE(GUARANA_ANTARTICA,FANTA,COCA_COLA)
 BISCOITO(BISCNESTLE,BISCNABISCO,BISCPIRAQUE,BISCARILAN)
 DETERGENTE(DETERGYPE,DETERGLIMPOL)
 PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
 MACARRAO(NISSIM_L_AMEM,MACRENATA,MACGALO)
 CREME_DENTAL(CDSORRISO,CDCOLGATE)
 LEITE(LEITE_PARMALAT,LEITE_NILZA)
 FARINHA_TRIGO(FARTRIGO_RENATA,FARTRIGO_DONA_BENTA)
 SABONETE(SABVINOLIA,SABLUX)

Figura 6.11: Conjunto de taxonomias Tax09

CHOCOLATE(CHOCNESTLE,CHOCCLACTA)
 REFRIGERANTE(GUARANA_ANTARTICA,FANTA,COCA_COLA)
 BISCOITO(BISCNESTLE,BISCNABISCO,BISCPIRAQUE,BISCMARILAN)
 DETERGENTE(DETERGYPE,DETERGLIMPOL)
 PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
 MACARRAO(NISSIM_L_AMEM,MACRENATA,MACGALO)
 CREME_DENTAL(CDSORRISO,CDCOLGATE)
 LEITE(LEITE_PARMALAT,LEITE_NILZA)
 FARINHA_TRIGO(FARTRIGO_RENATA,FARTRIGO_DONA_BENTA)
 SABONETE(SABVINOLIA,SABLUX,SABPHEBO,SABPROTEX,SABDOVE)
 AMACIANTE(AMACBABY_SOFT,AMACCOMFORT)
 MOLHO_TOMATE(EXTRATO_ELEFANTE_KNORRCICA,MOLHO_TOMATE_CIRIO)
 FEIJAO(FEIJAO_GRAO_DO_CAMPO,FEIJAO_TORRESAN,FEIJAO_BROTO_LEGAL)
 CALDOS(CALDO_KNORR,CALDO_MAGGI)
 PAES_WICKBOLD(PAO_WICKBOLD_LIGHT,PAO_WICKBOLD_FORMA)
 IORGUTE(IOGBATAVO,IOGNESTLE,IOGCORPUS)
 ACHOCOLATADO(ACHOCTODDY,ACHOCNESCAU)
 SUCOS(TANG,SUCO_DEL_VALLE)
 GUARDANAPO_PAPEL(GUARDSANTEPEL,GUARDMASCOT)
 ABSORVENTE(ABSCAREFREE,ABSSLIVRE)
 CATCHUP(CATCHUP_PIC,CATCHUP_HELLMANN_S)
 TOALHA_PAPEL(TOALHA_PAPEL_MASCOT,TOALHA_PAPEL_SNOB)
 MANTEIGA(MARGQUALY,MANTEIGA_SAO_CARLOS,MARGDORIANA)
 ELMA_CHIPS(BATATA_RUFFLES,BATATA_PALHA_ELMA_CHIPS)
 ACUCAR(ACUCAR_DA_BARRA,ACUCAR_CRISTALCUCAR,ACUCAR_UNIAO)
 OVOS(OVOS_ESTRELA,OVOS_BRANCOS_KAMIMURA)
 CEREAL(SUCRILHOS_KELLOGG_S,CEREAL_NESTLE)
 ARROZ(ARROZ_SANO,ARROZ_ALBARUSKA)
 OLEO(OLEO_SOJA_SADIA,OLEO_SOJA_SOYA)
 CAFE(CAFE_RIBEIRAO_BONITO,CAFE_OURO_BRASILEIRO)
 REQUEIJAO(REQPAULISTA,REQPOCOS_DE_CALDAS)
 ALCOOL(ALCOOL_CANDURA,ALCOOL_MULTIALCOOL)
 QUEIJO_RALADO(QUEIJO_RALADO_TEIXEIRA_PARMESAO,QUEIJO_RALADO_TEIXEIRA_4_FORMAGGI,
 QUEIJO_RALADO_FAIXA_AZUL)
 SOBREMESA_CREMOSA_CHOCOLATE(CHANDELLE,DANETTE)
 BARRA_CEREAL(TRIO,NESTON_BARRA)
 FOSFORO(FOSFORO_FIAT_LUX,FOSFORO_GUARANY)
 ESPONJA(ESPONJA_SCOTCH_BRITE,ESPONJA_BOMBRIL)
 LEITE_ACHOCOLATADO(TODDYNHO,NESCAU_PRONTINHO)

Figura 6.12: Conjunto de taxonomias Tax10

ADES(ADES_ABACAXI_ALIMLIQDE_SOJA,ADES_PESSEGO,ADES_UVA)
 BISCOITO(BISCSAO_CARLOS,BISCAUDUCCO,BISCPARMALAT,BISCPIRAQUE,BISCTOSTINES,BISCNESTLE,BISCMARILAN)
 IOGURTE(IOGMOLICO,IOGCORPUS,IOGBATAVO,IOGNESTLE)
 REFRIGERANTE(COCA_COLA,FANTA,GUARANA_KUAT,GUARANA_ANTARTICA)
 LEITE(LEITE_NILZA,LEITE_PARMALAT,LEITE_LACTUS,LEITE_BATAVO)
 FEIJAO(FEIJAO_TORRESAN,FEIJAO_GRAO_DO_CAMPO)
 ACHOCOLATADO(ACHOCTODDY,ACHOCNESCAU)
 CEREAL(SUCRILHOS_KELLOGG_S,CEREAL_NESTLE)
 SUCOS(SUCO_MAGUARY,SUCO_DEL_VALLE,TANG)
 CREME_DENTAL(CDSORRISO,CDCLOSEUP,CDCOLGATE)
 CHOCOLATE(CHOCCLACTA,CHOCNESTLE)
 ACUCAR(ACUCAR_UNIAO,ACUCAR_DA_BARRA)
 PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
 DETERGENTE(DETERGLIMPOL,DETERGYPE)

Figura 6.13: Conjunto de taxonomias Tax11

ADES(ADES_ABACAXI_ALIMLIQDE_SOJA,ADES_PESSEGO,ADES_UVA)
 BISCOITO(BISCSAO_CARLOS,BISCBAUDUCCO,BISCPARMALAT,BISCPIRAQUE,BISCTOSTINES,BISCNESTLE,BISCMARILAN)
 IOGURTE(IOGMOLICO,IOGCRYPUS,IOGBATAVO,IOGNESTLE)
 REFRIGERANTE(COCA_COLA,FANTA)
 GUARANAS(GUARANA_KUAT,GUARANA_ANTARTICA)
 LEITE(LEITE_NILZA,LEITE_PARMALAT,LEITE_LACTUS,LEITE_BATAVO)
 FEIJAO(FEIJAO_TORRESAN,FEIJAO_GRAO_DO_CAMPO,FEIJAO_BROTO_LEGAL)
 ACHOCOLATADO(ACHOCTODDY,ACHOCNESCAU)
 CEREAL(SUCRILHOS_KELLOGG_S,CEREAL_NESTLE)
 SUCOS(SUCO_MAGUARY,SUCO_DEL_VALLE,TANG)
 CREME_DENTAL(CDSORRISO,CDCLOSEUP,CDCOLGATE)
 CHOCOLATE(CHOCLACTA,CHOCNESTLE)
 ACUCAR(ACUCAR_UNIAO,ACUCAR_DA_BARRA)
 PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
 DETERGENTE(DETERGLIMPOL,DETERGYPE)
 MACARRAO(NISSIM_L_AMEM,MACRENATA,MACGALO,MACMAGGI)
 AGUA_SANITARIA(AGUA_SANITVAREK,AGUA_SANITCANDURA)
 FARINHA_TRIGO(FARTRIGO_RENATA,FARTRIGO_DONA_BENTA)
 SABONETE(SABLUX,SABVINOLIA)
 AMACIANTE(AMACBABY_SOFT,AMACCOMFORT)
 MOLHO_TOMATE(EXTRATO_ELEFANTE_KNORRICA,MOLHO_TOMATE_CIRIO)
 MANTEIGA(MANTEIGA_SAO_CARLOS,MARGDORIANA)

Figura 6.14: Conjunto de taxonomias Tax12

OLEO(OLEO_SOJA_SOYA,OLEO_SOJA_LIZA)
 OLEO(OLEO_SOJA_SADIA,OLEO_SOJA_SOYA)
 MOLHO_TOMATE(EXTRATO_ELEFANTE_KNORRICA,MOLHO_TOMATE_CIRIO)
 MOLHO_TOMATE(POMAROLA_CICA,SALSARETTI,MOLHO_TOMATE_CIRIO)
 PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
 ACUCAR(ACUCAR_UNIAO,ACUCAR_DA_BARRA)
 ACUCAR(ACUCAR_CRISTALCUCAR,ACUCAR_UNIAO)
 ACUCAR(ACUCAR_DA_BARRA,ACUCAR_CRISTALCUCAR,ACUCAR_UNIAO)
 BISCOITO(BISCNESTLE,BISCNABISCO,BISCPIRAQUE,BISCMARILAN)
 BISCOITO(BISCNESTLE,BISCNABISCO,BISCTOSTINES)
 BISCOITO(BISCSAO_CARLOS,BISCBAUDUCCO,BISCPARMALAT,BISCPIRAQUE,BISCTOSTINES,BISCNESTLE,BISCMARILAN)
 BISCOITO(BISCNESTLE,BISCNABISCO)
 AGUA_SANITARIA(AGUA_SANITVAREK,AGUA_SANITCANDURA)
 CREME_DENTAL(CDSORRISO,CDCLOSEUP,CDCOLGATE)
 CREME_DENTAL(CDSORRISO,CDCOLGATE)
 CREME_DENTAL(CDSORRISO,CDCLOSEUP)
 MACARRAO(NISSIM_L_AMEM,MACRENATA,MACGALO)
 MACARRAO(NISSIM_L_AMEM,MACRENATA,MACGALO,MACMAGGI,MACBASILAR)
 MACARRAO(NISSIM_L_AMEM,MACRENATA)
 MACARRAO(NISSIM_L_AMEM,MACRENATA,MACGALO,MACMAGGI)
 CALDOS(CALDO_KNORR,CALDO_MAGGI)
 CHOCOLATE(CHOCHNESTLE,CHOCLACTA)
 FEIJAO(FEIJAO_TORRESAN,FEIJAO_GRAO_DO_CAMPO)
 FEIJAO(FEIJAO_TORRESAN,FEIJAO_BROTO_LEGAL,FEIJAO_GRAO_DO_CAMPO)
 DETERGENTE(DETERGLIMPOL,DETERGYPE)
 MANTEIGA(MANTEIGA_SAO_CARLOS,MARGDORIANA)
 MANTEIGA(MARGQUALY,MANTEIGA_SAO_CARLOS,MARGDORIANA)
 MANTEIGA(MARGDORIANA,MARGBECEL)
 FARINHA_TRIGO(FARTRIGO_RENATA,FARTRIGO_DONA_BENTA)
 LEITE(LEITE_PARMALAT,LEITE_NILZA)
 LEITE(LEITE_NILZA,LEITE_PARMALAT,LEITE_LACTUS,LEITE_BATAVO)
 LEITE(LEITE_NILZA,LEITE_PARMALAT,LEITE_LACTUS)
 LEITE(LEITE_LACTUS,LEITE_PARMALAT)

Figura 6.15: Conjunto de taxonomias Tax13

```

IORGUTE(IOGNESTLE, IOGCORPUS)
IORGUTE(IOGCORPUS, IOGNESTLE, IOGMOLICO, IOGBATAVO)
IORGUTE(IOGBATAVO, IOGNESTLE, IOGCORPUS)
SOBREMESA_CREMOSA_CHOCOLATE(CHANDELLE,DANETTE)
SABONETE(SABLUX, SABVINOLIA)
SABONETE(SABVINOLIA, SABLUX, SABPHEBO, SABPROTEX, SABDOVE)
SABONETE(SABVINOLIA, SABLUX, SABPROTEX, SABFRANCIS, SABPHEBO, SABPOM_POM)
ADES(ADES_ABACAXI_ALIMLIQDE_SOJA, ADES_PESSEGO, ADES_UVA)
REFRIGERANTE(FANTA, SPRITE, COCA_COLA)
REFRIGERANTE(GUARANA_ANTARTICA, FANTA, COCA_COLA)
REFRIGERANTE(COCA_COLA, FANTA, GUARANA_KUAT, GUARANA_ANTARTICA)
REFRIGERANTE(FANTA, COCA_COLA)
ACHOCOLATADO(ACHOCTODDY, ACHOCNESCAU)
CEREAL(SUCRILHOS_KELLOGG_S, CEREAL_NESTLE)
SUCOS(SUCO_DEL_VALLE, SUCO_MAGUARY, TANG)
SUCOS(TANG, SUCO_DEL_VALLE)
SUCOS(SUCO_DEL_VALLE, SUCO_MAGUARY)
CAFE(CAFE_OURO_BRASILEIRO, CAFE_SERRA_DA_GRAMA)
CAFE(CAFE_MELITTA, CAFE_CABOCLO, CAFE_RIBEIRAO_BONITO, CAFE_SERRA_DA_GRAMA)
CAFE(CAFE_RIBEIRAO_BONITO, CAFE_OURO_BRASILEIRO)
SABAO_PO(SPO_CAMPEIRO, SPO_MINERVA, SPO_OMO)
SABAO_PO(SPO_OMO, SPO_MINERVA)
AMACIANTE(AMACBABY_SOFT, AMACCOMFORT)
AMACIANTE(AMACMON_BIJOU, AMACCOMFORT)
ABSORVENTE(ABSCAREFREE, ABSSLIVRE)
PAES_WICKBOLD(PAO_WICKBOLD_LIGHT, PAO_WICKBOLD_FORMA, PAO_WICKBOLD_SCASCA)
PAES_WICKBOLD(PAO_WICKBOLD_LIGHT, PAO_WICKBOLD_FORMA)
ELMA_CHIPS(BATATA_RUFFLES, BATATA_PALHA_ELMA_CHIPS)
REQUEIJAO(REQNILZA, REQPOCOS_DE_CALDAS)
REQUEIJAO(REQPAULISTA, REQPOCOS_DE_CALDAS)
REQUEIJAO(REQCATUPIRY, REQNILZA)
PRODUTO_PANCO(BISNAGUINHA_PANCO, PAO_PANCO_FORMA)
TOALHA_PAPEL(TOALHA_PAPEL_SNOB, TOALHA_PAPEL_MASCOT)
GUARDANAPO_PAPEL(GUARDSANTEPEL, GUARDMASCOT)
CATCHUP(CATCHUP_PIC, CATCHUP_HELLMANN_S)
GUARANAS(GUARANA_KUAT, GUARANA_ANTARTICA)
OVOS(OVOS_BRANCOS_KAMIMURA, OVOS_ESTRELA)
PALMITO(PALMITO_INDAIA, PALMITO_DOIDAO_ACAI)
ARROZ(ARROZ_PRATO_FINO, ARROZ_ALBARUSKA)
ARROZ(ARROZ_SANO, ARROZ_ALBARUSKA, ARROZ_PRATO_FINO)
ARROZ(ARROZ_SANO, ARROZ_ALBARUSKA)
DESODORANTE(DESREXONA, DESAXE)
BARRA_CEREAL(TRIO, NESTON_BARRA)
FOSFORO(FOSFORO_FIAT_LUX, FOSFORO_GUARANY)
ESPONJA(ESPONJA_SCOTCH_BRITE, ESPONJA_BOMBRIL)
QUEIJO_RALADO(QUEIJO_RALADO_TEIXEIRA_4_FORMAGGI, QUEIJO_RALADO_TEIXEIRA_PARMESAO)
QUEIJO_RALADO(QUEIJO_RALADO_TEIXEIRA_PARMESAO, QUEIJO_RALADO_TEIXEIRA_4_FORMAGGI,
QUEIJO_RALADO_FAIXA_AZUL)
ALCOOL(ALCOOL_CANDURA, ALCOOL_MULTIALCOOL)
LEITE_ACHOCOLATADO(TODDYNHO, NESCAU_PRONTINHO)

```

Figura 6.15: Conjunto de taxonomias Tax13 (continuação)

Como já citado, o conjunto de taxonomias Tax13 foi construído a partir da união dos conjuntos Tax01 a Tax12. Desse modo, no conjunto Tax13 podem haver algumas taxonomias que apresentam o mesmo item de generalização máxima para itens de especialização máxima diferentes. Por exemplo, na Figura 6.15 tem-se duas taxonomias na qual OLEO é o item de generalização máxima, sendo que, em uma taxonomia, o item OLEO generaliza os itens de especialização máxima OLEO_SOJA_SOYA e OLEO_SOJA_LIZA e, na outra, o item OLEO generaliza os itens de especialização máxima OLEO_SOJA_SADIA

e OLEO_SOJA_SOYA.

Na Tabela 6.5 é apresentado o número de taxonomias contidas nos conjuntos de taxonomias Tax01 a Tax13. Nessa tabela alguns conjuntos possuem o mesmo número de taxonomias, porém, com taxonomias diferentes.

Tabela 6.5: Descrição do número de taxonomias armazenadas em cada conjunto

Conjuntos de Taxonomias	Número de Taxonomias
Tax01	26
Tax02	27
Tax03	27
Tax04	16
Tax05	38
Tax06	36
Tax07	14
Tax08	40
Tax09	09
Tax10	40
Tax11	14
Tax12	22
Tax13	82

Cada conjunto de taxonomia apresentado nesta seção foi utilizado na generalização dos conjuntos de Regras de Associação RuleSet_1dia, RuleSet_7dias, RuleSet_14dias, RuleSet_1mes e RuleSet_3meses (descritos na Seção 6.3). Cabe ressaltar que a generalização foi realizada no lado esquerdo das regras. Como apresentado no Capítulo 4, o algoritmo *GART* generaliza apenas um dos lados das Regras de Associação (esquerdo ou direito). Os resultados obtidos com a generalização dos conjuntos de regras são descritos a seguir.

Resultados dos Experimentos

Para verificar o desempenho do algoritmo *GART*, na redução do volume dos conjuntos de Regras de Associação, foram realizados alguns experimentos no qual cada conjunto de taxonomias (descrito na seção anterior) foi submetido ao algoritmo *GART* junto com cada um dos conjuntos de Regras de Associação descritos na Seção 6.3. Os resultados das reduções dos conjuntos de regras são apresentados na Tabela 6.6. Nessa tabela é apresentado o número de regras dos conjuntos de Regras de Associação após serem generalizados por cada um dos conjuntos de taxonomias, sendo que ao lado do número de regras é apresentado, entre parênteses, a porcentagem de redução das regras com relação ao conjunto original (não generalizado). Cabe ressaltar que cada generalização foi realizada utilizando os conjuntos de Regras de Associação originais e que em nenhum momento foram realizadas generalizações utilizando conjuntos de regras já generalizados.

Tabela 6.6: Resultados obtidos com o uso de conjuntos de taxonomias na generalização dos conjuntos de Regras de Associação

Conjuntos de Taxonomias	# regras do RuleSet_1dia	# regras do RuleSet_7dias	# regras do RuleSet_14dias	# regras do RuleSet_1mes	# regras do RuleSet_3meses
Sem Taxonomias	32668 (0%)	19166 (0%)	16053 (0%)	21505 (0%)	19936 (0%)
Tax01	30566 (6,42%)	17225 (10,12%)	14161 (11,78%)	18700 (13,04%)	17603 (11,70%)
Tax02	29969 (8,23%)	17048 (11,05%)	14214 (11,45%)	18657 (13,24%)	17434 (12,55%)
Tax03	30244 (7,42%)	16896 (11,84%)	14109 (12,10%)	18504 (13,95%)	17327 (13,08%)
Tax04	30331 (7,15%)	17178 (10,37%)	14289 (10,98%)	18769 (12,72%)	17547 (11,98%)
Tax05	30182 (7,60%)	16772 (12,49%)	13989 (12,85%)	18309 (14,86%)	17180 (13,82%)
Tax06	29913 (8,43%)	17181 (10,35%)	14325 (10,76%)	18903 (12,09%)	17690 (11,26%)
Tax07	30864 (5,52%)	17305 (9,70%)	14315 (10,82%)	18841 (12,38%)	17694 (11,24%)
Tax08	30446 (6,80%)	17331 (9,57%)	14346 (10,63%)	18807 (12,54%)	17553 (11,95%)
Tax09	31155 (4,63%)	17582 (8,26%)	14465 (9,89%)	19147 (10,96%)	17945 (9,98%)
Tax10	30565 (6,43%)	17219 (10,15%)	14131 (11,97%)	18678 (13,14%)	17578 (11,82%)
Tax11	31365 (3,98%)	18192 (5,08%)	15074 (6,09%)	19908 (7,42%)	18418 (7,61%)
Tax12	30365 (7,04%)	17341 (9,52%)	14355 (10,57%)	18845 (12,36%)	17581 (11,81%)
Tax13	29219 (10,55%)	16413 (14,36%)	13540 (15,65%)	17729 (17,55%)	16684 (16,31%)

Na Tabela 6.6 pode-se verificar que a menor taxa de redução foi de 3,98% (RuleSet_1dia generalizado pela Tax11) e a maior foi de 17,55% (RuleSet_1mes generalizado pela Tax13). Além disso, verifica-se que o conjunto de taxonomias Tax11 (contendo 14 taxonomias) gerou a menor taxa de redução em cada um dos cinco conjuntos de regras, e que Tax13 (contendo 82 taxonomias) proporcionou as maiores taxas de redução para cada um dos cinco conjuntos de regras.

Na Figura 6.16 estão ilustrados os gráficos que mostram as reduções ocorridas em cada conjunto de regras. Em cada gráfico é apresentada a redução do volume de um conjunto de regras para cada conjunto de taxonomias apresentado na seção anterior. As barras, nos gráficos, representam o número de regras do conjunto após ser generalizado utilizando um determinado conjunto de taxonomias. Na parte superior das barras é apresentado o percentual de redução do conjunto de regras após ser generalizado. Além disso, vale lembrar que nos gráficos, a escala do eixo que representa o número de regras inicia em 8000.

A análise dos gráficos da Figura 6.16 possibilita verificar, para cada conjunto de Regras de Associação, que a redução do volume de regras é muito semelhante entre os conjuntos de taxonomias, o que leva a induzir que as mesmas possuem comportamentos semelhantes em cada conjunto de regras, entretanto, nada pode ser afirmado. Além disso, vale ressaltar que embora as taxas de redução tenham sido pequenas, as mesmas não são permanentes, uma vez que o uso de outras taxonomias pode gerar taxas de redução maiores.

(a) Conjunto de regras RuleSet_1dia

(b) Conjunto de regras RuleSet_7dias

(c) Conjunto de regras RuleSet_14dias

(d) Conjunto de regras RuleSet_1mes

(e) Conjunto de regras RuleSet_3meses

Figura 6.16: Gráficos dos resultados obtidos com o uso de conjuntos de taxonomias na generalização dos conjuntos de Regras de Associação

6.4.2 Pós-Processamento Utilizando Taxonomias sem Classificação

Durante a realização dos experimentos, utilizando os conjuntos de taxonomias com classificação na generalização dos conjuntos de Regras de Associação RuleSet_1dia, RuleSet_7dias, RuleSet_14dias, RuleSet_1mes e RuleSet_3meses, foi verificado a presença de algumas taxonomias sem classificação.

O uso do algoritmo *GART* com uma taxonomia sem classificação ao invés do uso de taxonomias com classificação poderia resultar, por exemplo, em regras do tipo:

$$\text{PRODUTOS1} \ \& \ \text{TANG} \Rightarrow \text{ACUCAR_UNIAO}.$$

A regra, do modo como é apresentada, não possui significado completo para o usuário, uma vez que é difícil identificar quais produtos fazem parte do PRODUTOS1. Entretanto, a regra pode apresentar significado completo, se a mesma for analisada utilizando a interface de análise de Regras de Associação generalizadas do módulo computacional *RuleE-GAR* (descrito na Seção 5.3.3 do Capítulo 5). Essa interface permite ao usuário verificar os itens originais que deram origem a um item generalizado, desse modo, a mesma pode ser utilizada para verificar os itens que estão contidos no PRODUTOS1, viabilizando, assim, a compreensão da regra.

Nesse contexto foram realizados alguns experimentos utilizando taxonomias sem classificação. Para facilitar a compreensão dos experimentos, a seguir são apresentados os conjuntos contendo as taxonomias sem classificação, os experimentos realizados e a análise dos resultados obtidos.

Conjuntos de Taxonomias sem Classificação Utilizados

Os conjuntos contendo as taxonomias sem classificação são ilustrados nas Figuras 6.17, 6.18 e 6.19. Os conjuntos estão no formato do arquivo de taxonomias definido para o algoritmo de generalização de regras *GART* (Seção 4.3.2 do Capítulo 4). Cada conjunto, contendo Taxonomias sem classificação, foi denominado de “Tsc” seguido de um número de identificação, como: Tsc01.

```

PRODUTOS1(ACUCAR_UNIAO,LEITE_MOCA,BISCNESTLE,COCA_COLA)
PRODUTOS2(PAPEL_HIGNEVE,BOMBRIL,DETERGLIMPOL,CREME_DE_LEITE_NESTLE)
PRODUTOS3(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
PRODUTOS4(BOMBRIL,DETERGLIMPOL,PAPEL_HIGPERSONAL)
PRODUTOS5(MAIONESE_HELLMANN_S,ARROZ_PRATO_FINO,SAL_CISNE)
PRODUTOS6(BOMBRIL,SPO_OMO,PAPEL_HIGNEVE)

```

Figura 6.17: Tsc01 – Conjunto contendo 6 taxonomias sem classificação

```

PRODUTOS1(ACUCAR_UNIAO,LEITE_MOCA,BISCNESTLE,COCA_COLA)
PRODUTOS2(PAPEL_HIGNEVE,BOMBRIL,DETERGLIMPOL,CREME_DE_LEITE_NESTLE)
PRODUTOS3(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
PRODUTOS4(BOMBRIL,DETERGLIMPOL,PAPEL_HIGPERSONAL)
PRODUTOS5(MAIONESE_HELLMANN_S,ARROZ_PRATO_FINO,SAL_CISNE)
PRODUTOS6(BOMBRIL,SPO_OMO,PAPEL_HIGNEVE)
PRODUTOS7(BISCNESTLE,COCA_COLA)
PRODUTOS8(DETERGLIMPOL,PAPEL_HIGPERSONAL)
PRODUTOS9(LEITE_MOCA,BISCNESTLE,COCA_COLA)
PRODUTOS10(ARROZ_PRATO_FINO,SPO_OMO,BOMBRIL)
PRODUTOS11(MAIONESE_HELLMANN_S,FARTRIGO_RENATA,CREME_DE_LEITE_NESTLE,SPO_OMO,BOMBRIL,DETERGLIMPOL,
ACHOCNESCAU,BISCNESTLE,LEITE_MOCA,ACUCAR_UNIAO)
PRODUTOS12(SPO_OMO,BOMBRIL)
PRODUTOS13(BISCNESTLE,LEITE_MOCA)
PRODUTOS14(SAL_CISNE,ARROZ_PRATO_FINO)
PRODUTOS15(CREME_DE_LEITE_NESTLE,BOMBRIL,ACUCAR_UNIAO)
PRODUTOS16(ARROZ_PRATO_FINO,MAIONESE_HELLMANN_S)

```

Figura 6.18: Tsc02 – Conjunto contendo 16 taxonomias sem classificação

```

PRODUTOS1(ACUCAR_UNIAO,LEITE_MOCA,BISCNESTLE,COCA_COLA)
PRODUTOS2(PAPEL_HIGNEVE,BOMBRIL,DETERGLIMPOL,CREME_DE_LEITE_NESTLE)
PRODUTOS3(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL)
PRODUTOS4(BOMBRIL,DETERGLIMPOL,PAPEL_HIGPERSONAL)
PRODUTOS5(MAIONESE_HELLMANN_S,ARROZ_PRATO_FINO,SAL_CISNE)
PRODUTOS6(BOMBRIL,SPO_OMO,PAPEL_HIGNEVE)
PRODUTOS7(BISCNESTLE,COCA_COLA)
PRODUTOS8(DETERGLIMPOL,PAPEL_HIGPERSONAL)
PRODUTOS9(LEITE_MOCA,BISCNESTLE,COCA_COLA)
PRODUTOS10(ARROZ_PRATO_FINO,SPO_OMO,BOMBRIL)
PRODUTOS11(MAIONESE_HELLMANN_S,FARTRIGO_RENATA,CREME_DE_LEITE_NESTLE,SPO_OMO,BOMBRIL,DETERGLIMPOL,
ACHOCNESCAU,BISCNESTLE,LEITE_MOCA,ACUCAR_UNIAO)
PRODUTOS12(SPO_OMO,BOMBRIL)
PRODUTOS13(BISCNESTLE,LEITE_MOCA)
PRODUTOS14(SAL_CISNE,ARROZ_PRATO_FINO)
PRODUTOS15(CREME_DE_LEITE_NESTLE,BOMBRIL,ACUCAR_UNIAO)
PRODUTOS16(ARROZ_PRATO_FINO,MAIONESE_HELLMANN_S)
PRODUTOS17(ACUCAR_UNIAO,LEITE_MOCA)
PRODUTOS18(CREME_DE_LEITE_NESTLE,ACHOCNESCAU)
PRODUTOS19(ACUCAR_UNIAO,LEITE_MOCA,CREME_DE_LEITE_NESTLE)
PRODUTOS20(BISCNESTLE,SPO_OMO)
PRODUTOS21(SPO_OMO,COCA_COLA)
PRODUTOS22(SAL_CISNE,MAIONESE_HELLMANN_S)
PRODUTOS23(CREME_DE_LEITE_NESTLE,FARTRIGO_RENATA)
PRODUTOS24(ARROZ_PRATO_FINO,SPO_OMO)
PRODUTOS25(FARTRIGO_RENATA,ACHOCNESCAU)
PRODUTOS26(FARTRIGO_RENATA,MAIONESE_HELLMANN_S)

```

Figura 6.19: Tsc03 – Conjunto contendo 26 taxonomias sem classificação

Os três conjuntos contendo taxonomias sem classificação foram utilizados na generalização dos conjuntos de Regras de Associação RuleSet_1dia, RuleSet_7dias, RuleSet_14dias, RuleSet_1mes e RuleSet_3meses. Novamente, a generalização foi realizada no lado esquerdo das regras. Os resultados obtidos com a generalização dos conjuntos de regras são descritos a seguir.

Resultados dos Experimentos

Nesta seção são apresentados alguns experimentos realizados para verificar a redução do volume dos conjuntos de Regras de Associação utilizando taxonomias sem classificação. Nos experimentos, cada conjunto contendo taxonomias sem classificação foi submetido ao algoritmo *GART* junto com cada um dos conjuntos de Regras de Associação descritos na Seção 6.3. Os resultados das reduções dos conjuntos de regras são apresentados na Tabela 6.7. Nessa tabela é apresentado o número de regras dos conjuntos de Regras de Associação após serem generalizados por cada um dos conjuntos de taxonomias sem classificação, sendo que ao lado do número de regras é apresentado, entre parênteses, a porcentagem de redução das regras com relação ao conjunto original (não generalizado). Cabe ressaltar, novamente, que cada generalização foi realizada utilizando os conjuntos de Regras de Associação originais e que em nenhum momento foram realizadas generalizações utilizando conjuntos de regras já generalizados.

Tabela 6.7: Resultados obtidos com o uso de conjuntos de taxonomias sem classificação na generalização dos conjuntos de Regras de Associação

Conjuntos de Tsc	# regras do RuleSet_1dia	# regras do RuleSet_7dias	# regras do RuleSet_14dias	# regras do RuleSet_1mes	# regras do RuleSet_3meses
Sem Tsc	32668 (0%)	19166 (0%)	16053 (0%)	21505 (0%)	19936 (0%)
Tsc01	28073 (14,12%)	14643 (23,59%)	12054 (24,91%)	15826 (26,40%)	15129 (24,11%)
Tsc02	24808 (24,12%)	12867 (32,86%)	10615 (33,87%)	13729 (36,15%)	13323 (33,17%)
Tsc03	22541 (31,06%)	11416 (40,43%)	9387 (41,52%)	12264 (42,97%)	11978 (39,91%)

Na Figura 6.20 são ilustrados os gráficos que mostram as reduções ocorridas em cada conjunto de regras. Em cada gráfico é apresentada a redução do volume de regras de um determinado conjunto para cada conjunto de taxonomias sem classificação apresentado na seção anterior. As barras, nos gráficos, representam o número de regras do conjunto após ser generalizado utilizando um determinado conjunto de taxonomias sem classificação. Na parte superior das barras é apresentado o percentual de redução do conjunto de regras após ser generalizado. Além disso, vale lembrar que nos gráficos, a escala do eixo que representa o número de regras inicia em 8000.

(a) Conjunto de regras RuleSet_1dia

(b) Conjunto de regras RuleSet_7dias

(c) Conjunto de regras RuleSet_14dias

(d) Conjunto de regras RuleSet_1mes

(e) Conjunto de regras RuleSet_3meses

Figura 6.20: Gráficos dos resultados obtidos com o uso de conjuntos de taxonomias sem classificação na generalização dos conjuntos de Regras de Associação

Na Tabela 6.7 pode-se verificar que a menor taxa de redução foi de 14,12% (RuleSet_1dia generalizado pelo Tsc01) e a maior foi de 42,97% (RuleSet_1mes genera-

lizado pelo Tsc03). Comparando esses dados com os da tabela dos experimentos que utilizam conjuntos de taxonomias com classificação na generalização (Tabela 6.6) verifica-se que a menor taxa de redução obtida com os conjuntos de taxonomias sem classificação (14,12%) está próxima da maior taxa de redução obtida com os conjuntos de taxonomias com classificação (17,55%). Além disso, a comparação dos gráficos da Figura 6.20 com os da Figura 6.16 possibilita verificar, para cada conjunto de Regras de Associação, que a redução do volume de regras utilizando conjuntos de taxonomias sem classificação foi muito maior do que a redução utilizando conjuntos de taxonomias com classificação, e que, as taxas de redução entre os conjuntos de taxonomias sem classificação não são tão semelhantes como as dos conjuntos de taxonomias com classificação.

Na próxima seção são apresentados experimentos que combinam conjuntos de taxonomias, com e sem classificação, na generalização dos conjuntos de Regras de Associação.

6.4.3 Combinação de Taxonomias, com e sem Classificação, no Pós-Processamento de Regras de Associação

Nesta seção são apresentados alguns experimentos realizados com o objetivo de verificar o comportamento da combinação dos conjuntos de taxonomias, com e sem classificação, na generalização dos conjuntos de Regras de Associação. Buscando facilitar a compreensão dos experimentos, a seguir são apresentadas as combinações dos conjuntos de taxonomias, os experimentos realizados e a análise dos resultados obtidos.

Combinações de Conjuntos de Taxonomias com e sem Classificação

As combinações dos conjuntos de taxonomias, com e sem classificação, utilizadas nos experimentos, foram criadas considerando os conjuntos de taxonomias com classificação descritos na Seção 6.4.1 e os conjuntos de taxonomias sem classificação apresentados na Seção 6.4.2. Para a realização das combinações foram utilizados os três conjuntos de taxonomias sem classificação e seis conjuntos de taxonomias com classificação. Foram escolhidos dois conjuntos de taxonomias com classificação com o menor número de taxonomias, dois conjuntos com número mediano de taxonomias e dois conjuntos com o maior número de taxonomias com classificação. Cada combinação foi denominada de “Comb” seguida de um número de identificação, como: Comb01. As combinações são apresentadas na Tabela 6.8.

Tabela 6.8: Combinação dos conjuntos de taxonomias com e sem classificação

Nome das Combinações	Combinação dos Conjuntos de Taxonomias	
	com Classificação	com os sem Classificação
Comb01	Tax07	Tsc01
Comb02	Tax09	Tsc01
Comb03	Tax03	Tsc01
Comb04	Tax01	Tsc01
Comb05	Tax13	Tsc01
Comb06	Tax08	Tsc01
Comb07	Tax07	Tsc02
Comb08	Tax09	Tsc02
Comb09	Tax03	Tsc02
Comb10	Tax01	Tsc02
Comb11	Tax13	Tsc02
Comb12	Tax08	Tsc02
Comb13	Tax07	Tsc03
Comb14	Tax09	Tsc03
Comb15	Tax03	Tsc03
Comb16	Tax01	Tsc03
Comb17	Tax13	Tsc03
Comb18	Tax08	Tsc03

As combinações, apresentadas nesta seção, foram utilizadas na generalização dos conjuntos de Regras de Associação RuleSet_1dia, RuleSet_7dias, RuleSet_14dias, RuleSet_1mes e RuleSet_3meses. Novamente a generalização foi realizada no lado esquerdo das regras. Os resultados obtidos com o uso das combinações na generalização dos conjuntos de regras são descritos a seguir.

Resultados dos Experimentos

Para verificar o comportamento do uso das combinações, na redução do volume dos conjuntos de Regras de Associação, foram realizados experimentos em que cada combinação (descrita na seção anterior) foi submetida ao algoritmo *GART* junto com cada um dos conjuntos de Regras de Associação descritos na Seção 6.3. Os resultados das reduções dos conjuntos de regras são apresentados na Tabela 6.9. Nessa tabela é apresentado o número de regras dos conjuntos de Regras de Associação após serem generalizados por cada uma das combinações, sendo que ao lado do número de regras é apresentado, entre parênteses, a porcentagem de redução das regras com relação ao conjunto original (não generalizado). Vale lembrar que cada generalização foi realizada utilizando os conjuntos de Regras de Associação originais e que em nenhum momento foram realizadas generalizações utilizando conjuntos de regras já generalizados.

Tabela 6.9: Resultados obtidos com o uso de combinações de conjuntos de taxonomias, com e sem classificação, na generalização dos conjuntos de regras

Combinações	# regras do RuleSet_1dia	# regras do RuleSet_7dias	# regras do RuleSet_14dias	# regras do RuleSet_1mes	# regras do RuleSet_3meses
Sem Combinação	32668 (0%)	19166 (0%)	16053 (0%)	21505 (0%)	19936 (0%)
Comb01	27687 (15,24%)	14261 (25,59%)	11647 (27,44%)	15166 (29,47%)	14554 (26,99%)
Comb02	27894 (14,61%)	14395 (24,89%)	11689 (27,18%)	15282 (28,93%)	14649 (26,51%)
Comb03	27213 (16,69%)	13908 (27,43%)	11418 (28,87%)	14845 (30,96%)	14215 (28,69%)
Comb04	27418 (16,07%)	14097 (26,44%)	11398 (28,99%)	14900 (30,71%)	14323 (28,15%)
Comb05	26626 (18,49%)	13712 (28,45%)	11120 (30,72%)	14468 (32,72%)	13867 (30,44%)
Comb06	27427 (16,04%)	14078 (26,54%)	11434 (28,77%)	14825 (31,06%)	14162 (28,96%)
Comb07	24307 (25,59%)	12333 (35,65%)	10068 (37,28%)	12881 (40,10%)	12613 (36,73%)
Comb08	24437 (25,19%)	12414 (35,22%)	10067 (37,28%)	12929 (39,87%)	12638 (36,60%)
Comb09	23865 (26,94%)	11985 (37,46%)	9855 (38,60%)	12567 (41,56%)	12279 (38,40%)
Comb10	23998 (26,53%)	12115 (36,78%)	9792 (39,00%)	12555 (41,61%)	12320 (38,20%)
Comb11	23369 (28,46%)	11866 (38,08%)	9604 (40,17%)	12266 (42,96%)	11982 (39,89%)
Comb12	23958 (26,66%)	12072 (37,01%)	9811 (38,88%)	12475 (41,99%)	12158 (39,01%)
Comb13	22041 (32,53%)	10859 (43,34%)	8848 (44,88%)	11407 (46,95%)	11255 (43,54%)
Comb14	22199 (32,04%)	10964 (42,79%)	8862 (44,79%)	11498 (46,53%)	11289 (43,37%)
Comb15	21671 (33,66%)	10561 (44,89%)	8660 (46,05%)	11133 (48,23%)	10959 (45,02%)
Comb16	21834 (33,16%)	10705 (44,14%)	8613 (46,34%)	11155 (48,12%)	11007 (44,78%)
Comb17	21471 (34,27%)	10699 (44,17%)	8711 (45,73%)	11228 (47,78%)	10980 (44,92%)
Comb18	21828 (33,18%)	10670 (44,32%)	8646 (46,14%)	11087 (48,44%)	10868 (45,48%)

Na Figura 6.21 são ilustrados os gráficos que mostram as reduções ocorridas em cada conjunto de regras. Em cada gráfico é apresentada a redução do volume de regras de um determinado conjunto para cada combinação apresentada na seção anterior. As barras, nos gráficos, representam o número de regras do conjunto após ser generalizado utilizando uma determinada combinação. Na parte superior das barras é apresentado o percentual de redução do conjunto de regras após ser generalizado. Vale lembrar, que nos gráficos, a escala do eixo que representa o número de regras inicia em 8000.

Já na Figura 6.22 são ilustrados três gráficos que mostram as reduções dos conjuntos de Regras de Associação quando esses são generalizados por um conjunto de taxonomias (com classificação), um conjunto de taxonomias sem classificação e pela combinação de ambos os conjuntos de taxonomias (com e sem classificação). Para construir os três gráficos foram escolhidas três das combinações apresentadas na seção anterior: Comb02 – que gerou as menores taxas de redução de volume nos conjuntos de regras, Comb09 – que gerou taxas medianas de redução e Comb18 – que gerou as maiores taxas de redução de volume dos conjuntos de regras. Em cada gráfico é apresentado, para os cinco conjuntos de Regras de Associação, as taxas de redução obtidas com o uso da combinação e com o uso dos conjuntos de taxonomias, com e sem classificação, que compõem a combinação. Na parte superior das barras é apresentado o percentual de redução do conjunto de regras.

(a) Conjunto de regras RuleSet_1dia

(b) Conjunto de regras RuleSet_7dias

(c) Conjunto de regras RuleSet_14dias

(d) Conjunto de regras RuleSet_1mes

(e) Conjunto de regras RuleSet_3meses

Figura 6.21: Gráficos dos resultados obtidos com o uso de combinações de conjuntos de taxonomias, com e sem classificação, na generalização dos conjuntos de Regras de Associação

(a) Comparação das taxas de redução obtidas com Tax09, Tsc01 e Comb02

(b) Comparação das taxas de redução obtidas com Tax03, Tsc02 e Comb09

(c) Comparação das taxas de redução obtidas com Tax08, Tsc03 e Comb18

Figura 6.22: Gráficos comparando as taxas de redução obtidas com os conjuntos de taxonomias, com e sem classificação, e com a combinação de ambos os conjuntos

Na Tabela 6.9 pode-se verificar que a menor taxa de redução foi de 14,61% (RuleSet_1dia generalizado pela Comb02) e a maior foi de 48,44% (RuleSet_1mes generalizado pela Comb18). Comparando esses dados com os da tabela dos experimentos que utilizam conjuntos de taxonomias (com classificação) na generalização dos conjuntos de regras (Tabela 6.6) e com os da tabela dos experimentos que utilizam conjuntos de taxonomias sem classificação (Tabela 6.7), verifica-se que a menor taxa de redução obtida com as combinações (14,61%) está próxima da maior taxa de redução obtida com os conjuntos de taxonomias (17,55%) e muito próxima da menor taxa obtida com os conjuntos de taxonomias sem classificação (14,12%). Além disso, a maior taxa obtida com as combinações (48,44%) está próxima da maior taxa obtida com os conjuntos de taxonomias sem classificação (42,97%). Além da proximidade da maior e menor taxa de redução entre as combinações e os conjuntos de taxonomias sem classificação, os dados apresentados nas Tabelas 6.7 e 6.9 mostram que as taxas de redução nas duas tabelas estão muito próximas, o que leva a induzir que o acréscimo dos conjuntos de taxonomias (com classificação) aos de taxonomias sem classificação não exerce grande influência na redução de volume dos conjuntos de regras.

A análise dos gráficos da Figura 6.21 possibilita verificar, para cada conjunto de Regras de Associação, que a redução do volume de regras é muito semelhante entre as combinações que apresentam conjuntos de taxonomias (com classificação) diferentes, mas o mesmo conjunto de taxonomias sem classificação em sua composição. Já as combinações que apresentam conjuntos de taxonomias sem classificação diferentes em sua composição, a redução do volume de regras apresenta uma variação considerável, independente do conjunto de taxonomias (com classificação) que compõe a combinação, assim, pode-se verificar que o comportamento dos conjuntos de taxonomias, com e sem classificação, continuam presentes nas combinações.

Com os gráficos da Figura 6.22 é possível comparar as taxas de redução de volume dos conjuntos de Regras de Associação para as três possibilidades de generalização de regras adotadas nos experimentos, utilizando conjuntos de taxonomias (com classificação), conjuntos de taxonomias sem classificação e combinações. Pode-se verificar que as taxas de redução de volume dos conjuntos de regras, utilizando conjuntos de taxonomias sem classificação e combinações na generalização, estão muito próximas e, que ambas apresentam uma distância considerável com relação as taxas de redução obtidas com o uso de conjuntos de taxonomias (com classificação) na generalização dos conjuntos de Regras de Associação.

6.5 Síntese Geral dos Experimentos

Os experimentos apresentados na Seção 6.4.1 mostram que a generalização utilizando conjuntos de taxonomias possibilita a redução do volume dos conjuntos de Regras de Associação. Embora nos experimentos realizados, a maior taxa de redução tenha sido de 17,55%, vale ressaltar que, como os conjuntos de taxonomias são construídos pelo usuário, outros conjuntos de taxonomias podem gerar taxas de redução maiores do que as apresentadas nos experimentos.

Nos experimentos com os conjuntos de taxonomias sem classificação na generalização dos conjuntos de Regras de Associação (Seção 6.4.2), foi possível verificar altas taxas de redução de volume dos conjuntos de regras. Entretanto, para realizar a generalização utilizando conjuntos de taxonomias sem classificação, há a necessidade de uma ferramenta apropriada para a análise das regras generalizadas, caso contrário, a generalização irá resultar um conjunto de regras generalizadas de significado incompleto para o usuário. Nesses experimentos realizados, alguns conjuntos de Regras de Associação chegaram a atingir taxas de redução acima de 40% de seu volume.

Já os experimentos que combinam conjuntos de taxonomias, com e sem classificação, para generalizar Regras de Associação (Seção 6.4.3) mostram que os conjuntos de taxonomias sem classificação apresentam uma influência muito maior do que os de taxonomias com classificação durante a generalização das regras. Tal fato é observado pelas taxas de redução obtidas estarem muito próximas das taxas de redução obtidas com os conjuntos de taxonomias sem classificação, Seção 6.4.2.

Além dos experimentos apresentados anteriormente neste capítulo, foram realizados alguns experimentos utilizando a opção **-sort** do algoritmo *GART*. Como apresentado na Seção 4.3.3 do Capítulo 4, a opção **-sort** ordena (em ordem decrescente) o conjunto de taxonomias antes de generalizar as Regras de Associação, possibilitando uma maior generalização do conjunto de regras. Entretanto, para viabilizar o uso dessa opção, o conjunto de taxonomias precisa possuir duas ou mais taxonomias sobre o mesmo conjunto de itens. Por exemplo, considerando os itens short, bermuda e camiseta, o conjunto precisaria possuir taxonomias como roupa_esporte(short,bermuda) e roupa_esporte(short,bermuda,camiseta).

Com relação aos conjuntos apresentados nas Seções 6.4.1 e 6.4.2, apenas Tax13 possui duas ou mais taxonomias sobre o mesmo conjunto de itens. Desse modo, foi realizado um experimento no qual o conjunto Tax13 foi submetido ao algoritmo *GART* (utilizando a opção **-sort**) junto com cada um dos conjuntos de Regras de Associação descritos na Seção 6.3. O resultado desse experimento apresentou uma taxa de redução 1% maior do que

as apresentadas na Tabela 6.6 para o conjunto Tax13. Além disso, como as combinações Comb05, Comb11 e Comb17 (Seção 6.4.3) apresentam em sua composição o conjunto de taxonomias Tax13, foram realizados alguns experimentos com essas três combinações, utilizando a opção **-sort** do algoritmo *GART*. Os resultados dos experimentos apresentaram taxas de redução 4% maiores do que as apresentadas na Tabela 6.9 para as combinações Comb05, Comb11 e Comb17. Além disso, a combinação Comb17 ao generalizar o conjunto de regras RuleSet_1mes apresentou uma taxa de redução de 50,11% do volume de regras desse conjunto. Assim, pode-se verificar que a opção **-sort**, implementada para o algoritmo *GART*, proporciona uma maior redução do volume de um conjunto de Regras de Associação.

6.6 Considerações Finais

Para a compreensão dos experimentos apresentados neste capítulo, inicialmente foram descritas a Base de Dados utilizada e as atividades de Pré-processamento e mineração de regras realizadas. Em seguida, no Pós-processamento, foram apresentados os experimentos realizados com a generalização dos conjuntos de Regras de Associação utilizando conjuntos de taxonomias.

Nos experimentos realizados foi verificado que a generalização utilizando conjuntos de taxonomias possibilita a redução do volume dos conjuntos de Regras de Associação. Além disso, durante a realização desses experimentos foi constatado a presença de taxonomias sem classificação nos conjuntos de regras. Para a análise das regras generalizadas com taxonomias sem classificação há a necessidade de uma ferramenta apropriada, caso contrário, a generalização irá gerar regras que apresentam significado incompleto para o usuário. Cabe enfatizar que a generalização utilizando conjuntos de taxonomias sem classificação possibilitou taxas de redução de até 48% do volume de um conjunto de regras. Além disso, nos experimentos utilizando a opção **-sort** do algoritmo *GART*, a combinação Comb17 ao generalizar o conjunto de regras RuleSet_1mes apresentou uma taxa de redução de 50,11% do volume de regras desse conjunto.

No próximo capítulo são apresentadas as conclusões sobre o trabalho realizado e as propostas de trabalhos futuros.

Conclusão

Com os avanços tecnológicos, as organizações adquiriram a capacidade de armazenar informações detalhadas sobre cada transação que efetuam, gerando grandes Bases de Dados. As organizações reconhecem o valor das informações contidas em suas Bases de Dados e a necessidade de extrair conhecimento a partir das mesmas.

Como o uso de procedimentos manuais na obtenção de conhecimento a partir de dados tem se tornado lento, subjetivo e custoso, existe a necessidade de desenvolvimento de novas tecnologias para extrair conhecimento de Bases de Dados, como a Mineração de Dados.

Um dos objetivos do processo de Mineração de Dados é que seus usuários finais possam analisar, compreender e utilizar o conhecimento extraído em um Sistema Inteligente ou utilizá-lo como apoio em processos de tomada de decisão. Entretanto, um problema encontrado no final do processo é que muitos dos algoritmos utilizados geram uma grande quantidade de padrões, dificultando consideravelmente sua análise. Esse problema recebe uma maior ênfase em Regras de Associação, uma vez que essa técnica de Mineração de Dados procura identificar todos os padrões intrínsecos ao conjunto de dados.

Diante desse contexto, este trabalho objetivou generalizar e eliminar Regras de Associação redundantes, facilitando, dessa maneira, a análise das regras obtidas. Para realizar a generalização foram utilizadas taxonomias, com e sem classificação, e a combinação de ambos os tipos de taxonomias.

7.1 Principais Contribuições deste Trabalho

Para alcançar o objetivo proposto neste trabalho, inicialmente foi realizado um estudo das principais técnicas, disponíveis na literatura, que utilizam taxonomias em Regras de Associação. Com esse estudo foi verificada a existência de alguns algoritmos que utilizam taxonomias durante a mineração de Regras de Associação (Srikant & Agrawal 1997a) e uma ferramenta que permite o uso de taxonomias na análise e exploração de Regras de Associação (Liu, Hsu, Chen, & Ma 2000).

Durante a realização desse estudo, foi constatado que os algoritmos que utilizam taxonomias para minerar Regras de Associação, primeiro geram todas as Regras de Associação possíveis (regras com e sem taxonomia) e, em um segundo momento, buscam reduzir o volume de regras geradas utilizando a medida de interesse (*desvio*), apresentada na Seção 3.3.1 do Capítulo 3, para remover as regras que não são interessantes. Como a escolha do valor da medida de interesse é realizada de maneira subjetiva (de acordo com a experiência do usuário), não é garantido que haja uma redução do volume de regras.

Também foi constatado que a ferramenta *Association Interestingness Analysis System* (AIAS), desenvolvida por (Liu, Hsu, Chen, & Ma 2000), possibilita apenas o uso de taxonomias como filtros, identificando/filtrando as regras que possuem uma maior conformidade com as taxonomias.

Diante da problemática apresentada pelos algoritmos, que utilizam taxonomias para minerar Regras de Associação, e das limitações da ferramenta AIAS, neste trabalho foi proposto e implementado um algoritmo que generaliza Regras de Associação durante a etapa de Pós-processamento do conhecimento, e um módulo computacional para análise das regras generalizadas.

O algoritmo proposto e implementado, denominado *GART* (*Generalization of Association Rules using Taxonomies* – Generalização de Regras de Associação usando Taxonomias), utiliza taxonomias para generalizar Regras de Associação na etapa de Pós-processamento do conhecimento extraído. Desse modo, é possível reduzir o volume das regras geradas, facilitando, assim, a análise e o uso dessas.

Já o módulo computacional *RuleE-GAR*, além de utilizar o algoritmo *GART* para generalizar regras, disponibiliza funcionalidades para a análise das Regras de Associação generalizadas. O módulo foi implementado e integrado ao *RuleE* (Paula 2003), que é um Ambiente para Exploração de Regras de Classificação, Regressão e Associação, descrito no Capítulo 5.

Cabe ressaltar que este trabalho faz parte do *Discover* (Baranauskas & Batista 2000;

Rezende, Monard, Batista, Prati, Pugliesi, & Melanda 2004), um projeto em desenvolvimento pelos pesquisadores do LABIC e que tem como objetivo fornecer um ambiente integrado para apoiar as etapas do processo de Extração de Conhecimento de Dados e Textos. Com a proposta e a implementação do algoritmo *GART* e do módulo *RuleE-GAR*, o Ambiente para Exploração de Regras *RuleE*, e por consequência, o projeto *Discover* adquiriram funcionalidades para generalizar Regras de Associação e para analisar as regras generalizadas.

7.2 Resultados dos Experimentos Realizados

Para mostrar que o uso de taxonomias na generalização de um conjunto de Regras de Associação pode reduzir o volume desse conjunto, foram realizados alguns experimentos utilizando o algoritmo *GART*. Os experimentos, utilizando taxonomias com classificação, apresentaram taxas de redução, do volume de um conjunto de Regras de Associação, na faixa de 3,98% a 17,55%. Entretanto, como as taxonomias são construídas pelo usuário, outras taxonomias podem gerar taxas de redução maiores do que as apresentadas nos experimentos, principalmente se as mesmas forem construídas por um especialista no domínio.

Já nos experimentos realizados, utilizando conjuntos de taxonomias sem classificação, as taxas de redução de volume de alguns conjuntos de Regras de Associação apresentaram variação entre 14,12% e 42,97%. Vale lembrar, que o uso do algoritmo *GART* com taxonomias sem classificação ao invés do uso de taxonomias com classificação poderia resultar, por exemplo, em regras do tipo PRODUTOS1 & TANG $\Rightarrow$ ACUCAR.UNIAO. A regra, do modo como é apresentada, não possui significado completo para o usuário, uma vez que é difícil identificar quais produtos fazem parte do PRODUTOS1. Entretanto, a regra pode apresentar significado completo, se a mesma for analisada utilizando a interface de análise de Regras de Associação generalizadas do módulo computacional *RuleE-GAR* (descrito na Seção 5.3.3 do Capítulo 5). Essa interface permite ao usuário verificar os itens originais que deram origem a um item generalizado, desse modo, a mesma pode ser utilizada para verificar os itens que estão contidos no PRODUTOS1, viabilizando, assim, a compreensão da regra.

Já os experimentos, combinando conjuntos de taxonomias, com e sem classificação, apresentaram taxas de redução variando entre 14,61% e 48,44%. Comparando os resultados desses experimentos com os realizados utilizando conjuntos de taxonomias sem classificação pode-se verificar que houve uma influência muito maior dos conjuntos de taxonomias sem classificação, na generalização das regras, do que dos conjuntos de taxono-

nomias com classificação, uma vez que os resultados de ambos os experimentos são muito semelhantes.

Por fim, foram realizados alguns experimentos utilizando a opção **-sort**, implementada neste trabalho para o algoritmo *GART*. Como apresentado na Seção 4.3.3 do Capítulo 4, essa opção ordena (em ordem decrescente) o conjunto de taxonomias antes de generalizar as Regras de Associação, possibilitando uma maior generalização do conjunto de regras. Entretanto, para viabilizar o uso dessa opção, o conjunto de taxonomias precisa possuir duas ou mais taxonomias sobre o mesmo conjunto de itens. Como apenas o conjunto de taxonomias Tax13 (Figura 6.15 do Capítulo 6) possui duas ou mais taxonomias sobre o mesmo conjunto de itens, foram realizados experimentos com esse conjunto e com as combinações Comb05, Comb11 e Comb17 (que apresentam Tax13 em sua composição). Os resultados desses experimentos apresentaram taxas de redução de 1% a 4% maiores do que as obtidas com os experimentos apresentados anteriormente (não utilizando a opção **-sort**). Além disso, cabe ressaltar que a generalização do conjunto de regras RuleSet_1mes utilizando a combinação Comb17 proporcionou uma taxa de redução de 50,11% do volume de regras desse conjunto.

7.3 Trabalhos Futuros

Nesta seção são apresentadas algumas propostas de trabalhos futuros que complementam o desenvolvimento relatado nesta dissertação e podem trazer novas contribuições para o uso de taxonomias na generalização de Regras de Associação:

- Implementação de uma versão do algoritmo *GART* que permita a generalização de ambos os lados das Regras de Associação simultaneamente. A versão atual do algoritmo *GART* permite a generalização de apenas um lado das Regras de Associação (esquerdo ou direito);
- Incorporação de novos métodos de visualização de conjuntos de regras, ao módulo computacional *RulEE-GAR*, para facilitar a montagem das taxonomias. Atualmente, o módulo *RulEE-GAR* permite visualizar conjuntos de regras ordenados por antecedentes ou conseqüentes semelhantes. Uma outra proposta seria o agrupamento das Regras de Associação utilizando algum algoritmo de *clustering* e a exibição das mesmas ordenadas por clusters;

- Realização de experimentos utilizando outros conjuntos de dados para melhor verificar o uso de taxonomias na generalização e remoção de Regras de Associação não interessantes e/ou redundantes.

Referências

- Adamo, J.-M. (2001). *Data Mining for Association Rules and Sequential Patterns*. New York, NY: Springer-Verlag.
- Agrawal, R., T. Imielinski, & A. N. Swami (1993). Mining association rules between sets of items in large databases. Em P. Buneman & S. Jajodia (eds.), *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Washington, D.C., pp. 207–216.
- Agrawal, R. & R. Srikant (1994). Fast algorithms for mining association rules. Em J. B. Bocca, M. Jarke, & C. Zaniolo (eds.), *Proceedings 20th International Conference on Very Large Data Bases, VLDB*, pp. 487–499.
- Axmark, D., M. Widenius, & P. DuBois (2002). Mysql 4.0.4 reference manual. Disponível em: <http://www.mysql.com/doc.html>. Acesso em: 15/08/2003.
- Baesens, B., S. Viaene, & J. Vanthienen (2000). Post-processing of association rules. Em *Proceedings of the Special Workshop on Post-Processing. The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2–8.
- Baranauskas, J. A. & G. E. A. P. A. Batista (2000). O projeto *Discover*: Idéias iniciais (comunicação pessoal).
- Baranauskas, J. A. (2001). Extração automática de conhecimento por múltiplos indutores. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Batista, G. E. A. P. A., A. C. P. L. Carvalho, & M. C. Monard (2000). Applying one-sided selection to unbalanced datasets. Em O. Cairó, L. E. Sucar, & F. J. Cantu (eds.), *Proceedings of the Mexican Congress on Artificial Intelligence MICAI, Lecture Notes in Artificial Intelligence*, pp. 315–325.

- Batista, G. E. A. P. A. (2003). Pré-processamento de dados em aprendizado de máquina supervisionado. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Batista, G. E. & M. C. Monard (2003). Descrição da arquitetura e do projeto do ambiente computacional *Discover Learning Environment* - DLE. Relatório Técnico do ICMC/USP - Número 187. Disponível em: http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel_tec/RT_187.PDF. Acesso em: 15/12/2003.
- Bernardini, F. C. (2002). Combinação de classificadores para melhorar o poder preditivo e descritivo de ensembles. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Breiman, L. (2000). Some infinite theory for predictor ensembles. Relatório Técnico, Statistics Department - UC Berkeley. Disponível em: <http://citeseer.nj.nec.com/breiman00some.html>. Acesso em: 18/10/2002.
- Brin, S., R. Motwani, J. D. Ullman, & S. Tsur (1997). Dynamic itemset counting and implication rules for market basket data. Em J. Peckham (ed.), *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, pp. 255–264.
- Bruha, I. & A. Famili (2000). Postprocessing in machine learning and data mining. *ACM SIGKDD Explorations Newsletter* 2(2), 110–114.
- Clementini, E., P. D. Felice, & K. Koperski (2000). Mining multiple-level spatial association rules for objects with a broad boundary. *Data & Knowledge Engineering* 34(3), 251–270.
- Domingues, M. A. & S. O. Rezende (2004). Descrição de um algoritmo para generalização de regras de associação. Relatório Técnico do ICMC/USP - Número 228.
- Dosualdo, D. G. & S. O. Rezende (2003). Descrição do ambiente computacional DiPER. Relatório Técnico do ICMC/USP - Número 204. Disponível em: http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel_tec/RT_204.zip. Acesso em: 15/12/2003.
- Dosualdo, D. G. (2003). Investigação de regressão no processo de mineração de dados. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Fayyad, U. M., G. Piatetsky-Shapiro, & P. Smyth (1996a). From data mining to knowledge discovery: An overview. Em U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (eds.), *Advances in Knowledge Discovery and Data Mining*, Volume 1, pp. 1–30. Menlo Park, CA: American Association for Artificial Intelligence.

- Fayyad, U. M., G. Piatetsky-Shapiro, & P. Smyth (1996b). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM* 39(11), 27–34.
- Geromini, M. R. (2002). Projeto e desenvolvimento da interface gráfica do sistema *Discover*. Monografia do Exame de Qualificação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Glymour, C., D. Madigan, D. Pregibon, & P. Smyth (1997). Statistical themes and lessons for data mining. *Data Mining and Knowledge Discovery* 1(1), 11–28.
- Gonçalves, L. S. M. (2002). Categorização em text mining. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Han, J. & Y. Fu (1995). Discovery of multiple-level association rules from large databases. Em U. Dayal, P. M. D. Gray, & S. Nishio (eds.), *Proceedings of 21th International Conference on Very Large Data Bases*, pp. 420–431.
- Hipp, J., U. Güntzer, & G. Nakhaeizadeh (2002). Data mining of association rules and the process of knowledge discovery in databases. Em P. Perner (ed.), *Advances in Data Mining 2002, LNAI 2394*, pp. 15–36.
- Hipp, J., A. Myka, R. Wirth, & U. Güntzer (1998). A new algorithm for faster mining of generalized association rules. Em J. M. Zytkow & M. Quafafou (eds.), *Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD '98)*, Nantes, France, pp. 74–82.
- Horst, P. S. (1999). Avaliação do conhecimento adquirido por algoritmos de aprendizado de máquina utilizando exemplos. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Houtsma, M. & A. Swami (1995). Set-oriented mining for association rules in relational databases. Em P. S. Yu & A. L. P. Chen (eds.), *Proceedings of the Eleventh International Conference on Data Engineering*, Taipei, Taiwan, pp. 25–33.
- Imamura, C. Y. (2001). Pré-processamento para extração de conhecimento de bases textuais. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Lavrač, N., P. Flach, & R. Zupan (1999). Rule evaluation measures: A unifying view. Em S. Dzeroski & P. Flach (eds.), *Proceedings of the Ninth International Workshop on Inductive Logic Programming (ILP-99)*, Volume 1634, pp. 174–185. Springer-Verlag. LNAI.

- Lee, H. D. (2000). Seleção e construção de features relevantes para o aprendizado de máquina. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Liu, B., W. Hsu, S. Chen, & Y. Ma (2000). Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems & their Applications* 15(5), 47–55.
- Martins, C. A. (2003). Uma abordagem para pré-processamento de dados textuais em algoritmos de aprendizado. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Matsubara, E. T., C. A. Martins, & M. C. Monard (2003). Pretext: uma ferramenta para pré-processamentos de textos utilizando a abordagem bag-of-words. Relatório Técnico do ICMC/USP - Número 209. Disponível em: http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel_tec/RT_209.zip. Acesso em: 15/12/2003.
- Matsubara, E. T. (2003). Melhorando classificadores a partir de poucos exemplos rotulados aplicados a mineração de textos. Monografia do Exame de Qualificação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Melanda, E. A. & S. O. Rezende (2003). Sintaxe padrão para representar regras de associação. Relatório Técnico do ICMC/USP - Número 206. Disponível em: http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel_tec/rt_206.pdf. Acesso em: 15/12/2003.
- Melanda, E. A. (2002). Pós-processamento de conhecimento de regras de associação. Monografia do Exame de Qualificação de Doutorado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Mitchell, T. M. (1998). *Machine Learning*. Boston, USA: McGraw-Hill.
- Monard, M. C. & J. A. Baranauskas (2003a). Conceitos sobre aprendizado de máquina. Em S. O. Rezende (ed.), *Sistemas Inteligentes: Fundamentos e Aplicações*, Volume 1, pp. 89–114. Barueri, SP: Editora Manole.
- Monard, M. C. & J. A. Baranauskas (2003b). Indução de regras e árvores de decisão. Em S. O. Rezende (ed.), *Sistemas Inteligentes: Fundamentos e Aplicações*, Volume 1, pp. 115–139. Barueri, SP: Editora Manole.
- Nagai, W. A. (2000). Avaliação do conhecimento extraído de problemas de regressão. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.

- Park, J. S., M.-S. Chen, & P. S. Yu (1997). Using a hash-based method with transaction trimming for mining association rules. *IEEE Transactions on Knowledge and Data Engineering* 9(5), 813–825.
- Paula, M. F. (2003). Ambiente para exploração de regras. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Pei, J., J. Han, & R. Mao (2000). CLOSET: An efficient algorithm for mining frequent closed itemsets. Em D. Gunopulos & R. Rastogi (eds.), *Proceedings of ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp. 21–30.
- Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. Em G. Piatetsky-Shapiro & W. J. Frawley (eds.), *Proceedings of Knowledge Discovery in Databases*, pp. 229–248. MIT.
- Piatetsky-Shapiro, G. (1999). The data-mining industry coming of ages. *IEEE Intelligent Systems* 14(6), 32–34.
- Prati, R. C., J. A. Baranauskas, & M. C. Monard (2001). Uma proposta de unificação da linguagem de representação de conceitos de algoritmos de aprendizado de máquina simbólicos. Relatório Técnico do ICMC/USP - Número 137. Disponível em: http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel_tec/RT_137.ps.zip. Acesso em: 15/12/2003.
- Prati, R. C. (2003). O framework de integração do sistema *Discover*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.
- Pugliesi, J. B., D. G. Dosualdo, & S. O. Rezende (2003). Sintaxe padrão para representar regras de regressão no *Discover*. Relatório Técnico do ICMC/USP - Número 193. Disponível em: http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel_tec/RT_193.zip. Acesso em: 15/12/2003.
- Pugliesi, J. B. & S. O. Rezende (2003). Medidas de erro para regras de regressão. Em R. de Oliveira Anido & P. C. Masiero (eds.), *Anais do XXIII Congresso da Sociedade Brasileira de Computação – IV Encontro Nacional de Inteligência Artificial (ENIA 2003)*, Volume VII, pp. 517–526. SBC – Campinas.
- Pugliesi, J. B. (2001). O pós-processamento em extração de conhecimento de bases de dados. Monografia do Exame de Qualificação de Doutorado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP - Brasil.

- Rezende, S. O., M. C. Monard, G. E. A. P. A. Batista, R. C. Prati, J. B. Pugliesi, & E. A. Melanda (2004). Ambiente *Discover*: Sintaxes padrão de dados e regras. Relatório Técnico do ICMC/USP. Prelo.
- Rezende, S. O., J. B. Pugliesi, E. A. Melanda, & M. F. Paula (2003). Mineração de dados. Em S. O. Rezende (ed.), *Sistemas Inteligentes: Fundamentos e Aplicações*, Volume 1, pp. 307–335. Barueri, SP: Editora Manole.
- Semenova, T., M. Hegland, W. Graco, & G. Williams (2001). Effectiveness of mining association rules for identifying trends in large health databases. Em *Workshop on Integrating Data Mining and Knowledge Management. ICDM'01: The 2001 IEEE International Conference on Data Mining*. Disponível em <http://cui.unige.ch/~hilarario/icdm-01/DM-KM-Final/Semenova.pdf>. Acesso em: 11/12/2002.
- Silberschatz, A. & A. Tuzhilin (1995). On subjective measures of interestingness in knowledge discovery. Em U. M. Fayyad & R. Uthurusamy (eds.), *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, pp. 275–281.
- Srikant, R. & R. Agrawal (1997a). Mining generalized association rules. *Future Generation Computer Systems* 13(2–3), 161–180.
- Srikant, R. & R. Agrawal (1997b). Mining generalized association rules. Relatório Técnico, IBM Research Division. Disponível em: http://www.almaden.ibm.com/cs/people/srikant/papers/vldb95_rj.pdf. Acesso em: 18/10/2002.
- Srikant, R. (2001). Association rules: Past, present and future. ICCS 2001 International Workshop on Concept Lattice-based theory, methods and tools for Knowledge Discovery in Databases. Invited Talk. Disponível em: <http://www.almaden.ibm.com/cs/people/srikant/talks/assoc.pdf>. Acesso em: 19/09/2002.
- Till, D. (1996). *Teach Yourself Perl 5 in 21 Days*. Indianapolis, USA: Sams Publishing.
- Webb, G. I. (1995). Opus: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research* 3, 431–465.
- Weber, I. (1998). On pruning strategies for discovery of generalized and quantitative association rules. Em I. L. Bing, W. Hsu, & W. Ke (eds.), *Proceedings Knowledge Discovery and Data Mining Workshop (Pricai'98)*. Disponível em: <http://citeseer.nj.nec.com/25197.html>. Acesso em: 10/12/2002.
- Weiss, S. M. & N. Indurkha (1998). *Predictive Data Mining: A Practical Guide*. San Francisco, CA: Morgan Kaufmann.

- Zaki, M. & C. Hsiao (2002). Charm: An efficient algorithm for closed association rule mining. Em R. Grossman, J. Han, V. Kumar, H. Mannila, & R. Motwani (eds.), *2nd SIAM International Conference on Data Mining*. Disponível em: <http://www.siam.org/meetings/sdm02/proceedings/sdm02-27.pdf>. Acesso em: 05/12/2002.
- Zheng, Z., R. Kohavi, & L. Mason (2001). Real world performance of association rule algorithms. Em F. Provost & R. Srikant (eds.), *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 401–406. ACM Press.