UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2569

***AprioriGen*: Uma extensão do algoritmo *Apriori* para obtenção de Regras de Associação Generalizadas**

André Roberto Martins

Veronica de Oliveira Carvalho

Solange Oliveira Rezende

Nº 308

RELATÓRIOS TÉCNICOS

São Carlos - SP

Agosto/2007

AprioriGen: Uma extensão do algoritmo *Apriori* para obtenção de Regras de Associação Generalizadas[1]

André Roberto Martins

Veronica de Oliveira Carvalho

Solange Oliveira Rezende

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística
C.P. 668, 13560-970 - São Carlos, SP - Brasil
e-mail: andrerm@grad.icmc.usp.br, {solange, veronica}@icmc.usp.br

---

Resumo: O processo de Mineração de Dados tem como objetivo encontrar conhecimento a partir de um conjunto de dados para ser utilizado como auxílio em um processo de tomada de decisão. As Regras de Associação estão entre as técnicas de mineração de dados que vêm recebendo destaque nos últimos anos. Essa técnica possui algumas variações, dentre as quais, uma que utiliza taxonomias na etapa de extração de padrões para obter Regras de Associação Generalizadas. Existem diversos algoritmos que extraem esse tipo de regra, embora não se tenha encontrado uma implementação disponível dos mesmos. A fim de utilizar taxonomias durante a etapa de extração de padrões, visando a obtenção de Regras de Associação Generalizadas, este trabalho tem como objetivo modificar uma implementação do algoritmo *Apriori* a fim de obter uma versão semelhante aos algoritmos de Regras de Associação Generalizadas existentes.

Palavras Chaves: Mineração de Dados, Regras de Associação, *Apriori*, Generalização de Regras, Taxonomias.

Agosto/2007

[1]Trabalho realizado com apoio da FAPESP.

Sumário

Sumário i

Lista de Figuras ii

Lista de Tabelas iii

Lista de Algoritmos iv

1 Introdução 1

2 Regras de Associação 3

3 O Algoritmo *Apriori* 6

3.1 Geração de *itemsets* candidatos 7

3.1.1 Função apriori-gen 7

3.1.2 Função subset 9

3.2 Algoritmo Simples para Gerar Regras de Associação 11

4 Algoritmos que utilizam Taxonomias 12

4.1 Notações e Definições 14

4.2 O Algoritmo Básico 15

4.3 O Algoritmo Cumulate 16

5 O Algoritmo *AprioriGen* 16

5.1 Algoritmo *AprioriGen* 18

5.2 Exemplo de Funcionamento do Algoritmo *AprioriGen* 19

6 Implementação do Algoritmo *AprioriGen* 22

6.1 Arquivos de Entrada e Saída 24

6.2 Modo de Execução 25

7 Experimentos Realizados 26

7.1 Descrição das Bases de Dados 26

7.2 Mineração das Regras de Associação Generalizadas 29

7.2.1 Experimentos com a Base de Dados 1 29

7.2.2 Experimentos com a Base de Dados 2 31

7.3 Comparação e Avaliação dos Resultados 35

7.4 Comparação com o Estado da Arte 36

7.4.1 Pré-processamento 37

7.4.2 Extração de Padrões 37

7.4.3 Pós-processamento 38

8 Considerações Finais **42**

A *AprioriGen* no Pré-processamento **44**

A.1 *AprioriGen* versão 2 44

A.2 Experimentos Realizados 46

A.2.1 Experimento com a Base de Dados 1 46

A.2.2 Experimento com a Base de Dados 2 47

A.3 Análise dos Resultados 49

Referências Bibliográficas **50**

Lista de Figuras

1 *Itemsets* candidatos armazenados em uma árvore-*hash*. 9

2 Exemplo de uma taxonomia para vestuário. 13

3 Regras de Associação Generalizadas obtidas na etapa de Extração de Padrões usando uma taxonomia. 17

4 Conjunto de Taxonomias $\mathcal{T}$. 20

5 Exemplo de arquivo de transações (dados.data). 24

6 Exemplo do arquivo de taxonomias da Figura 2 (taxonomia.tax). 24

7 Exemplo de um conjunto de saída de regras de associação (saida.rules). 25

8 Conjunto de dados da primeira base de dados (base1.data). 26

9 Conjunto de taxonomias da primeira base de dados (base1tax.tax). . . . 27

10 Amostra do conjunto de dados da segunda base de dados (base2.data). . 27

11 Conjunto de taxonomias da segunda base de dados (base2tax.tax). . . . 28

12 Conjunto de regras obtidas da primeira base de dados utilizando o algoritmo *AprioriGen* com suporte mínimo igual a 30% e confiança mínima igual a 60% (saidabase1grupo1exp2.rules). 30

13 Conjunto de regras obtidas da primeira base de dados utilizando o algoritmo *Apriori* com suporte mínimo igual a 10% e confiança mínima igual a 50% (saidabase1grupo2exp3.rules). 30

14 Conjunto de regras obtidas da primeira base de dados utilizando o algoritmo *AprioriGen* com suporte mínimo igual a 10% e confiança mínima igual a 50% (saidabase1grupo2exp4.rules). 31

15 Conjunto de regras obtidas da segunda base de dados utilizando o algoritmo *AprioriGen* com suporte mínimo igual a 15% e confiança mínima igual a 15% (saidabase2grupo1exp6.rules). 33

16 Conjunto de regras obtidas da segunda base de dados utilizando o algoritmo *Apriori* com suporte mínimo igual a 4% e confiança mínima igual a 5% (saidabase2grupo2exp7.rules). 34
17 Amostra do conjunto de regras obtidas da segunda base de dados utilizando o algoritmo *AprioriGen* com suporte mínimo igual a 4% e confiança mínima igual a 5% (saidabase2grupo2exp8.rules). 35
18 Visão geral da abordagem de pós-processamento de regras de associação (APRA). 39
19 Conjunto de regras generalizadas obtidas pela *APRA* utilizando a opção lhs-sup-0. 41
20 Conjunto de regras generalizadas obtidas pela *APRA* utilizando a opção rhs-sup-0. 41
21 Conjunto de regras generalizadas obtidas pela *APRA* utilizando a opção lrhs-sup-0. 41
22 Listagem de contribuição de itens obtida pela *APRA* para as configurações: lhs-sup-0, rhs-sup-0 e lrhs-sup-0. 41
23 Conjunto de regras obtidas da primeira base de dados utilizando o algoritmo *AprioriGen* v.2 (saidabase1v2exp9.rules). 47
24 Conjunto de regras obtidas utilizando o algoritmo *AprioriGen* v.2 com suporte mínimo igual a 10% e confiança mínima igual a 50% (saidabase1v2exp9.rules). 47
25 Conjunto de regras obtidas da segunda base de dados utilizando o algoritmo *AprioriGen* v.2 (saidabase2v2exp11.rules). 48
26 Amostra do conjunto de regras obtidas utilizando o algoritmo *AprioriGen* v.2 com suporte mínimo igual a 4% e confiança mínima igual a 5% (saidabase2v2exp12.rules). 49

Lista de Tabelas

1 Relação de itens comprados por transação do Exemplo 1. 5
2 *Itemsets* freqüentes do Exemplo 1. 5
3 Conjunto *Resposta* contendo os *itemsets* freqüentes. 11
4 Relação de itens comprados por transação. 20
5 Relação de transações após execução das linhas 1 a 8 do Algoritmo 4. . 20
6 Conjunto *Resposta* contendo os *itemsets* freqüentes. 21
7 Regras de Associação Generalizadas apresentadas em Srikant & Agrawal (1997a). 35

Lista de Algoritmos

1 *Apriori*. 6
2 Função apriori-gen. 8
3 Gera regras de associação. 11
4 Algoritmo *AprioriGen*. 18
5 *AprioriGen* versão 2. 45

1 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, entre outros fatores, 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 e têm investido na aquisição e desenvolvimento de ferramentas de análise de dados.

Durante anos, métodos predominantemente manuais têm sido utilizados para transformar dados em conhecimento. Entretanto, o uso desses métodos tem se tornado dispendioso (tanto em termos financeiros quanto de tempo), subjetivo e inviável, quando aplicados a grandes Bases de Dados.

Devido às limitações dos métodos manuais tradicionais, tornou-se necessário o desenvolvimento de processos de análise automática dos dados, 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 et al., 1996).

O processo de Mineração de Dados pode ser dividido em um ciclo composto de três grandes etapas: Pré-processamento dos dados, Extração de Padrões e Pós-processamento do conhecimento. Na etapa de Pré-processamento dos dados, procedimentos de integração, transformação, limpeza, entre outros, são realizados para que os dados possam ser utilizados adequadamente no processo de extração do conhecimento. Essa etapa é importante uma vez que influencia a qualidade do conhecimento extraído. Já a etapa de Extração de Padrões é responsável por identificar os padrões embutidos nos dados. Para tanto, podem-se utilizar diversos algoritmos de aprendizado de máquina. A etapa de Pós-processamento é responsável pela avaliação, interpretação, explanação, filtragem e integração do conhecimento obtido. Essa etapa possui grande importância para o processo de mineração, já que os usuários estão interessados em utilizar o conhecimento extraído para apoiar suas decisões.

As diversas técnicas de Mineração de Dados podem ser agrupadas em duas categorias: *Descritivas* e *Preditivas*. A primeira descreve o conjunto de exemplos (também chamados de casos ou dados) de uma maneira concisa e sumarizada, apresentando algumas propriedades gerais dos dados; a segunda constrói hipóteses fazendo inferências acerca dos exemplos disponíveis na tentativa de prever o comportamento de novos casos.

Uma das técnicas Descritivas de Mineração de Dados que vem se destacando nos últimos anos é a de Regra de Associação (Rezende et al., 2003). Em linhas gerais, 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). Além da utilização dessa técnica no desenvolvimento de pesquisas acadêmicas, os resultados da mesma têm sido utilizados pelas organizações no comércio, em contratos de seguro, na saúde, no geoprocessamento, na biologia molecular, entre outras áreas (Liu et al., 2000; Clementini et al., 2000; Semenova et al., 2001).

As Regras de Associação são obtidas a partir de uma base de dados de transações de itens permitindo a geração de um conjunto de Regras de Associação. Com o objetivo de obter um conjunto de Regras de Associação mais geral, a fim de facilitar a interpretação das mesmas, taxonomias vêm sendo utilizadas freqüentemente nos últimos anos. As taxonomias refletem uma visão coletiva ou individual de como os itens (ou seja, os elementos do domínio de aplicação) podem ser hierarquicamente classificados (Adamo, 2001). Em função dessa hierarquia é possível se obter regras mais gerais que representem um conjunto de itens (por exemplo, o conjunto formado por todos os tipos de café). Embora existam atualmente diversos algoritmos que utilizam taxonomias na etapa de Extração de Padrões, não foram encontradas implementações disponíveis dos mesmos.

Diante desse contexto, neste relatório é descrito o algoritmo *AprioriGen*, o qual tem por objetivo gerar Regras de Associação Generalizadas. Assim, este trabalho visa modificar uma implementação do algoritmo *Apriori* a fim de obter uma nova versão, semelhante aos algoritmos de Regras de Associação Generalizadas existentes.

Este relatório está organizado da seguinte maneira: na Seção 2 é apresentada a técnica de Mineração de Regras de Associação. Na Seção 3 é apresentado o Algoritmo *Apriori* e seu funcionamento. Na Seção 4 são apresentadas as características dos Algoritmos Básico e Cumulate em que o *AprioriGen* se baseia. Na Seção 5 é descrito o Algoritmo *AprioriGen*. As informações sobre a implementação do algoritmo *AprioriGen* são descritas na Seção 6. Na Seção 7 são descritos os experimentos realizados com o algoritmo *AprioriGen*. Na Seção 8 são apresentadas algumas considerações finais sobre este relatório. Por fim, no Apêndice A é apresentada uma segunda versão do algoritmo *AprioriGen*.

2 Regras de Associação

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 compram o produto Q também adquirem, na mesma ocasião, o produto W. Em outras palavras, pode-se dizer que esse regra apresenta confiabilidade de 80%.

O formato de uma Regra de Associação pode ser representado como uma implicação $LHS \Rightarrow RHS$, 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. As Regras de Associação podem ser definidas como descrito a seguir (Agrawal & Srikant, 1994):

> Seja D uma Base de Dados composta por um conjunto de itens $A = \{a_1, ..., a_m\}$ ordenados lexicograficamente e por um conjunto de transações $T = \{t_1, ..., t_n\}$, na qual cada transação $t_i \in T$ é composta por um conjunto de itens (*itemset*) 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 $LHS \cap RHS = \varnothing$. 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 D ocorre $LHS \cup RHS$.

O valor do suporte mede a força da associação entre LHS e RHS e não relaciona possíveis dependências de RHS com LHS. Por outro lado, a confiança mede a força da implicação lógica descrita pela regra (Zhang & Zhang, 2002). 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 freqüência 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:

$$sup(X) = \frac{n(X)}{N}, \tag{1}$$

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

$$sup(LHS \Rightarrow RHS) = sup(LHS \cup RHS) = \frac{n(LHS \cup RHS)}{N}, \qquad (2)$$

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 consideradas.

Confiança: indica a freqüência 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:

$$conf(LHS \Rightarrow RHS) = \frac{sup(LHS \cup RHS)}{sup(LHS)} = \frac{n(LHS \cup RHS)}{n(LHS)}, \qquad (3)$$

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

Usualmente, valores de suporte e confiança mínimos são definidos pelo usuário antes da mineração das Regras de Associação. Em geral, a definição de altos valores para esses parâmetros gera apenas regras triviais; já a definição de baixos valores gera, em geral, um grande volume de conhecimento no formato de regras, dificultando a análise do usuário na etapa de Pós-processamento. Uma maneira de superar as dificuldades na análise dessas regras no Pós-processamento é utilizar-se de algoritmos que possibilitem usar taxonomias já durante a etapa de Extração de Padrões.

O problema de obtenção de regras de associação é decomposto em dois sub-problemas (passos) (Agrawal et al., 1993):

1. Encontrar todos os *k-itemsets* (conjunto de k itens) 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;

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 $\{a, b, c, d\}$ e um subconjunto de *itemsets* freqüentes $\{a, b\}$, por exemplo, pode-se gerar uma regra do tipo $ab \Rightarrow cd$, desde que $conf(ab \Rightarrow cd) \geq$ **conf-min**, em que, $conf(ab \Rightarrow cd) = sup(a, b, c, d)/sup(a, b)$.

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 $A = \{$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 1.*

Tabela 1: 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 1 utilizando os dois passos descritos anteriormente.*

1. *Encontrar todos os k-itemsets, a partir da Tabela 1, que possuam suporte maior ou igual a* **sup-min** *(itemsets freqüentes). Na Tabela 2 são apresentados todos os k-itemsets freqüentes.*

Tabela 2: *Itemsets* freqüentes.

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

2. *Com os k-itemsets freqüentes obtidos no passo 1, em que $k \geq 2$, gerar todas as regras de associação contidas na Tabela 1, que nesse exemplo são:*

   regra 1: *tênis $\Rightarrow$ camiseta,*

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

A obtenção de *itemsets* freqüentes para gerar Regras de Associação pode ser realizada utilizando diversos algoritmos. Um dos mais importantes algoritmos para gerar *itemsets* freqüentes, o algoritmo *Apriori*, é descrito a seguir.

3 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* (Agrawal & Srikant, 1994).

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

Assegura: Um conjunto *Resposta* contendo todos os *k-itemsets* freqüentes.

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

Inicialmente o algoritmo 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.

A seguir, é apresentado a geração de *itemsets* candidatos no Algoritmo *Apriori* com as funções ***apriori-gen*** e ***subset***.

3.1 Geração de *itemsets* candidatos

Como explicado no início desta seção, o Algoritmo *Apriori* é utilizado para encontrar todos os *k-itemsets* freqüentes contidos em uma base de dados. Assim, após obter um conjunto de *k-itemsets* freqüentes com o Algoritmo *Apriori*, este é submetido a um outro algoritmo que gera as regras de associação a partir de *itemsets* freqüentes obtidos de uma base de dados. Um algoritmo simples proposto por Agrawal & Srikant (1994) para gerar regras de associação a partir dos *itemsets* freqüentes é apresentado na Subseção 3.2.

A fim de poder gerar os *k-itemsets* candidatos para se obter os *itemsets* freqüentes, são apresentados os algoritmos das funções ***apriori-gen*** e ***subset*** responsáveis pela geração de *itemsets* candidatos do Algoritmo *Apriori*.

3.1.1 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 poda (linhas 5 a 11 do Algoritmo 2) é garantida pela propriedade *downward closure* de suporte de um *itemset*.

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

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

Assegura: Um conjunto de *itemsets* candidatos C_k.

1: **insert into** C_k
2: **select** $p.item_1, p.item_2, \ldots, p.item_{k-1}, q.item_{k-1}$ **from** $L_{k-1}p, L_{k-1}q$
3: **where** $p.item_1 = q.item_1, \ldots, p.item_{k-2} = q.item_{k-2}$,
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: **para todo** *itemset* $c \in C_k$ **faça**
6: **para todo** $(k-1)$-subconjuntos s de c **faça**
7: **se** $(s \notin L_{k-1})$ **então**
8: **delete** c de C_k;
9: **fim-se**
10: **fim-para**
11: **fim-para**

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*$\}\}$.

3.1.2 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 raíz, 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 raíz, 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 1 (neste exemplo a árvore possui apenas um nível) e uma transação* $t_i = \{$*camiseta,meia,tênis*$\}$*, a função subset irá encontrar todos os itemsets 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*$\}$*.*

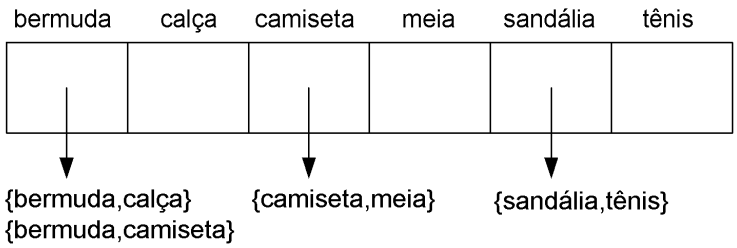

Figura 1: *Itemsets* candidatos armazenados em uma árvore-*hash*.

Para finalizar esta seção, no Exemplo 4 é apresentado o funcionamento do algoritmo *Apriori* aplicado a um pequeno conjunto de transações.

Exemplo 4 *Seja D uma base de dados que contém um conjunto de itens* $A = \{$*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 1 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 1, 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.

Percorre a Base D $\Rightarrow$

C_1

Itemset	***Suporte***
{*calça*}	50%
{*camiseta*}	50%
{*tênis*}	75%
{*bermuda*}	25%
{*sandália*}	25%

L_1

Itemset	***Suporte***
{*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

Itemset
{*calça,camiseta*}
{*calça,tênis*}
{*camiseta,tênis*}

Percorre a Base D $\Rightarrow$

C_2

Itemset	***Suporte***
{*calça,camiseta*}	25%
{*calça,tênis*}	25%
{*camiseta,tênis*}	50%

L_2

Itemset	***Suporte***
{*camiseta,tênis*}	50%

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

C_3

Itemset
$\{\emptyset\}$

Percorre a Base D $\Rightarrow$

L_3

Itemset
$\{\emptyset\}$

Ao término de sua execução é gerado o conjunto Resposta (Tabela 3) pela união de todos os conjuntos L_k que contêm os k-itemsets freqüentes (linha 12 do Algoritmo 1).

O conjunto Resposta é utilizado como entrada para algum algoritmo que gera regras de associação, como o algoritmo apresentado na Seção 3.2.

Tabela 3: Conjunto *Resposta* contendo os *itemsets* freqüentes.

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

3.2 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 (Agrawal & Srikant, 1994).

Requisito: Um conjunto *Resposta* contendo todos os *k-itemsets* freqüentes, com $k \geq 2$.
Assegura: Um conjunto de regras de associação.

1: **para todo** (*k-itemset* freqüente $l_k, k \geq 2$) **faça**
2: **Call** genrules (l_k, l_k);
3: **fim-para**
//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: $A := \{(m-1) - itemsets\ a_{m-1} \mid a_{m-1} \subset a_m\}$;
6: **para todo** $(a_{m-1} \in A)$ **faça**
7: $conf := sup(l_k)/sup(a_{m-1})$;
8: **se** $(conf \geq$ **conf-min**$)$ **então**
9: **Output:** regra $a_{m-1} \Rightarrow (l_k - a_{m-1})$, com *confiança* $= conf$ e *suporte* $= sup(l_k)$;
10: **se** $(m - 1 > 1)$ **então**
11: **Call** genrules (l_k, a_{m-1}); //Gera regras com subconjuntos de a_{m-1} como antecedente
12: **fim-se**
13: **fim-se**
14: **fim-para**

O algoritmo deve ser executado para todos os *k-itemsets* freqüentes para $k \geq 2$. 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 3 do Exemplo 4.

Exemplo 5 *Seja* $resposta_4 = \{camiseta, tênis\}$*, o 2-itemset freqüente contido no conjunto Resposta apresentado na Tabela 3. 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):*

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

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,*

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

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

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

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

4 Algoritmos que utilizam Taxonomias

As taxonomias refletem uma caracterização coletiva ou individual de como os itens podem ser hierarquicamente classificados (Adamo, 2001). Além disso, as informações das taxonomias possibilitam a definição de medidas que podem ser utilizadas para remover regras não interessantes segundo algum critério. Na Figura 2 é apresentado um 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.

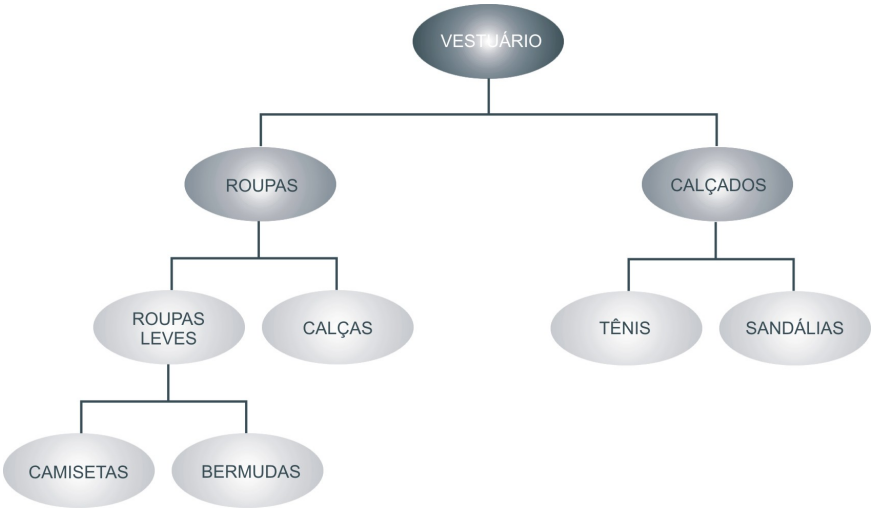

Figura 2: Exemplo de uma taxonomia para vestuário.

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.

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 (Srikant & Agrawal, 1997b). Por exemplo, usando a taxonomia apresentada na Figura 2, pode-se inferir uma regra em que pessoas que compram Roupas Leves também compram Sandálias, do fato de pessoas comprarem Camisetas com Sandálias e Bermudas com Sandálias, sem necessidade de utilizar as regras que indicam que pessoas que compram Camisetas também compram Sandálias e que pessoas que compram Bermudas também compram 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 et al., 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;
- 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, serão apresentadas algumas notações e definições importantes sobre taxonomias e Regras de Associação Generalizadas para a compreensão dos algoritmos.

4.1 Notações e Definições

Nesta seção são apresentadas algumas notações e definições baseadas nos trabalhos de Srikant & Agrawal (1997b) 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 taxonomias. 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, 1997b):

> Seja D uma base de dados composta por um conjunto de itens $A = \{a_1, ..., a_m\}$ ordenados lexicograficamente e por um conjunto de transações $T = \{t_1, ..., t_n\}$, 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 $LHS \subset A$, $RHS \subset A$, $LHS \cap RHS = \varnothing$ 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 $LHS \cup RHS$.

Na próxima seção são descritos os algoritmos Básico e Cumulate propostos por Srikant & Agrawal (1995) que geram *itemsets* freqüentes com taxonomias, possibilitando extrair Regras de Associação Generalizadas durante a etapa de Extração de Padrões.

4.2 O Algoritmo Básico

Em Regras de Associação que usam taxonomias, o problema de verificar se uma transação $t_i \in T$ suporta um *itemset* $X \in A$, 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.

Utilizando a mesma notação de Regras de Associação, o Algoritmo Básico (Srikant & Agrawal, 1995) torna essa tarefa mais simples, uma vez que adiciona a T, conjunto de transações, todos os ancestrais de cada item presente nas transações t_i que pertencem a T. Esse conjunto de transações estendidas é chamado de T'. Assim, T suporta X se, e somente se, T' um superconjunto de X.

Dessa maneira, os *itemsets* freqüentes usando taxonomias são gerados pela aplicação de qualquer algoritmo que gere *itemsets* freqüentes com o 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.

4.3 O Algoritmo Cumulate

O Algoritmo Cumulate é uma versão melhorada do Algoritmo Básico (Srikant & Agrawal, 1995). Entre as melhorias acrescentadas estão:

1. **Filtro dos ancestrais adicionados às 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.

2. **Pré-cálculo dos ancestrais.** Ao invés de calcular os ancestrais de cada item para cada execução do algoritmo, pode-se pré-calcular os ancestrais de cada item e utilizá-los durante as execuções do algoritmo.

3. **Remoção de *itemsets* que contenham apenas um item e seu ancestral.** Essa melhoria é realizada removendo os 2-*itemset* candidatos (*itemsets* candidatos de tamanho dois) constituídos de um item e seu ancestral. A remoção de *itemsets* candidatos de tamanho dois também assegura que em passos subseqüentes do algoritmo nunca serão gerados *itemsets* candidatos que contenham um item e seu ancestral.

A seguir, são apresentadas as modificações do algoritmo *Apriori*, a fim de obter uma nova versão denominada *AprioriGen*, baseada nos algoritmos Básico e Cumulate.

5 O Algoritmo *AprioriGen*

As Regras de Associação Generalizadas podem ser obtidas utilizando taxonomias já durante a etapa de Extração de Padrões, conforme é ilustrado na Figura 3.

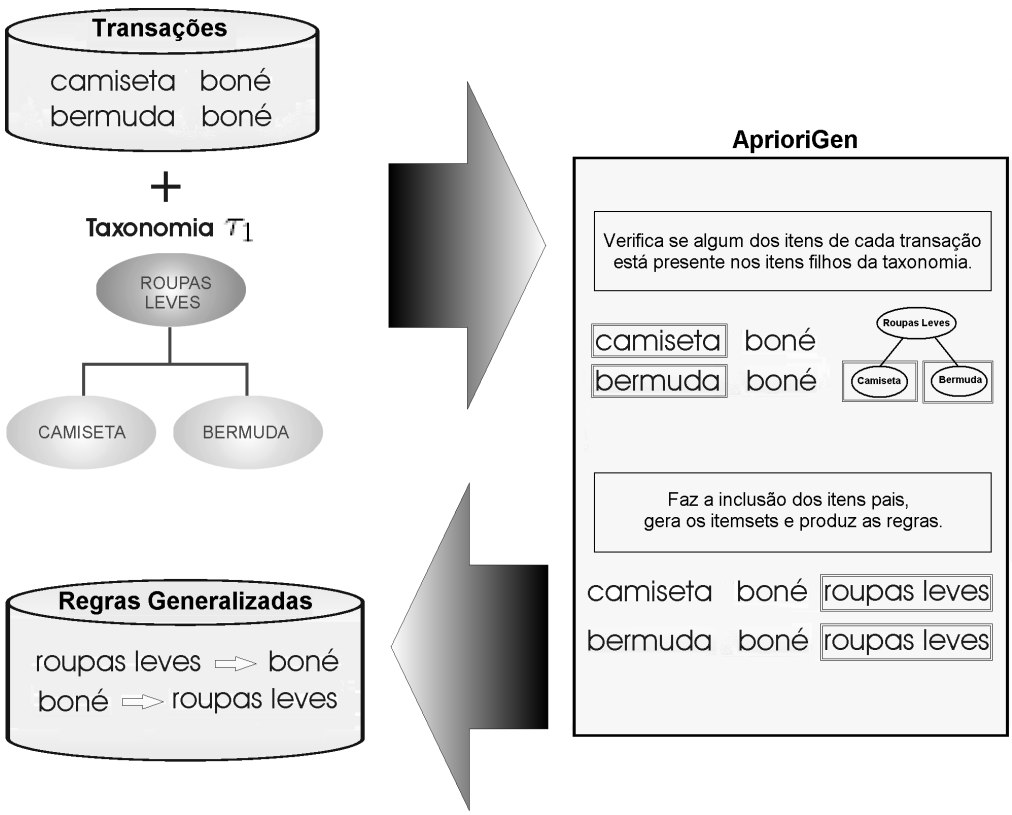

Figura 3: Regras de Associação Generalizadas obtidas na etapa de Extração de Padrões usando uma taxonomia.

A fim de gerar Regras de Associação Generalizadas foi desenvolvido o algoritmo *AprioriGen* através de modificações do algoritmo *Apriori*. Na Figura 3 são ilustradas as transações da base de dados, assim como a taxonomia roupas leves (Taxonomia τ_1 na Figura 3), que são as entradas para o algoritmo *AprioriGen*. O primeiro procedimento do algoritmo *AprioriGen* é a inclusão do item roupas leves nas transações que ocorrem os itens camiseta e bermuda; portanto, nas duas transações desse exemplo foi adicionado o item roupas leves. Esse procedimento corresponde ao algoritmo Básico conforme descrito na Subseção 4.2. Após esse procedimento de inclusão dos itens ancestrais dos itens da transação, são gerados os *itemsets* freqüentes, sendo depois extraídas as regras roupas leves $\Rightarrow$ boné e boné $\Rightarrow$ roupas leves, em função do valor de suporte e confiança mínimo. Através de testes, foi verificado que, durante a geração da regra a partir dos *itemsets* freqüentes, esses *itemsets* podem gerar regras como, por exemplo, roupas leves $\Rightarrow$ camiseta ou camiseta $\Rightarrow$ roupas leves a partir do *itemset* $\{roupas\ leves, camiseta\}$, as quais não fazem sentido ao usuário, pelo fato de um item ser o ancestral e o outro ser o descendente. Desse modo, foi acrescentada uma verificação a fim de desconsiderar *itemsets* freqüentes de tamanho igual a 2 em que um item é ancestral do outro. Esse verificação corresponde à melhoria número 3 do algoritmo Cumulate conforme descrito na Subseção 4.3.

5.1 Algoritmo *AprioriGen*

O algoritmo *AprioriGen* é descrito no Algoritmo 4, onde as linhas sublinhadas indicam os locais alterados do algoritmo *Apriori*. É utilizada a notação L_k para representar o conjunto de *k-itemsets* freqüentes; C_k para representar o conjunto de *k-itemsets* candidatos; Tax para representar o conjunto de taxonomias.

Algoritmo 4 Algoritmo *AprioriGen*.

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

Assegura: Um conjunto *Resposta* contendo todos os *k-itemsets* freqüentes.

1: **para todo** (transações $t \in T$) **faça**
2: **para todo** (itens $a \in t$) **faça**
3: **se** item a possui item pai $b \in Tax$ **então**
4: $t := t + b+$ todos ancestrais de b;
5: remove itens repetidos de t;
6: **fim-se**
7: **fim-para**
8: **fim-para**
9: $L_1 := \{$*1-itemsets* freqüente$\}$;
10: **para** $(k := 2; L_{k-1} \neq \varnothing; k++)$ **faça**
11: $C_k :=$ apriori-gen(L_{k-1}); //Gera novos conjuntos candidatos
12: **para todo** (transações $t \in T$) **faça**
13: $C_t :=$ subset(C_k,t); //Conjuntos candidatos contidos em t
14: **para todo** candidatos $c \in C_t$ **faça**
15: c.count++;
16: **fim-para**
17: **fim-para**
18: $L_k := \{c \in C_k \mid c.count \geq$ **sup-min**$\}$;
19: **se** $k = 2$ **então**
20: **para todo** (itemset $x \in L_{k=2}$) **faça**
21: **se** $item_1 \in x$ for parente de $item_2 \in x$ OU $item_2 \in x$ for parente de $item_1 \in x$ **então**
22: retirar itemset x do conjunto $L_{k=2}$;
23: **fim-se**
24: **fim-para**
25: **fim-se**
26: **fim-para**
27: Resposta $:= \bigcup_k L_k$

Diferente do Algoritmo *Apriori* descrito anteriormente na Seção 3, inicialmente o Algoritmo *AprioriGen* percorre cada item a de cada transação t do conjunto de transações T da base de dados D. Se o item a for um item específico (item filho) de um item geral b (item pai) em alguma taxonomia no conjunto de taxonomias Tax, então é adicionado na transação o item geral b (item pai) e todos os itens gerais do conjunto de

taxonomias que possuem parentesco com o item geral b (item pai) nos níveis acima da hierarquia. Caso contrário, se o item a não possuir nenhum item geral b (item pai) parte para o próximo item ou transação. Essa primeira parte corresponde ao procedimento do Algoritmo Básico (Srikant & Agrawal, 1995), descrito na Seção 4.2. Depois de percorrer todas as transações da base de dados adicionando os itens gerais (linhas 1 a 8 do Algoritmo 4), a segunda parte do algoritmo procede sua execução exatamente como o Algoritmo *Apriori* descrito anteriormente. Assim, o algoritmo 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 11 do Algoritmo 4) o conjunto de *itemsets* freqüentes L_{k-1} obtido no passo (k - 1) é utilizado para gerar o conjunto de 1-*itemsets* candidatos C_k usando a função apriori-gen (descrito na Seção 3.1.1). A seguir (linhas 12 a 17 do Algoritmo 4), 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 18 do Algoritmo 4).

Na última etapa (linhas 19 a 26 do Algoritmo 4) é acrescentada uma verificação no passo $k = 2$ para garantir que no conjunto de *itemsets* freqüentes $L_{k=2}$ os 2 itens de cada *itemset* não possuam nenhum parentesco um com o outro em qualquer que seja o nível da hierarquia entre eles no conjunto de taxonomias definidas. Essa etapa corresponde à melhoria número 3 do algoritmo Cumulate (Srikant & Agrawal, 1995) (remover *itemsets* que contenham apenas um item e seu ancestral), descrita na Seção 4.3.

A solução final é dada pela união dos conjuntos L_k de k-*itemsets* freqüentes (linha 27). Essa solução é utilizada como entrada para algum algoritmo que gera regras de associação a partir do *itemset* (por exemplo, o descrito na Seção 3.2).

A seguir, é apresentado um pequeno exemplo de funcionamento do algoritmo *Apriori-Gen* para um melhor entendimento do algoritmo.

5.2 Exemplo de Funcionamento do Algoritmo *AprioriGen*

No Exemplo 6 é ilustrado o funcionamento do Algoritmo 4, *AprioriGen*, aplicado a um pequeno conjunto de transações.

Exemplo 6 *Seja D uma Base de Dados que contém um conjunto de itens $A =$* {bermuda, boné, camiseta, chinelo, sandália, short, 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 4.*

Tabela 4: Relação de itens comprados por transação.

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

Também é apresentado na Figura 4 o conjunto de Taxonomias $\mathcal{T}$ que será fornecido ao algoritmo para gerar as Regras de Associação Generalizadas.

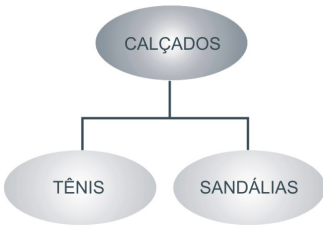

Figura 4: Conjunto de Taxonomias $\mathcal{T}$.

*Supondo uma execução com o valor do suporte mínimo (***sup-min***) igual a* 50% *e da confiança mínima (***conf-min***) igual a* 50%*, o algoritmo inicialmente percorre cada transação t_i visitando cada item verificando se ele pertence a algum item filho do conjunto de Taxonomia $\mathcal{T}$. Caso encontre, os itens pais da taxonomia de cada item filho serão adicionados na transação (linhas 1 a 8 do Algoritmo 4). Nesse exemplo, pode-se verificar que os itens* tênis *e* sandália *da taxonomia $\mathcal{T}$ estão presente nas transações* 1, 2 *e* 3*. De acordo com a taxonomia $\mathcal{T}$ os itens* tênis *e* sandália *possuem somente um item pai (*calçados*), sendo este adicionado em cada transação na qual pertencem os itens* tênis *e* sandália*, como é apresentado na Tabela 5.*

Tabela 5: Relação de transações após execução das linhas 1 a 8 do Algoritmo 4.

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

Caso houvesse uma transação em que estivessem presentes os itens tênis *e* sandália*, o item pai (*calçado*) iria se repetir na transação. Assim, a fim de evitar a repetição de*

itens na transação é executado um procedimento de remoção dos itens repetidos para cada transação t_i (linha 5 do algoritmo).

Em seguida, são executadas as linhas 9 a 18 do Algoritmo AprioriGen, os quais correspondem ao Algoritmo Apriori, sendo executado de forma análoga ao exemplo 4 da Seção 3.1.2 utilizando esse novo conjunto de transações obtido na Tabela 5.

Quando no passo $k = 2$, são executadas as linhas 19 a 26 do algoritmo, onde são verificados os itemset freqüentes de tamanho igual a 2 que pertencem ao conjunto $L_{k=2}$. Nesse exemplo, os itemsets freqüentes de tamanho igual a 2 são L_2 = ({camiseta, tênis}, {camiseta, calçados}, {tênis, calçados}). Para cada item de cada 2-itemset freqüente é verificado se o primeiro item é parente do segundo item ou vice-versa (ser parente significa ser possível um item alcançar outro item seguindo uma hierarquia de níveis na taxonomia). Caso os itens desses itemsets freqüentes de tamanho 2 sejam parentes entre si, esses itemsets são descartados e removidos do conjunto L_2 de itemsets freqüentes. De acordo com o exemplo, verifica-se que nos itemsets {camiseta, tênis} e {camiseta, calçados} os itens não são parentes, portanto ambos permanecem no conjunto. Já no itemset {tênis, calçados} os itens são parentes e, portanto, o mesmo é removido do conjunto L_2 de itemsets freqüentes. De fato, o itemset {tênis, calçados} produziria regras como tênis ⇒ calçados *e* calçados ⇒ tênis*, as quais não fazem sentido para o usuário.*

E, finalmente, na linha 27 do algoritmo, conforme a Tabela 6, é apresentado o conjunto Resposta de itemsets freqüentes. O conjunto Resposta é utilizado como entrada para algum algoritmo que gera regras de associação com itemsets freqüentes, como o algoritmo apresentado na Seção 3.2.

Tabela 6: Conjunto *Resposta* contendo os *itemsets* freqüentes.

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

No caso da implementação do algoritmo AprioriGen é realizado tanto a geração do conjunto de itemsets freqüentes quanto a produção de Regras de Associação Generalizadas a partir dos itemsets obtidos. Sendo assim, tem-se como resultado a obtenção das seguintes regras de associação generalizadas:

Regras geradas pelo *itemset* {*camiseta, tênis*} *:*

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**.

tênis $\Rightarrow$ camiseta,

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

Regras geradas pelo *itemset* {*camiseta, calçados*} *:*

camiseta $\Rightarrow$ calçados,

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

calçados $\Rightarrow$ camiseta,

- *suporte* = *suporte*($\{$*camiseta, calçados*$\}$) = 50%.
- *confiança* = $\frac{suporte(\{camiseta,\ calçados\})}{suporte(\{calçados\})} = \frac{50}{75} = 66,67\%$, *que é maior do que* **conf-min**.

O algoritmo gera tanto regras de associação generalizadas quanto regras tradicionais, assim como os outros algoritmos. Note que o itemset $\{$*camiseta, tênis*$\}$ *não possui o item pai (*calçado*) descrito na Figura 4. Portanto, as regras produzidas a partir dele não são consideradas Regras de Associação Generalizadas.*

Na próxima Seção, são descritas algumas observações e informações importantes sobre a implementação do algoritmo *AprioriGen*.

6 Implementação do Algoritmo *AprioriGen*

A implementação do Algoritmo *AprioriGen* foi realizada com base no estudo realizado com a versão 4.27 de 20/06/2005 da implementação do Algoritmo *Apriori* [2] desenvolvida por Borgelt & Kruse (2002), sendo a mesma escrita na linguagem de programação C ANSI.

[2]Uma implementação gratuita do algoritmo *Apriori* com toda a documentação pode ser obtida no site `http://fuzzy.cs.uni-magdeburg.de/~borgelt/apriori.html`.

A implementação desenvolvida por Borgelt & Kruse (2002) é baseada no conceito de uma árvore prefixa. Embora a idéia para usar esse tipo de estrutura de dados não seja nova, essa é uma das maneiras para organizar os nós de uma árvore, para codificar os itens, e para organizar as transações. Essa organização em árvore pode ser usada para minimizar o tempo necessário para procurar os *itemsets* freqüentes bem como para reduzir a quantidade de memória para armazenagem dos contadores (Borgelt & Kruse, 2002).

Essa implementação do Algoritmo *Apriori* desenvolvida por Christian Borgelt (Borgelt & Kruse, 2002) trabalha em duas etapas: na primeira etapa os *itemsets* freqüentes são determinados. Estes são conjuntos de itens que têm pelo menos o *suporte mínimo* especificado pelo usuário. Geralmente a primeira etapa é a mais importante, pois ela conta com a maior parte do tempo de processamento. Na segunda etapa as regras de associação são geradas a partir dos *itemsets* freqüentes encontrados na primeira etapa.

É importante destacar que a implementação proposta por Borgelt & Kruse (2002), assim como a implementação do Algoritmo *AprioriGen*, gera Regras de Associação com somente um item simples no conseqüente *(RHS)*, reduzindo assim a quantidade de regras e gerando regras menos complexas, diferente do Algoritmo *Apriori* tradicional proposto por Agrawal & Srikant (1994), que pode gerar regras com mais de um item em ambos os lados *(RHS e LHS)*.

Uma característica da implementação do algoritmo *AprioriGen* é a não possibilidade de um item filho (item específico) participar de mais de uma taxonomia. Por exemplo, se existir uma taxonomia em que o item sapato é um calçado e em outra taxonomia o item sapato é um vestuário, o algoritmo *AprioriGen* irá considerar somente a primeira ocorrência no arquivo de taxonomias.

Desde o início da alteração do código fonte, após o entendimento e estudo da implementação do algoritmo *Apriori* proposta por Borgelt & Kruse (2002), houve a preocupação em identificar somente as partes principais a serem modificadas, aproveitando ao máximo o que já estava implementado e adicionando funcionalidades que interferissem o menos possível no código fonte original. A fim de obter um controle das alterações do *AprioriGen* em relação à proposta inicial de Borgelt & Kruse (2002), a cada linha modificada foi acrescentado o comentário /**/.

Todas as funções implementadas estão incluídas em um único arquivo chamado taxonomia.c, fornecendo suporte para a geração de Regras de Associação Generalizadas com uso de taxonomias. Também é definida uma biblioteca denominada taxonomia.h, que apresenta as constantes e protótipos das funções.

A seguir, são descritos os formatos dos arquivos de entrada e saída da implementação

do algoritmo *AprioriGen* e o modo de execução.

6.1 Arquivos de Entrada e Saída

Para a execução da implementação deve-se ter como arquivos de entrada o arquivo do conjunto de transações (qualquer nome sugerido pelo usuário podendo ter a extensão opcional .data) e o arquivo de taxonomias (qualquer nome sugerido pelo usuário podendo ter a extensão opcional .tax) e como arquivo de saída o conjunto de regras de associação geradas (qualquer nome sugerido pelo usuário podendo ter a extensão opcional .rules).

No arquivo do conjunto de transações cada linha representa uma transação, por exemplo, uma lista de compras de um supermercado, onde cada linha representa a compra de um cliente com os itens comprados, como na Figura 5, apresentando um arquivo com 4 transações. Note que cada item da transação deve ser separado por espaços em branco.

```
calça camiseta tênis
camiseta tênis
bermuda sandália
calça boné
```

Figura 5: Exemplo de arquivo de transações (dados.data).

No arquivo de taxonomias cada linha do arquivo representa uma taxonomia que contém um item representando os ancestrais e, um ou mais itens representando os descendentes. Os itens que representam especializações máximas são delimitados por "( )" (parênteses) e separados por "," (vírgula). Na Figura 6 é apresentado um arquivo de taxonomias representando a taxonomia ilustrada na Figura 2 da Seção 4. Note que no final de cada linha deve ser adicionado um número entre parênteses, o qual indica o nível na taxonomia. As linhas do arquivo devem estar em ordem crescente de nível.

```
calçados(tênis,sandálias)(1).
roupas_leves(camisetas,bermudas)(1).
roupas(roupas_leves,calças)(2).
vestuario(roupas,calçados)(3).
```

Figura 6: Exemplo do arquivo de taxonomias da Figura 2 (taxonomia.tax).

Em cada linha do arquivo de saída é apresentada uma regra de associação obtida,

com o conseqüente RHS à esquerda da seta e o antecedente LHS à direita da seta, lembrando que o conseqüente RHS será sempre um único item simples. Na Figura 7 é apresentado um exemplo de um conjunto de regras de associação extraído.

```
calçados <-  (75.00%/3, 75.00%), (Y)
tênis <- camiseta  (50.00%/2, 100.00%), (N)
camiseta <- tênis  (50.00%/2, 100.00%), (N)
calçados <- camiseta  (50.00%/2, 100.00%), (Y)
camiseta <- calçados  (50.00%/2, 66.67%), (Y)
```

Figura 7: Exemplo de um conjunto de saída de regras de associação (saida.rules).

A primeira delimitação por parênteses "( )" informa as medidas de avaliação de cada regra de acordo com as opções definidas pelo usuário. Nesse exemplo, é mostrado apenas o Suporte e a Confiança. Já os últimos parênteses informam se a regra de associação é generalizada (Y), ou não generalizada (N). Note que a primeira regra não possui antecedente LHS, já que a mesma apenas fornece informações sobre o item, podendo ser desconsiderada, sendo uma ocorrência normal tanto na implementação do algoritmo *Apriori* desenvolvido por Borgelt & Kruse (2002) quanto do algoritmo *AprioriGen*.

6.2 Modo de Execução

A execução do algoritmo *AprioriGen* é realizada por linha de comando da mesma maneira que a versão disponível do autor (Borgelt & Kruse, 2002). Entretanto, foi acrescentada a opção -w, a qual permite habilitar o programa para gerar Regras de Associação Generalizadas. É necessário informar os arquivos seguindo necessariamente essa ordem: primeiro o arquivo de transações (.data), segundo o arquivo de saída (.rules) onde se quer armazenar as regras de associação e, por último, o arquivo de taxonomias (.tax), informando para cada argumento da linha de comando o nome e o local do arquivo onde estão presentes. Um exemplo de execução por linha de comando é apresentado a seguir.

```
aprioriGen.exe -c60 -s30 -o -a -p%.2f%% -w dados.data saida.rules taxonomia.tax
```

As opções -c60 e -s30 informam ao programa, respectivamente, a Confiança mínima e o Suporte mínimo que correspondem a 60% e 30%. A opção -o informa que será usada a definição tradicional de Suporte. A opção -a faz com que seja colocado o valor de Suporte em porcentagem e em valor absoluto (número de transações) em cada regra.

Com a opção -p%.2f%% é possível informar o formato de saída das medidas. Nesse exemplo foi adicionado o sinal de porcentagem com 2 casas decimais após a vírgula. A opção -w habilita o programa a gerar regras Regras de Associação Generalizadas.

Mais informações a respeito de outras opções adicionais que podem ser incluídas e outras formas de medidas de avaliação de regras podem ser obtidas na documentação do software *Apriori* (Borgelt & Kruse, 2002).

7 Experimentos Realizados

Nesta seção são descritos alguns experimentos realizados com o objetivo de mostrar a geração de Regras de Associação Generalizadas durante a fase de Extração de Padrões com o uso de taxonomias. Na realização desses experimentos foram utilizados dois conjuntos de dados e seus respectivos conjuntos de taxonomias. O primeiro conjunto de dados (base1.data) foi extraído de um relatório técnico em Srikant & Agrawal (1997a) referente a itens de vestuário, contendo 6 transações com 5 itens distintos. O segundo conjunto de dados (base2.data) contém informações referentes as operações de vendas de um dia em um supermercado localizado na cidade de São Carlos, possuindo 1716 transações com 1939 itens distintos. A seguir é apresentada a descrição dos conjuntos de dados e do conjunto de taxonomia para cada uma das bases de dados.

7.1 Descrição das Bases de Dados

Os experimentos foram realizados em conjuntos de dados já pré-processados das bases de dados. O conjunto de dados da primeira base de dados (base1.data) é apresentado na Figura 8.

```
shirt
jacket  hiking_boots
ski_pants  hiking_boots
shoes
shoes
jacket
```

Figura 8: Conjunto de dados da primeira base de dados (base1.data).

Na Figura 9 é apresentado o conjunto de taxonomias da primeira base de dados.

```
outerwear(jacket,ski_pants)(1).
footwear(shoes,hiking_boots)(1).
clothes(outerwear,shirt)(2).
```

Figura 9: Conjunto de taxonomias da primeira base de dados (base1tax.tax).

É apresentado a seguir, na Figura 10, somente uma pequena amostra de um conjunto de transações da segunda base de dados, devido a grande quantidade de transações disponíveis (base2.data).

```
AGUA_SANITCANDURA CERA_BRAVO
ADES_ORIGINAL LEITE_PARMALAT
DESBA_BANHO SHALL_CLEAR
AGUA_DE_COCO_SOCOCO BALAS_RECHMEL_DORI CHOCNESTLE CORN_FLAKES_KELLOG_S LEITE_NILZA MARGDORIANA
MOLHO_PIC_3_EM_1_HELLMANN_S NESTON_FLOCOS_DE_CEREAIS
OVOS_BRANCOS_KAMIMURA
PORTA_MANTIMCRISTAL_PLASUTIL
PAO_WICKBOLD_SIRIO
CAFE_SERRA_DA_GRAMA DESINFSCARLIM PAPEL_HIGSUBLIME
CLIGHT_DIET CREME_DE_QUEIJO_KNORR SAL_CISNE
OVOS_BRANCOS_KAMIMURA PALITOS_GINA PAPEL_HIGNEVE
ADES_ORIGINAL BEBLACTEA_DAN_UP BISCTRAKINAS
CHEIRO_VERDE PAO_WICKBOLD_LIGHT RABANETE_SATO
```

Figura 10: Amostra do conjunto de dados da segunda base de dados (base2.data).

Na Figura 11 é apresentado o conjunto de taxonomias da segunda base de dados.

```
CHOCOLATE(CHOCNESTLE,CHOCLACTA,CHOCGAROTO)(1).
REFRIGERANTE(GUARANA_SCARLOS,GUARANA_SCHIN,GUARANA_ANTARTICA,FANTA,COCA_COLA,SPRITE)(1).
BISCOITO(BISCNESTLE,BISCNABISCO,BISCPIRAQUE,BISCMARILAN,BISCSAO_CARLOS,BISCPARMALAT,BISCTOSTINES)(1).
DETERGENTE(DETERGYPE,DETERGLIMPOL,DETERGMINUANO)(1).
PAPEL_HIGIENICO(PAPEL_HIGNEVE,PAPEL_HIGPERSONAL,PAPEL_HIGCAMELIA,PAPEL_HIGSUBLIME,PAPEL_HIGDUALETTE)(1).
MACARRAO_NINST(MACBARILLA,MACMAGGI,MACBASILAR,MACRENATA,MACARISCO,MACGALO)(1).
MACARRAO_INST(NISSIM_L_AMEM,MACESPAGUETE_MIOJO)(1).
CREME_DENTAL(CDSORRISO,CDCOLGATE,CDPREVENT,CDSIGNAL,CDCLOSEUP)(1).
LEITE_UHT(LEITE_PARMALAT,LEITE_NILZA,LEITE_LACTUS,LEITE_SALUTE,LEITE_BATAVO,LEITE_SAO_CARLOS)(1).
FARINHA_TRIGO(FARTRIGO_RENATA,FARTRIGO_DONA_BENTA)(1).
SABONETE(SABVINOLIA,SABLUX,SABPHEBO,SABPROTEX,SABDOVE)(1).
AMACIANTE(AMACBABY_SOFT,AMACCOMFORT,AMACCANDURA,AMACMON_BIJOU,AMACFOFO,AMACFLEUR_D_YPE)(1).
MOLHO_TOMATE(EXTRATO_ELEFANTE_KNORRCICA,MOLHO_TOMATE_CIRIO)(1).
FEIJAO(FEIJAO_GRAO_DO_CAMPO,FEIJAO_TORRESAN,FEIJAO_BROTO_LEGAL)(1).
CALDOS(CALDO_KNORR,CALDO_MAGGI)(1).
IORGUTE(IOGBATAVO,IOGNESTLE,IOGCORPUS,IOGMOLICO,IOGNINHO_SOLEIL)(1).
ACHOCOLATADO(ACHOCTODDY,ACHOCNESCAU,ACHOCOVOMALTINE,ACHOCGOLD)(1).
SUCOS(TANG,SUCO_DEL_VALLE,SUCO_JAL,SUCO_SINUELO,SUCO_NESTLE,SUCO_MAIS)(1).
```

```
ABSORVENTE(ABSCAREFREE,ABSSLIVRE,ABSINTGEL,ABSALWAYS)(1).
CATCHUP(CATCHUP_PIC,CATCHUP_HELLMANN_S)(1).
ELMA_CHIPS(BATATA_RUFFLES,BATATA_PALHA_ELMA_CHIPS)(1).
AMENDOIM(AMENDOIM_DORI,AMENDOIM_YOKI,AMENDOIM_C_R)(1).
ACUCAR(ACUCAR_DA_BARRA,ACUCAR_CRISTALCUCAR,ACUCAR_UNIAO)(1).
ARROZ(ARROZ_SANO,ARROZ_ALBARUSKA,ARROZ_PRATO_FINO,ARROZ_UNCLE_BENS,ARROZ_TIO_JOAO)(1).
OLEO(OLEO_SOJA_SADIA,OLEO_SOJA_SOYA,OLEO_SOJA_LIZA,OLEO_SOJA_SALADA)(1).
CAFE(CAFE_RIBEIRAO_BONITO,CAFE_OURO_BRASILEIRO,CAFE_SERRA_DA_GRAMA,CAFE_CABOCLO,CAFE_PELE)(1).
ALCOOL(ALCOOL_CANDURA,ALCOOL_MULTIALCOOL,ALCOOL_CLARA,ALCOOL_COPERALCOOL)(1).
QUEIJO_RALADO(QUEIJO_RALADO_TEIXEIRA_PARMESAO,QUEIJO_RALADO_TEIXEIRA_4_FORMAGGI,QUEIJO_RALADO_FAIXA_AZUL)(1).
QUEIJO_PECA(QUEIJO_SANCOR,QUEIJO_DANUBIO_MINAS,QUEIJO_FRESCATINO_MINAS_FRESCAL)(1).
SOBREMESA_CREMOSA_CHOCOLATE(CHANDELLE,DANETTE)(1).
ESPONJA(ESPONJA_SCOTCH_BRITE,ESPONJA_BOMBRIL)(1).
LEITE_ACHOCOLATADO(TODDYNHO,NESCAU_PRONTINHO)(1).
ALHO(ALHO_DA_ROCA,ALHO_BOIANI,ALHO_TOTAL,PURO_ALHO_BOIANI,PURO_ALHO_ARISCO)(1).
SABAO_PO(SPO_OMO,SPO_MINERVA,SPO_BRILHANTE,SPO_ACE,SPO_CAMPEIRO)(1).
ESPONJA_ACO(BOMBRIL,ESPONJA_DE_ACO_ASSOLAN)(1).
FLUOR(CEPACOL,ANTI_SEPTLISTERINE)(1).
FIO_DENTAL(FIO_DENTAL_REACH,FIO_DENTAL_TEK,FIO_DENTAL_DENTICS)(1).
ESCOVA_DENTE(EDJ_J_REACH,EDJOHNSON___JOHNSON,EDJOHNSON_S)(1).
CERVEJA(CERVXINGU,CERVCRYSTAL,CERVBRAHMA,CERVANTARCTICA,CERVSCHINCARIOL,CERVKAISER)(1).
VODKA(VODKA_ORLOFF,VODKA_BAIKAL,VODKA_SMIRNOFF)(1).
CHAMPAGNE(CHAMPAGNE_CHANDON,CHAMPAGNE_PETERLONGO,CHAMPAGNE_DE_GREVILLE,CHAMPAGNE_ESPDE_PRATA)(1).
ALFACE(ALFACE_CRESPA,ALFACE_AMERICANA,ALFACE_LISA)(1).
RUCULA(RUCULA_BIO_TERRA,RUCULA_SATO)(1).
COUVE(COUVE_MANTEIGA_ORGANICA,COUVE_MANTEIGA)(1).
VERDURAS(ALFACE,RUCULA,COUVE)(2).
HIGIENE_BUCAL(CREME_DENTAL,FLUOR,FIO_DENTAL,ESCOVA_DENTE)(2).
HIGIENE_PESSOAL(PAPEL_HIGIENICO,SABONETE,ABSORVENTE)(2).
BEBIDA_ALCOOLICA(CERVEJA,VODKA,CHAMPAGNE)(2).
BEBIDA_NALCOOLICA(REFRIGERANTE,IORGUTE,SUCOS,LEITE_UHT,LEITE_ACHOCOLATADO)(2).
PRODUTOS_LIMPEZA_COZINHA(DETERGENTE,ALCOOL,ESPONJA,ESPONJA_ACO)(2).
PRODUTOS_LIMPEZA_LAVANDERIA(AMACIANTE,SABAO_PO)(2).
DOCES(CHOCOLATE,BISCOITO,SOBREMESA_CREMOSA_CHOCOLATE)(2).
ENLATADOS(MOLHO_TOMATE,ACHOCOLATADO,CATCHUP)(2).
QUEIJO(QUEIJO_RALADO,QUEIJO_PECA)(2).
APERITIVOS(ELMA_CHIPS,AMENDOIM)(2).
MACARRAO(MACARRAO_NINST,MACARRAO_INST)(2).
PRODUTOS_BASICOS(FARINHA_TRIGO,FEIJAO,ACUCAR,ARROZ,OLEO,CAFE)(2).
TEMPEROS(CALDOS,ALHO)(2).
PRODUTOS_LIMPEZA(PRODUTOS_LIMPEZA_LAVANDERIA,PRODUTOS_LIMPEZA_COZINHA)(3).
PRODUTOS_ALIMENTICIOS(DOCES,ENLATADOS,QUEIJO,APERITIVOS,MACARRAO,PRODUTOS_BASICOS,TEMPEROS,VERDURAS)(3).
PRODUTOS_HIGIENE(HIGIENE_BUCAL,HIGIENE_PESSOAL)(3).
BEBIDA(BEBIDA_ALCOOLICA,BEBIDA_NALCOOLICA)(3).
```

Figura 11: Conjunto de taxonomias da segunda base de dados (base2tax.tax).

A seguir, são apresentados os procedimentos de como as Regras de Associação Generalizadas foram extraídas a partir dos conjuntos de dados e dos conjuntos de taxonomias de cada base de dados.

7.2 Mineração das Regras de Associação Generalizadas

A fim de possibilitar a comparação e comprovação das funcionalidades do algoritmo *AprioriGen*, foram realizados dois grupos de dois experimentos para cada uma das duas bases de dados, variando em cada grupo a taxa de suporte e confiança mínimas. O primeiro experimento utilizou o algoritmo *Apriori* tradicional utilizando a implementação de Borgelt & Kruse (2002) e o segundo experimento utilizou o algoritmo *AprioriGen*, que faz a inclusão dos itens ancestrais nas transações onde ocorrem os itens descendentes (procedimento que corresponde ao Algoritmo Básico) e a remoção dos *itemsets* que contenham apenas um item e seu ancestral (procedimento que corresponde a melhoria 3 do Algoritmo Cumulate). Na Subseção 7.3 são discutidos os resultados obtidos dos experimentos realizados comparando as duas bases de dados.

7.2.1 Experimentos com a Base de Dados 1

Na primeira base de dados, utilizando o conjunto de dados (base1.data) e o conjunto de taxonomia (base1tax.tax), foram efetuadas execuções com a implementação do algoritmo *Apriori* e com a versão da implementação do algoritmo *AprioriGen* usando as opções de linha de comando apresentadas a seguir. O primeiro grupo de experimentos (1 e 2) da primeira base de dados (base1.data) foi conduzido usando suporte mínimo igual a 30% e confiança mínima igual a 60%.

Grupo 1:

- Experimento 1 – Para a execução do Algoritmo *Apriori*:

```
apriori.exe -c60 -s30 -o -a -p%.2f%% base1.data saidabase1grupo1exp1.rules
```

- Experimento 2 – Para a execução do Algoritmo *AprioriGen*:

```
aprioriGen.exe -c60 -s30 -o -a -p%.2f%% -w base1.data saidabase1grupo1exp2.rules base1tax.tax
```

O Experimento 1 com a execução do algoritmo *Apriori* apresenta a mesma linha que o Experimento 2, exceto por não ter a opção -w (opção que habilita o uso de taxonomias) e o conjunto de taxonomias (.tax). A opção -c informa o valor da confiança mínima, -s o valor de suporte mínimo, -o que será usada a definição tradicional de Suporte e -p o formato de saída das medidas. No Experimento 2 foi definido o suporte mínimo igual a 30% e a confiança mínima igual a 60% de acordo com o exemplo do relatório técnico de Srikant & Agrawal (1997a), na qual esses experimentos se baseiam.

Em relação ao resultado referente ao Experimento 1, não foi obtida nenhuma regra após sua execução devido à pouca quantidade de itens contidos em cada transação e aos altos valores de suporte mínimo e confiança mínima usados pelo exemplo do relatório técnico de Srikant & Agrawal (1997a). Em relação ao resultado referente ao Experimento 2, foram obtidas 6 regras após a execução do algoritmo *AprioriGen*, as quais são apresentadas na Figura 12.

```
footwear <-  (66.67%/4, 66.67%), (Y)
clothes <-  (66.67%/4, 66.67%), (Y)
outerwear <- hiking_boots  (33.33%/2, 100.00%), (Y)
hiking_boots <- outerwear  (33.33%/2, 66.67%), (Y)
clothes <- hiking_boots  (33.33%/2, 100.00%), (Y)
footwear <- outerwear  (33.33%/2, 66.67%), (Y)
```

Figura 12: Conjunto de regras obtidas da primeira base de dados utilizando o algoritmo *AprioriGen* com suporte mínimo igual a 30% e confiança mínima igual a 60% (saidabase1grupo1exp2.rules).

Com esta base também foi realizado um segundo grupo de experimentos definindo um valor de 10% para suporte mínimo e 50% para confiança mínima.

Grupo 2:

- Experimento 3 – Para a execução do Algoritmo *Apriori*:

  apriori.exe -c50 -s10 -o -a -p%.2f%% base1.data saidabase1grupo2exp3.rules

- Experimento 4 – Para a execução do Algoritmo *AprioriGen*:

aprioriGen.exe -c50 -s10 -o -a -p%.2f%% -w base1.data saidabase1grupo2exp4.rules base1tax.tax

Para esses novos valores foram executados os Experimentos 3 e 4, cujos resultados são apresentados, respectivamente, nas Figuras 13 e 14.

```
hiking_boots <- ski_pants  (16.67%/1, 100.00%), (N)
ski_pants <- hiking_boots  (16.67%/1, 50.00%), (N)
hiking_boots <- jacket  (16.67%/1, 50.00%), (N)
jacket <- hiking_boots  (16.67%/1, 50.00%), (N)
```

Figura 13: Conjunto de regras obtidas da primeira base de dados utilizando o algoritmo *Apriori* com suporte mínimo igual a 10% e confiança mínima igual a 50% (saidabase1grupo2exp3.rules).

```
outerwear <-  (50.00%/3, 50.00%), (Y)
footwear <-  (66.67%/4, 66.67%), (Y)
clothes <-  (66.67%/4, 66.67%), (Y)
hiking_boots <- ski_pants  (16.67%/1, 100.00%), (N)
ski_pants <- hiking_boots  (16.67%/1, 50.00%), (N)
footwear <- ski_pants  (16.67%/1, 100.00%), (Y)
hiking_boots <- jacket  (16.67%/1, 50.00%), (N)
jacket <- hiking_boots  (16.67%/1, 50.00%), (N)
footwear <- jacket  (16.67%/1, 50.00%), (Y)
outerwear <- hiking_boots  (33.33%/2, 100.00%), (Y)
hiking_boots <- outerwear  (33.33%/2, 66.67%), (Y)
clothes <- hiking_boots  (33.33%/2, 100.00%), (Y)
hiking_boots <- clothes  (33.33%/2, 50.00%), (Y)
footwear <- outerwear  (33.33%/2, 66.67%), (Y)
outerwear <- footwear  (33.33%/2, 50.00%), (Y)
clothes <- footwear  (33.33%/2, 50.00%), (Y)
footwear <- clothes  (33.33%/2, 50.00%), (Y)
```

Figura 14: Conjunto de regras obtidas da primeira base de dados utilizando o algoritmo *AprioriGen* com suporte mínimo igual a 10% e confiança mínima igual a 50% (saidabase1grupo2exp4.rules).

7.2.2 Experimentos com a Base de Dados 2

Assim como nos experimentos anteriores, na segunda base de dados, utilizando o conjunto de dados (base2.data) e o conjunto de taxonomia (base2tax.tax), foram efetuadas execuções com a implementação do algoritmo *Apriori* e a implementação do algoritmo *AprioriGen* usando as opções de linha de comando apresentadas a seguir. O primeiro grupo de experimentos (5 e 6) da segunda base de dados (base2.data) foi conduzido utilizando suporte mínimo igual a 15% e confiança mínima igual a 15%.

Grupo 1:

- Experimento 5 – Para a execução do Algoritmo *Apriori*:

```
apriori.exe -c15 -s15 -o -a -p%.2f%% base2.data saidabase2grupo1exp5.rules
```

- Experimento 6 – Para a execução do Algoritmo *AprioriGen*:

```
aprioriGen.exe -c15 -s15 -o -a -p%.2f%% -w base2.data saidabase2grupo1exp6.rules base2tax.tax
```

O Experimento 5, com a execução do algoritmo *Apriori*, apresenta a mesma linha de comando que o Experimento 6, exceto por não ter a opção -w (opção que habilita o

uso de taxonomias) e o conjunto de taxonomias (.tax). A opção -c informa o valor da confiança mínima, -s o valor de suporte mínimo, -o que será usada a definição tradicional de Suporte e -p o formato de saída das medidas.

Em relação ao resultado referente ao Experimento 5, não foi obtida nenhuma regra após sua execução devido aos altos valores de suporte mínimo (15%) e confiança mínima (15%). Em relação ao resultado referente ao Experimento 6, na Figura 15 é apresentado o arquivo de saída (.rules) com as regras obtidas após a execução do algoritmo *AprioriGen*.

```
COCA_COLA <-  (20.28%/348, 20.28%), (N)
PAPEL_HIGIENICO <-  (16.90%/290, 16.90%), (Y)
REFRIGERANTE <-  (26.22%/450, 26.22%), (Y)
PRODUTOS_LIMPEZA_COZINHA <-  (17.02%/292, 17.02%), (Y)
LEITE_UHT <-  (25.52%/438, 25.52%), (Y)
BISCOITO <-  (24.71%/424, 24.71%), (Y)
PRODUTOS_LIMPEZA <-  (21.91%/376, 21.91%), (Y)
HIGIENE_PESSOAL <-  (23.66%/406, 23.66%), (Y)
DOCES <-  (29.66%/509, 29.66%), (Y)
PRODUTOS_BASICOS <-  (28.73%/493, 28.73%), (Y)
PRODUTOS_HIGIENE <-  (26.05%/447, 26.05%), (Y)
BEBIDA_NALCOOLICA <-  (48.54%/833, 48.54%), (Y)
BEBIDA <-  (49.36%/847, 49.36%), (Y)
PRODUTOS_ALIMENTICIOS <-  (54.14%/929, 54.14%), (Y)
PRODUTOS_ALIMENTICIOS <- REFRIGERANTE  (18.30%/314, 69.78%), (Y)
REFRIGERANTE <- PRODUTOS_ALIMENTICIOS  (18.30%/314, 33.80%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_LIMPEZA_COZINHA  (15.50%/266, 91.10%), (Y)
PRODUTOS_LIMPEZA_COZINHA <- PRODUTOS_ALIMENTICIOS  (15.50%/266, 28.63%), (Y)
PRODUTOS_ALIMENTICIOS <- LEITE_UHT  (18.76%/322, 73.52%), (Y)
LEITE_UHT <- PRODUTOS_ALIMENTICIOS  (18.76%/322, 34.66%), (Y)
BEBIDA_NALCOOLICA <- BISCOITO  (17.54%/301, 70.99%), (Y)
BISCOITO <- BEBIDA_NALCOOLICA  (17.54%/301, 36.13%), (Y)
BEBIDA <- BISCOITO  (17.66%/303, 71.46%), (Y)
BISCOITO <- BEBIDA  (17.66%/303, 35.77%), (Y)
PRODUTOS_HIGIENE <- PRODUTOS_LIMPEZA  (15.33%/263, 69.95%), (Y)
PRODUTOS_LIMPEZA <- PRODUTOS_HIGIENE  (15.33%/263, 58.84%), (Y)
BEBIDA_NALCOOLICA <- PRODUTOS_LIMPEZA  (15.68%/269, 71.54%), (Y)
PRODUTOS_LIMPEZA <- BEBIDA_NALCOOLICA  (15.68%/269, 32.29%), (Y)
BEBIDA <- PRODUTOS_LIMPEZA  (15.73%/270, 71.81%), (Y)
PRODUTOS_LIMPEZA <- BEBIDA  (15.73%/270, 31.88%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_LIMPEZA  (18.88%/324, 86.17%), (Y)
PRODUTOS_LIMPEZA <- PRODUTOS_ALIMENTICIOS  (18.88%/324, 34.88%), (Y)
BEBIDA_NALCOOLICA <- HIGIENE_PESSOAL  (17.19%/295, 72.66%), (Y)
HIGIENE_PESSOAL <- BEBIDA_NALCOOLICA  (17.19%/295, 35.41%), (Y)
BEBIDA <- HIGIENE_PESSOAL  (17.37%/298, 73.40%), (Y)
HIGIENE_PESSOAL <- BEBIDA  (17.37%/298, 35.18%), (Y)
PRODUTOS_ALIMENTICIOS <- HIGIENE_PESSOAL  (19.81%/340, 83.74%), (Y)
HIGIENE_PESSOAL <- PRODUTOS_ALIMENTICIOS  (19.81%/340, 36.60%), (Y)
BEBIDA_NALCOOLICA <- DOCES  (20.05%/344, 67.58%), (Y)
DOCES <- BEBIDA_NALCOOLICA  (20.05%/344, 41.30%), (Y)
BEBIDA <- DOCES  (20.16%/346, 67.98%), (Y)
DOCES <- BEBIDA  (20.16%/346, 40.85%), (Y)
PRODUTOS_HIGIENE <- PRODUTOS_BASICOS  (15.15%/260, 52.74%), (Y)
PRODUTOS_BASICOS <- PRODUTOS_HIGIENE  (15.15%/260, 58.17%), (Y)
```

```
BEBIDA_NALCOOLICA <- PRODUTOS_BASICOS  (19.41%/333, 67.55%), (Y)
PRODUTOS_BASICOS <- BEBIDA_NALCOOLICA  (19.41%/333, 39.98%), (Y)
BEBIDA <- PRODUTOS_BASICOS  (19.52%/335, 67.95%), (Y)
PRODUTOS_BASICOS <- BEBIDA  (19.52%/335, 39.55%), (Y)
BEBIDA_NALCOOLICA <- PRODUTOS_HIGIENE  (18.82%/323, 72.26%), (Y)
PRODUTOS_HIGIENE <- BEBIDA_NALCOOLICA  (18.82%/323, 38.78%), (Y)
BEBIDA <- PRODUTOS_HIGIENE  (19.00%/326, 72.93%), (Y)
PRODUTOS_HIGIENE <- BEBIDA  (19.00%/326, 38.49%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_HIGIENE  (21.68%/372, 83.22%), (Y)
PRODUTOS_HIGIENE <- PRODUTOS_ALIMENTICIOS  (21.68%/372, 40.04%), (Y)
PRODUTOS_ALIMENTICIOS <- BEBIDA_NALCOOLICA  (32.52%/558, 66.99%), (Y)
BEBIDA_NALCOOLICA <- PRODUTOS_ALIMENTICIOS  (32.52%/558, 60.06%), (Y)
PRODUTOS_ALIMENTICIOS <- BEBIDA  (32.63%/560, 66.12%), (Y)
BEBIDA <- PRODUTOS_ALIMENTICIOS  (32.63%/560, 60.28%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_LIMPEZA BEBIDA_NALCOOLICA  (15.03%/258, 95.91%), (Y)
BEBIDA_NALCOOLICA <- PRODUTOS_LIMPEZA PRODUTOS_ALIMENTICIOS  (15.03%/258, 79.63%), (Y)
PRODUTOS_LIMPEZA <- BEBIDA_NALCOOLICA PRODUTOS_ALIMENTICIOS  (15.03%/258, 46.24%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_LIMPEZA BEBIDA  (15.09%/259, 95.93%), (Y)
BEBIDA <- PRODUTOS_LIMPEZA PRODUTOS_ALIMENTICIOS  (15.09%/259, 79.94%), (Y)
PRODUTOS_LIMPEZA <- BEBIDA PRODUTOS_ALIMENTICIOS  (15.09%/259, 46.25%), (Y)
PRODUTOS_ALIMENTICIOS <- HIGIENE_PESSOAL BEBIDA_NALCOOLICA  (15.68%/269, 91.19%), (Y)
BEBIDA_NALCOOLICA <- HIGIENE_PESSOAL PRODUTOS_ALIMENTICIOS  (15.68%/269, 79.12%), (Y)
HIGIENE_PESSOAL <- BEBIDA_NALCOOLICA PRODUTOS_ALIMENTICIOS  (15.68%/269, 48.21%), (Y)
PRODUTOS_ALIMENTICIOS <- HIGIENE_PESSOAL BEBIDA  (15.79%/271, 90.94%), (Y)
BEBIDA <- HIGIENE_PESSOAL PRODUTOS_ALIMENTICIOS  (15.79%/271, 79.71%), (Y)
HIGIENE_PESSOAL <- BEBIDA PRODUTOS_ALIMENTICIOS  (15.79%/271, 48.39%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_HIGIENE BEBIDA_NALCOOLICA  (17.19%/295, 91.33%), (Y)
BEBIDA_NALCOOLICA <- PRODUTOS_HIGIENE PRODUTOS_ALIMENTICIOS  (17.19%/295, 79.30%), (Y)
PRODUTOS_HIGIENE <- BEBIDA_NALCOOLICA PRODUTOS_ALIMENTICIOS  (17.19%/295, 52.87%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_HIGIENE BEBIDA  (17.31%/297, 91.10%), (Y)
BEBIDA <- PRODUTOS_HIGIENE PRODUTOS_ALIMENTICIOS  (17.31%/297, 79.84%), (Y)
PRODUTOS_HIGIENE <- BEBIDA PRODUTOS_ALIMENTICIOS  (17.31%/297, 53.04%), (Y)
```

Figura 15: Conjunto de regras obtidas da segunda base de dados utilizando o algoritmo *AprioriGen* com suporte mínimo igual a 15% e confiança mínima igual a 15% (saidabase2grupo1exp6.rules).

Com esta base também foi realizado um segundo grupo de experimentos definindo um valor de 4% para suporte mínimo e 5% para confiança mínima.

Grupo 2:

- Experimento 7 – Para a execução do Algoritmo *Apriori*:

apriori.exe -c5 -s4 -o -a -p%.2f%% base2.data saidabase2grupo2exp7.rules

- Experimento 8 – Para a execução do Algoritmo *AprioriGen*:

aprioriGen.exe -c5 -s4 -o -a -p%.2f%% -w base2.data saidabase2grupo2exp8.rules base2tax.tax

Para esses valores foram executados os Experimentos 7 e 8, cujos resultados são apresentados, respectivamente, nas Figuras 16 e 17. Na Figura 17 é apresentada somente uma pequena amostra do conjunto de regras, devido a grande quantidade de regras obtidas (25702 regras utilizando o *AprioriGen*).

```
OVOS_BRANCOS_KAMIMURA <-  (5.13%/88, 5.13%), (N)
FANTA <-  (5.65%/97, 5.65%), (N)
CAFE_SERRA_DA_GRAMA <-  (5.65%/97, 5.65%), (N)
CHOCLACTA <-  (5.48%/94, 5.48%), (N)
LEITE_LACTUS <-  (7.52%/129, 7.52%), (N)
LEITE_PARMALAT <-  (7.17%/123, 7.17%), (N)
GUARANA_ANTARTICA <-  (7.23%/124, 7.23%), (N)
ARROZ_PRATO_FINO <-  (5.48%/94, 5.48%), (N)
PAPEL_HIGNEVE <-  (7.63%/131, 7.63%), (N)
MACRENATA <-  (5.65%/97, 5.65%), (N)
SAL_CISNE <-  (6.00%/103, 6.00%), (N)
ACHOCNESCAU <-  (5.65%/97, 5.65%), (N)
BOMBRIL <-  (5.36%/92, 5.36%), (N)
FARTRIGO_RENATA <-  (6.88%/118, 6.88%), (N)
MAIONESE_HELLMANN_S <-  (6.18%/106, 6.18%), (N)
DETERGLIMPOL <-  (6.88%/118, 6.88%), (N)
SPO_OMO <-  (6.53%/112, 6.53%), (N)
PAPEL_HIGPERSONAL <-  (7.98%/137, 7.98%), (N)
ACUCAR_UNIAO <-  (8.45%/145, 8.45%), (N)
CREME_DE_LEITE_NESTLE <-  (8.45%/145, 8.45%), (N)
BISCNESTLE <-  (12.30%/211, 12.30%), (N)
LEITE_MOCA <-  (11.07%/190, 11.07%), (N)
COCA_COLA <-  (20.28%/348, 20.28%), (N)
COCA_COLA <- GUARANA_ANTARTICA  (4.95%/85, 68.55%), (N)
GUARANA_ANTARTICA <- COCA_COLA  (4.95%/85, 24.43%), (N)
LEITE_MOCA <- CREME_DE_LEITE_NESTLE  (5.13%/88, 60.69%), (N)
CREME_DE_LEITE_NESTLE <- LEITE_MOCA  (5.13%/88, 46.32%), (N)
COCA_COLA <- CREME_DE_LEITE_NESTLE  (4.31%/74, 51.03%), (N)
CREME_DE_LEITE_NESTLE <- COCA_COLA  (4.31%/74, 21.26%), (N)
LEITE_MOCA <- BISCNESTLE  (4.43%/76, 36.02%), (N)
BISCNESTLE <- LEITE_MOCA  (4.43%/76, 40.00%), (N)
COCA_COLA <- BISCNESTLE  (4.84%/83, 39.34%), (N)
BISCNESTLE <- COCA_COLA  (4.84%/83, 23.85%), (N)
COCA_COLA <- LEITE_MOCA  (5.19%/89, 46.84%), (N)
LEITE_MOCA <- COCA_COLA  (5.19%/89, 25.57%), (N)
```

Figura 16: Conjunto de regras obtidas da segunda base de dados utilizando o algoritmo *Apriori* com suporte mínimo igual a 4% e confiança mínima igual a 5% (saida-base2grupo2exp7.rules).

```
PRODUTOS_ALIMENTICIOS <-  (54.14%/929, 54.14%), (Y)
BEBIDA <- CAFE_SERRA_DA_GRAMA  (4.02%/69, 71.13%), (Y)
CAFE_SERRA_DA_GRAMA <- BEBIDA  (4.02%/69, 8.15%), (Y)
BEBIDA_NALCOOLICA <- CHOCLACTA  (4.02%/69, 73.40%), (Y)
CHOCLACTA <- BEBIDA_NALCOOLICA  (4.02%/69, 8.28%), (Y)
BEBIDA <- CHOCLACTA  (4.02%/69, 73.40%), (Y)
```

```
CHOCLACTA <- BEBIDA  (4.02%/69, 8.15%), (Y)
BEBIDA_NALCOOLICA <- QUEIJO_RALADO  (4.14%/71, 74.74%), (Y)
QUEIJO_RALADO <- BEBIDA_NALCOOLICA  (4.14%/71, 8.52%), (Y)
LEITE_MOCA <- CREME_DE_LEITE_NESTLE  (5.13%/88, 60.69%), (N)
CREME_DE_LEITE_NESTLE <- LEITE_MOCA  (5.13%/88, 46.32%), (N)
PRODUTOS_LIMPEZA <- COCA_COLA DOCES PRODUTOS_HIGIENE  (5.54%/95, 84.82%), (Y)
COCA_COLA <- PRODUTOS_LIMPEZA DOCES PRODUTOS_HIGIENE  (5.54%/95, 54.29%), (Y)
PRODUTOS_BASICOS <- MACARRAO DETERGENTE BISCOITO  (4.25%/73, 91.25%), (Y)
CREME_DE_LEITE_NESTLE <- LEITE_MOCA  (5.13%/88, 46.32%), (N)
COCA_COLA <- CREME_DE_LEITE_NESTLE  (4.31%/74, 51.03%), (N)
CREME_DE_LEITE_NESTLE <- COCA_COLA  (4.31%/74, 21.26%), (N)
PRODUTOS_LIMPEZA <- MACARRAO PAPEL_HIGIENICO BEBIDA  (5.48%/94, 87.85%), (Y)
PAPEL_HIGIENICO <- MACARRAO PRODUTOS_LIMPEZA BEBIDA  (5.48%/94, 71.21%), (Y)
COCA_COLA <- GUARANA_ANTARTICA  (4.95%/85, 68.55%), (N)
GUARANA_ANTARTICA <- COCA_COLA  (4.95%/85, 24.43%), (N)
PRODUTOS_ALIMENTICIOS <- MARGDORIANA  (4.55%/78, 92.86%), (Y)
MARGDORIANA <- PRODUTOS_ALIMENTICIOS  (4.55%/78, 8.40%), (Y)
```

Figura 17: Amostra do conjunto de regras obtidas da segunda base de dados utilizando o algoritmo *AprioriGen* com suporte mínimo igual a 4% e confiança mínima igual a 5% (saidabase2grupo2exp8.rules).

Na próxima subseção os resultados dos experimentos das duas bases de dados são discutidos para mostrar o funcionamento do algoritmo *AprioriGen* proposto e implementado neste trabalho, comparando com a versão original do *Apriori*.

7.3 Comparação e Avaliação dos Resultados

A fim de poder comparar os resultados obtidos nos experimentos realizados, no caso do conjunto de dados da primeira base de dados (base1.data), é apresentado na Tabela 7 as Regras de Associação Generalizadas obtidas do exemplo do relatório técnico de Srikant & Agrawal (1997a).

Tabela 7: Regras de Associação Generalizadas apresentadas em Srikant & Agrawal (1997a).

Regra	**Suporte**	**Confiança**
outerwear ⇒ hiking_boots	33%	66.6%
outerwear ⇒ footwear	33%	66.6%
hiking_boots ⇒ outerwear	33%	100%
hiking_boots ⇒ clothes	33%	100%

Analisando o resultado das regras obtidas com o Experimento 2, Figura 12, utilizando o algoritmo *AprioriGen* (saidabase1grupo1exp2.rules) da primeira base de dados (base1.data), e comparando com o resultado das regras conhecidas da Tabela 7 é possí-

vel verificar que foram geradas as mesmas regras de associação generalizadas esperadas, com exceção das duas primeiras regras da Figura 12, as quais são desconsideradas por não possuírem nenhum item no antecedente da regra, confirmando assim a capacidade do algoritmo em extrair essas regras usando taxonomias.

Observa-se que no Experimento 4 do segundo grupo da base de dados 1 (Figura 14), aparecem regras que foram geradas utilizando itens de todos os níveis da taxonomia. Além das regras generalizadas utilizando os itens presentes na taxonomia também são geradas todas as regras obtidas com a execução do *Apriori* com os mesmos valores de suporte e confiança mínimos.

Em relação ao Experimento 6 do segundo grupo da base de dados 2 (Figura 15), é possível observar que foram geradas um total de 76 regras, sendo obtidas a partir de itens que pertencem as primeiras 14 regras que não possuem itens no antecedente da regra. Sendo assim, as outras 64 regras obtidas são permutações dos itens das primeiras 14 regras.

No caso da base de dados 2, analisando o Experimento 7 do grupo 2 (Figura 16), no qual foi utilizado o algoritmo *Apriori*, foram geradas ao todo 35 regras de associação, ao passo que, utilizando o algoritmo *AprioriGen* no Experimento 8 (Figura 17), foram extraídas um total de 25702 regras entre regras de associação tradicionais e generalizadas com o uso de taxonomias. Portanto, foi constatado que o algoritmo *AprioriGen* continua gerando uma grande quantidade de regras dado a possibilidade do usuário analisar regras usando conceitos mais gerais que as regras geradas pelo algoritmo *Apriori*.

A análise dos resultados obtidos e esperados pode comprovar o funcionamento do algoritmo *AprioriGen*, desenvolvido para obtenção de regras de associação generalizadas com uso de taxonomias durante a etapa de Extração de Padrões. Foi possível notar também que o algoritmo *AprioriGen* gera um número maior de regras que o algoritmo *Apriori*, pois há a inclusão de itens de todos os níveis das taxonomias nas transações durante a execução do algoritmo. Entretanto, há maior possibilidade de que as regras sejam mais interessantes e compreensíveis, com itens gerais indicando uma boa visão do domínio via um conjunto de taxonomias, facilitando, assim, a etapa de Pós-processamento.

7.4 Comparação com o Estado da Arte

Existem muitos trabalhos na literatura que utilizam taxonomias em regras de associação. O foco desses trabalhos difere em função da etapa do processo de mineração

em que as taxonomias são utilizadas. Sendo assim, essa seção apresenta uma revisão desses trabalhos em função dessas etapas.

7.4.1 Pré-processamento

A idéia dos trabalhos que utilizam taxonomias na etapa de pré-processamento é transformar a base de dados de entrada em uma base de dados generalizada. A idéia dos trabalhos de Zhou et al. (2001) e Galindo et al. (2002) é substituir os itens específicos (itens terminais) contidos na base de dados pelos itens gerais (itens não terminais) contidos na taxonomia. Na proposta de Zhou et al. (2001), o usuário pode definir em até que nível ele quer generalizar seus dados. A partir dessa base de dados generalizada é que a técnica de regras de associação é aplicada utilizando um algoritmo tradicional de extração de regras, por exemplo, o *Apriori*.

Em Giha et al. (2006) os autores trabalham com a idéia de base estendida, onde a base de dados contém tanto os itens específicos quantos os gerais, mantendo a mesma idéia utilizada no algoritmo *Basic* proposto por Srikant & Agrawal (1995). A partir dessa base estendida é que o processo de extração de regras é realizado com base no algoritmo *Apriori*, juntamente com algumas técnicas de poda e medidas de significância.

Chen et al. (2003) também trabalham com a idéia de base estendida. Entretanto, nesse caso, a base original já possui alguns itens gerais e a idéia é "arrumar" a base de forma que a mesma fique como a base estendida utilizada no algoritmo *Basic* proposto por Srikant & Agrawal (1995). Nesse caso, essa "arrumação" também pode ser feita a partir de um determinado nível escolhido pelo usuário. Entretanto, a abordagem por eles proposta, diferentemente das abordagens anteriormente descritas, é *top-down*, ou seja, começa dos itens mais gerais e vai descendo para os itens mais específicos.

A fim de comparar os resultados obtidos pelo *AprioriGen* com os obtidos quando taxonomias são utilizadas na etapa de pré-processamento, é apresentado no Apêndice A uma versão modificada do *AprioriGen* que obtém regras generalizadas no pré-processamento. Essa versão segue a mesma idéia dos trabalhos de Zhou et al. (2001) e Galindo et al. (2002).

7.4.2 Extração de Padrões

Os trabalhos relacionados à etapa de extração de padrões se referem aos algoritmos para obtenção de regras de associação generalizadas, como o *AprioriGen* apresentado neste trabalho. A diferença existente entre os algoritmos é a forma como os *itemsets* freqüentes são gerados.

Os primeiros trabalhos referentes a extração de regras de associação generalizadas são de Srikant & Agrawal (1995) (*Basic*, *Cumulate* e *Stratify*) e Han & Fu (1995) (*família ML-T**). As diferenças entre os dois trabalhos são: (1) no primeiro, as regras são obtidas considerando-se um único valor de suporte para todos os níveis e, no segundo, diferentes níveis de suporte são utilizados para cada nível da taxonomia; (2) no primeiro, uma regra pode conter itens de qualquer nível da taxonomia e, no segundo, uma regra só contém itens de um mesmo nível da taxonomia. Vários algoritmos de extração de regras de associação generalizadas surgiram após esses trabalhos, como os apresentados em Hipp et al. (1998) (*Prutax*), Weber (1998) (*Genex*), Baixeries et al. (2000) (*Jer* e *JerM*), Yen (2000) (*GMLAPG*), Yen & Chen (2001) (*GAPG* e *MLAPG*), Rajkumar et al. (2003) (*AprioriNewMulti*), Pramudiono & Kitsuregawa (2004) (*BU-FPtax* e *TD-FPtax*), Sriphaew & Theeramunkong (2004) (*SET* e *cSET*) e Thakur et al. (2006).

7.4.3 Pós-processamento

A idéia dos trabalhos que utilizam taxonomias na etapa de pós-processamento é reduzir o conjunto de regras de entrada de forma a fornecer uma visão mais geral do conhecimento extraído, facilitando, assim, a interpretação e análise do mesmo pelos usuários finais. Adomavicius & Tuzhilin (2001) propõem uma abordagem para realizar o agrupamento de um conjunto de regras (não apenas regras de associação) em função da similaridade existente entre as mesmas. Essa similaridade é medida por uma transformação sintática que se faz nas regras via a utilização de uma taxonomia de domínio.

Domingues & Rezende (2005); Domingues (2004) propõem uma abordagem que obtém regras de associação generalizadas a partir da união de regras específicas com base em uma taxonomia de domínio. Cada uma das regras generalizadas representa um conhecimento mais geral do domínio e engloba, de forma implícita, os conhecimentos específicos.

Em Carvalho et al. (2006) é proposta uma abordagem de pós-processamento de regras de associação, denominada *APRA*, que visa pós-processar um conjunto de regras de associação de forma a obter um conjunto de regras reduzido (compacto) e geral (representativo). Para tanto, utiliza-se conhecimento de domínio, expresso via taxonomias, para que o conjunto de regras seja generalizado.

Uma visão geral da *APRA* é apresentada na Figura 18. Considera-se que os elementos contidos no retângulo pontilhado estão disponíveis, a saber: um conjunto de regras de associação formado somente por regras específicas (regras compostas somente por

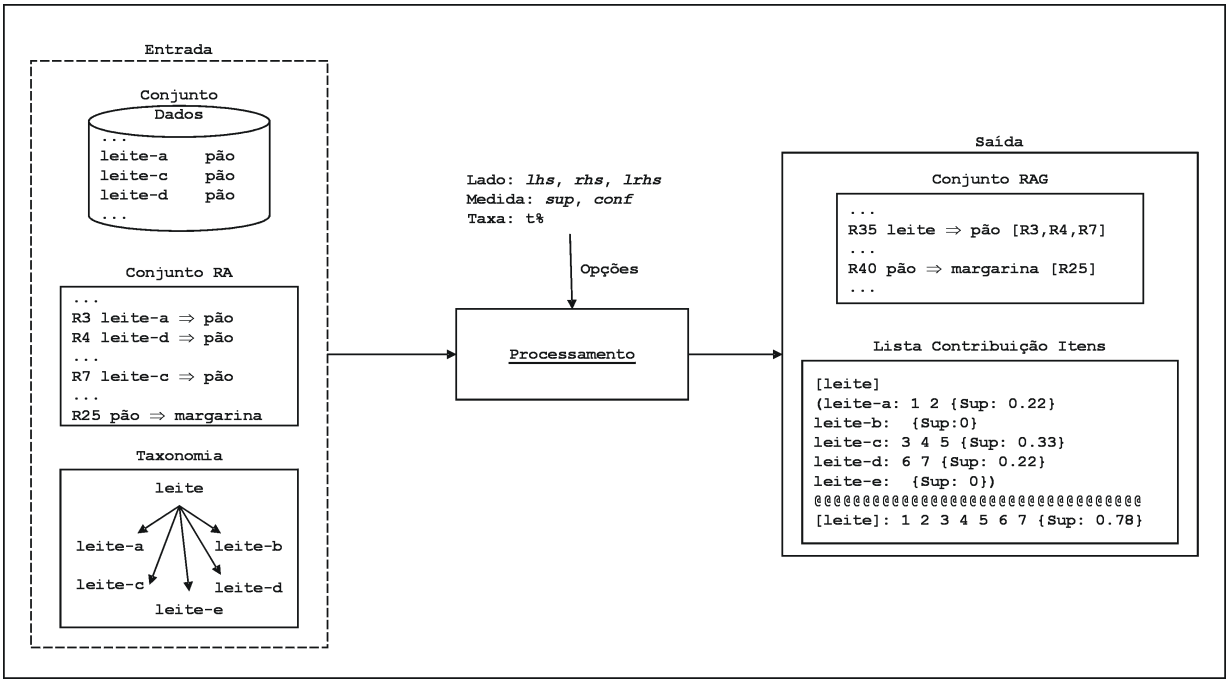

Figura 18: Visão geral da abordagem de pós-processamento de regras de associação (APRA).

itens contidos na base de dados), o conjunto de dados utilizado para gerar as regras específicas e o conhecimento de domínio expresso via taxonomias. Com base nessas entradas é possível obter com a abordagem proposta um conjunto de regras de associação generalizadas composto por regras específicas que não puderam ser generalizadas (por exemplo, regra R40 da Figura 18) e por regras generalizadas obtidas pelo agrupamento de algumas regras específicas via a utilização das taxonomias fornecidas (por exemplo, regra R35 da Figura 18 – regra obtida pelo agrupamento das regras *leite-a* ⇒ *pão* (R3), *leite-d* ⇒ *pão* (R4) e *leite-c* ⇒ *pão* (R7)).

De uma forma mais geral, a *APRA* consiste em pós-processar um conjunto de regras de associação, obtido por um algoritmo tradicional de extração de regras, nesse caso, o *Apriori*, por meio de um processo de generalização com base em taxonomias fornecidas pelo especialista do domínio. Essa generalização pode ser feita em apenas um dos lados da regra (antecedente (*lhs*) ou conseqüente (*rhs*)) ou em ambos os lados (*lrhs*) (opção *Lado* da Figura 18).

Na *APRA*, regras generalizadas são obtidas considerando-se todos ou alguns dos itens contidos na taxonomia, ou seja, é possível transformar regras específicas em regras gerais mesmo que um item geral da regra generalizada não represente todos os itens específicos contidos na taxonomia. Por exemplo: suponha que a regra *leite* ⇒ *pão* represente uma regra generalizada e que leite esteja representado na taxonomia por leite-a, leite-b, leite-c, leite-d e leite-e. A regra *leite* ⇒ *pão* irá existir mesmo que

não exista uma regra específica para cada tipo de leite. Sendo assim, para orientar o usuário na compreensão da regra generalizada, é gerada uma listagem contendo a participação de cada um dos itens específicos na composição dos itens gerais. Por exemplo, a listagem apresentada na Figura 18 é gerada para as entradas consideradas. Para interpretar essa listagem, considere a regra acima descrita ($leite \Rightarrow pão$). A listagem indica que se um determinado item possui 0% de suporte (caso dos itens leite-b e leite-e da Figura 18), ele não estava presente nas transações e, portanto, não contribuiu para o processo de generalização. Assim, essa listagem viabiliza a utilização de taxonomias gerais de domínio em subdomínios. Considere, por exemplo, uma taxonomia que contenha conhecimento sobre produtos alimentícios. Qualquer subdomínio que contenha informações a respeito desses produtos poderá utilizar a mesma taxonomia no processo de generalização, uma vez que identifica-se em uma listagem o suporte de cada um dos itens específicos, ou seja, os itens que contribuíram para o processo de generalização.

É importante ressaltar que a lista de contribuição de itens também favorece a identificação de regras de exceção. Como mencionado, para cada regra generalizada é possível se visualizar, por meio da listagem gerada, a contribuição de cada item específico na composição de cada um dos itens gerais contidos na regra. Assim, tem-se uma descrição mais elaborada do domínio, a partir da qual pode-se observar quais itens específicos não contribuem para uma categoria/classe de itens em uma determinada associação. Por exemplo, com base na listagem apresentada na Figura 18, a regra R35 pode ser interpretada como: "quem compra qualquer tipo de leite, com exceção aos leites leite-b e leite-e, também compra pão", já que o suporte desses tipos de leites é de 0%.

Como uma regra generalizada pode ser gerada sem a presença de todos os itens contidos na taxonomia, para evitar que ocorra uma "sobrecarga" de generalização, um subconjunto de regras específicas só poderá ser substituído por uma regra mais geral se o suporte (sup) ou a confiança ($conf$) da mesma (opção *Medida* da Figura 18) for t% maior do que o maior valor da mesma medida em suas regras específicas (opção *Taxa* da Figura 18). Esse critério pode ser visto como uma variação implícita do suporte/confiança que é explicitamente utilizado em alguns dos trabalhos mencionados na Seção 7.4.2.

A fim de comparar os resultados obtidos pelo *AprioriGen* com os obtidos quando taxonomias são utilizadas na etapa de pós-processamento, é apresentado a seguir alguns experimentos realizados utilizando a *APRA* (Carvalho et al., 2006).

Experimentos com a Base de Dados 1

Utilizando o conjunto de dados (base1.data) e o conjunto de taxonomia (base1tax.tax),

descritos na Seção 7.1, a abordagem *APRA* foi utilizada considerando três opções distintas (notação lado-medida-t%): (a) lhs-sup-0; (b) rhs-sup-0; (c) lrhs-sup-0. Os resultados obtidos encontram-se nas Figuras 19 (opção a), 20 (opção b) e 21 (opção c). A listagem de contribuição de itens, apresentada na Figura 22, é igual para todas as configurações, uma vez que a base de dados e a taxonomia consideradas é a mesma.

```
[R0001],hiking_boots,ski_pants,[0.166667,0.166667,0.666667,0.000000,6],[R0002]
[R0002],hiking_boots,jacket,[0.166667,0.166667,0.500000,0.166667,6],[R0004]
[R0003],outerwear,hiking_boots,[0.333333,0.166667,0.500000,0.000000,6],[R0001,R0003]
```

Figura 19: Conjunto de regras generalizadas obtidas pela *APRA* utilizando a opção lhs-sup-0.

```
[R0001],ski_pants,hiking_boots,[0.166667,0.000000,0.666667,0.166667,6],[R0001]
[R0002],jacket,hiking_boots,[0.166667,0.166667,0.500000,0.166667,6],[R0003]
[R0003],hiking_boots,outerwear,[0.333333,0.000000,0.500000,0.166667,6],[R0002,R0004]
```

Figura 20: Conjunto de regras generalizadas obtidas pela *APRA* utilizando a opção rhs-sup-0.

```
[R0001],footwear,outerwear,[0.333333,0.333333,0.166667,0.166667,6],[R0002,R0004]
[R0002],outerwear,footwear,[0.333333,0.166667,0.166667,0.333333,6],[R0001,R0003]
```

Figura 21: Conjunto de regras generalizadas obtidas pela *APRA* utilizando a opção lrhs-sup-0.

```
[outerwear]
(jacket: 2 6 {Sup: 0.333333333333333}
ski_pants: 3 {Sup: 0.166666666666667})

[footwear]
(shoes: 4 5 {Sup: 0.333333333333333}
hiking_boots: 2 3 {Sup: 0.333333333333333})

[clothes]
(outerwear: 2 3 6 {Sup: 0.5}
shirt:  {Sup: 0})

@@@
```

A sintaxe utilizada nas Figuras 19, 20 e 21 para representar as regras generalizadas possui o seguinte formato:

```
[Número da Regra], Antecedente, Conseqüente,
                   [Dados da Matriz de Contingência],
                   [Identificador(es) da(s) Regra(s) de Origem]
```

A lista de identificadores das regras de origem deve ser interpretada de duas maneiras distintas: (a) se a lista possuir apenas um identificador, a regra representa uma regra específica (não generalizada), e o identificador indica o número da regra no conjunto de entrada – conjunto de regras utilizado para gerar o conjunto de saída; (b) se a lista possuir dois ou mais identificadores, a regra representa uma regra generalizada, e os identificadores indicam os números das regras no conjunto de entrada – conjunto de regras utilizado para gerar o conjunto de saída – que deram origem a mesma. Por exemplo, a regra [R0001] da Figura 19 representa uma regra específica, e o seu identificador, [R0002], o número que a mesma encontra-se no conjunto de regras da Figura 13, ou seja, a regra permaneceu inalterada já que a mesma não pode ser generalizada. Já a regra [R0003] da Figura 19 representa uma regra generalizada, e o seus identificadores, [R0001] e [R0003], os números das regras específicas contidas no conjunto de regras da Figura 13 que deram origem a mesma.

O que se pôde observar com base nos experimentos realizados é que quando taxonomias são utilizadas na etapa de pós-processamento utilizando a *APRA* (Carvalho et al., 2006), um conjunto compacto e representativo de regras pode ser obtido. Nesse caso, embora o usuário tenha uma visão geral do domínio, ele ainda tem acesso as regras específicas (regras que foram agrupadas a fim de gerar a regra generalizada) através da lista de identificadores descrita anteriormente. Além disso, a abordagem contribui na identificação das regras gerais e suas exceções (itens específicos relacionados aos itens gerais da regra generalizada) através da listagem de contribuição de itens.

8 Considerações Finais

Este relatório apresentou uma descrição do Algoritmo *AprioriGen*, que gera Regras de Associação Generalizadas a partir de taxonomias durante a etapa de Extração de Padrões no Processo de Extração de Conhecimento de Bases de Dados. Para tanto, neste trabalho, foi apresentada uma breve descrição de conceitos sobre Regras de Associação e o uso de taxonomias no desenvolvimento de algoritmos que extraem Regras de Associação Generalizadas na etapa de Extração de Padrões no processo de

Mineração de Dados. Entre os algoritmos que extraem Regras de Associação destacou-se o algoritmo *Apriori* proposto por Agrawal & Srikant (1994), servindo como base para o desenvolvimento do algoritmo *AprioriGen*. Foram apresentados outros algoritmos que também geram Regras de Associação Generalizadas com uso de taxonomias como o Básico e o Cumulate, que auxiliaram no entendimento do uso de taxonomias na geração de Regras de Associação Generalizadas para o desenvolvimento do algoritmo *AprioriGen*.

Em relação aos experimentos realizados, todos foram efetuados com o objetivo de comprovar a obtenção de Regras de Associação Generalizadas com o algoritmo *AprioriGen* utilizando taxonomias durante a etapa de Extração de Padrões.

É importante destacar que durante a fase de alteração da implementação foram realizados vários testes com diferentes tipos de bases de dados em diversas condições de execução. Através desses testes foi possível validar e verificar se as modificações adicionais implementadas não comprometiam o funcionamento normal do programa, e se o mesmo continuava a desempenhar corretamente as funções esperadas a cada novo procedimento incluído.

Como mencionado anteriormente, existem muitos trabalhos na literatura que utilizam taxonomias em regras de associação. O foco desses trabalhos difere em função da etapa do processo de mineração em que as taxonomias são utilizadas. Assim, este trabalho também apresentou uma comparação dos resultados obtidos pelo *AprioriGen* com um trabalho que utiliza taxonomias no pós-processamento (Carvalho et al., 2006) e um que utiliza taxonomias no pré-processamento (trabalho aqui apresentado – *AprioriGen* versão 2). Entretanto, é importante mencionar que segundo Pohle (2003), a utilização de taxonomias na etapa de pré-processamento requer uma grande preparação de dados voltada especificamente ao problema, isto é, todo o processo de mineração freqüentemente deve ser reinicializado quando uma modificação, mesmo que sutil, ocorre nos objetivos da análise. Além disso, quando taxonomias são utilizadas no pré-processamento, o conhecimento específico contido na base de dados é perdido, não podendo ser explorado posteriormente junto com o conhecimento geral obtido. Ainda segundo Pohle (2003) a utilização de taxonomias na extração de padrões não trata o problema da interpretação dos padrões descobertos, mas sim o processo em si. Nesse caso, embora se tenha uma visão tanto geral como específica do domínio, o volume de regras se torna muito maior, o que dificulta ainda mais a exploração do domínio pelo usuário final. Sendo assim, é mais interessante que as taxonomias sejam utilizadas na etapa de pós-processamento, pois a utilização de algum conhecimento de fundo pode melhorar a análise dos padrões obtidos.

A *AprioriGen* no Pré-processamento

Neste apêndice é apresentado o algoritmo *AprioriGen* versão 2, algoritmo voltado para a utilização de taxonomias na etapa de pré-processamento. Nessa versão, ao contrário da primeira versão descrita na Figura 3, ao invés de realizar a inclusão do item pai roupas leves nas transações que ocorrem os itens camiseta e bermuda, há a troca dos itens camiseta e bermuda pelo item roupas leves. Na segunda versão (*AprioriGen* v.2), o procedimento do algoritmo Básico é alterado de modo a modificar as transações carregadas em memória, ocorrendo a troca dos itens filhos pelos itens pais de máxima generalização nas transações lidas em memória.

A.1 *AprioriGen* versão 2

A segunda versão do *AprioriGen* é apresentada no Algoritmo 5. Nessa versão, o procedimento do Algoritmo Básico (Srikant & Agrawal, 1995) foi alterado de modo a incluir nas transações somente o ancestral de máxima generalização do item específico na taxonomia, removendo o item específico da transação. A melhoria número 3 do Algoritmo Cumulate (Srikant & Agrawal, 1995) permanece como no algoritmo *AprioriGen* anterior.

Como no algoritmo da versão anterior as linhas sublinhadas indicam os locais que foram alterados. Inicialmente o Algoritmo *AprioriGen* versão 2 percorre cada item a de cada transação t do conjunto de transações T da base de dados D, armazenando o item a na variável aux e atribuindo à variável $flag$ o valor zero. Em seguida, enquanto o item aux for um item específico (item filho) de um item geral b (item pai) em alguma taxonomia no conjunto de taxonomias Tax, procura-se o próximo item pai no nível acima de b e sinaliza $flag$ com 1. Ao final do loop ($flag$ igual a 1) a variável aux estará com o pai de maior nível do item a (generalização máxima de a). A seguir, substitui-se o item a da transação pelo item aux na transação t.

Essa primeira parte corresponde ao procedimento do Algoritmo Básico (Srikant & Agrawal, 1995), descrito na Seção 4.2, com a alteração descrita. Depois de percorridas todas as transação da base de dados substituindo os itens específicos pelos itens gerais (linhas 1 a 14 do Algoritmo 5), a segunda parte do algoritmo procede sua execução exatamente como o Algoritmo *Apriori* anterior. Assim, o algoritmo 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 17 do Algoritmo 5) o conjunto de *itemsets* freqüentes L_{k-1} obtido no passo (k - 1) é utilizado para gerar o conjunto de 1-*itemsets* candidatos C_k usando a função apriori-gen (descrito

na Seção 3.1.1). A seguir (linhas 18 a 23 do Algoritmo 5), 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 24 do Algoritmo 5).

Algoritmo 5 *AprioriGen* versão 2.

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

Assegura: Um conjunto *Resposta* contendo todos os *k-itemsets* freqüentes.

1: **para todo** (transações $t \in T$) **faça**
2: **para todo** (itens $a \in t$) **faça**
3: $flag := 0$; //Indica se entrou ou não no loop
4: $aux := a$; //Armazena o item a
5: **enquanto** item aux possuir item pai $b \in Tax$ **faça**
6: $aux := b$
7: $flag := 1$
8: **fim-enquanto**
9: **se** $flag = 1$ **então**
10: remove item a de t; //remove o item específico (item filho) da transação
11: $t := t + aux$; //adiciona o item geral (item pai) de nível mais alto na taxonomia
12: **fim-se**
13: **fim-para**
14: **fim-para**
15: $L_1 := \{$*1-itemsets* freqüente$\}$;
16: **para** $(k := 2; L_{k-1} \neq \varnothing; k++)$ **faça**
17: $C_k :=$ apriori-gen(L_{k-1}); //Gera novos conjuntos candidatos
18: **para todo** (transações $t \in T$) **faça**
19: $C_t :=$ subset(C_k,t); //Conjuntos candidatos contidos em t
20: **para todo** candidatos $c \in C_t$ **faça**
21: c.count++;
22: **fim-para**
23: **fim-para**
24: $L_k := \{c \in C_k \mid c.count \geq$ **sup-min**$\}$;
25: **se** $k = 2$ **então**
26: **para todo** (itemset $x \in L_{k=2}$) **faça**
27: **se** $item_1 \in x$ for parente de $item_2 \in x$ OU $item_2 \in x$ for parente de $item_1 \in x$ **então**
28: retirar itemset x do conjunto $L_{k=2}$;
29: **fim-se**
30: **fim-para**
31: **fim-se**
32: **fim-para**
33: Resposta $:= \bigcup_k L_k$

Na última etapa (linhas 25 a 32 do Algoritmo 5), como na versão anterior do *Apriori-Gen* (Algoritmo 4), é acrescentada uma verificação no passo $k = 2$ para garantir que

no conjunto de *itemsets* freqüentes $L_{k=2}$ os 2 itens do *itemset* não possuam nenhum parentesco um com o outro em qualquer que seja o nível da hierarquia entre eles no conjunto de taxonomias definidas. Essa etapa corresponde a melhoria número 3 do algoritmo Cumulate (Srikant & Agrawal, 1995) (remover *itemsets* que contenham apenas um item e seu ancestral), descrita na Seção 4.3.

A solução final é dada pela união dos conjuntos L_k de *k-itemsets* freqüentes (linha 33). Essa solução é utilizada como entrada para algum algoritmo que gera regras de associação a partir do *itemset* (por exemplo, o descrito na Seção 3.2).

A seguir, são descritos os experimentos realizados utilizando a segunda versão do algoritmo *AprioriGen*.

A.2 Experimentos Realizados

Como nos experimentos anteriores, a fim de possibilitar a comparação e comprovação das funcionalidades do algoritmo *AprioriGen* versão 2, foram realizados dois experimentos para cada uma das duas bases de dados, variando a taxa de suporte e confiança mínimas.

A.2.1 Experimento com a Base de Dados 1

Assim como nos experimentos anteriores, na primeira base de dados utilizando o conjunto de dados (base1.data) e o conjunto de taxonomia (base1tax.tax), foi efetuada a execução da implementação do algoritmo *AprioriGen* versão 2 usando a opção de linha de comando apresentada a seguir.

- Experimento 9 – Para a execução do Algoritmo *AprioriGen* versão 2:

```
aprioriGenV2.exe -c60 -s30 -o -a -p%.2f%% -w base1.data saidabase1v2exp9.rules base1tax.tax
```

Nesse Experimento 9, assim como nos Experimentos 1 e 2, foi definido o suporte mínimo igual a 30% e confiança mínima igual a 60%, de acordo com o exemplo do relatório técnico de Srikant & Agrawal (1997a), na qual esse experimento se baseia.

Na Figura 23 é apresentado o arquivo de saída (.rules) com as regras obtidas após a execução da segunda versão do algoritmo *AprioriGen* referente ao Experimento 9. Nesse experimento foram geradas apenas 2 regras sem o antecedente (LHS), uma vez que esta versão não considera os itens intermediários da taxonomia.

```
clothes <-  (66.67%/4, 66.67%), (Y)
footwear <-  (66.67%/4, 66.67%), (Y)
```

Figura 23: Conjunto de regras obtidas da primeira base de dados utilizando o algoritmo *AprioriGen* v.2 (saidabase1v2exp9.rules).

Com esta base também foi executado um segundo experimento com a segunda versão do algoritmo *AprioriGen*, definindo um valor de 10% para suporte mínimo e 50% para confiança mínima.

- Experimento 10 – Para a execução do Algoritmo *AprioriGen* versão 2:

aprioriGenV2.exe -c50 -s10 -o -a -p%.2f%% -w base1.data saidabase1v2exp9.rules base1tax.tax

Na Figura 24 é apresentado o arquivo de saída (.rules) com as regras obtidas após a execução da segunda versão do algoritmo *AprioriGen* referente ao Experimento 10.

```
clothes <-  (66.67%/4, 66.67%), (Y)
footwear <-  (66.67%/4, 66.67%), (Y)
footwear <- clothes  (33.33%/2, 50.00%), (Y)
clothes <- footwear  (33.33%/2, 50.00%), (Y)
```

Figura 24: Conjunto de regras obtidas utilizando o algoritmo *AprioriGen* v.2 com suporte mínimo igual a 10% e confiança mínima igual a 50% (saidabase1v2exp9.rules).

A.2.2 Experimento com a Base de Dados 2

Na segunda base de dados, utilizando o conjunto de dados (base2.data) e o conjunto de taxonomia (base2tax.tax), foi efetuada a execução da implementação do algoritmo *AprioriGen* versão 2 usando a opção de linha de comando apresentada a seguir.

- Experimento 11 – Para a execução do Algoritmo *AprioriGen* versão 2:

aprioriGenV2.exe -c15 -s15 -o -a -p%.2f%% -w base2.data saidabase2v2exp11.rules base2tax.tax

Nesse Experimento 11, assim como no primeiro grupo de experimentos da segunda base de dados (Experimentos 5 e 6), foi definido suporte mínimo igual a 15% e confiança

mínima igual a 15%. Na Figura 25 é apresentado o arquivo de saída (.rules) com as regras obtidas após a execução da segunda versão do algoritmo *AprioriGen* referente ao Experimento 11.

```
PRODUTOS_LIMPEZA <-  (21.91%/376, 21.91%), (Y)
PRODUTOS_HIGIENE <-  (26.05%/447, 26.05%), (Y)
BEBIDA <-  (49.36%/847, 49.36%), (Y)
PRODUTOS_ALIMENTICIOS <-  (54.14%/929, 54.14%), (Y)
PRODUTOS_HIGIENE <- PRODUTOS_LIMPEZA  (15.33%/263, 69.95%), (Y)
PRODUTOS_LIMPEZA <- PRODUTOS_HIGIENE  (15.33%/263, 58.84%), (Y)
BEBIDA <- PRODUTOS_LIMPEZA  (15.73%/270, 71.81%), (Y)
PRODUTOS_LIMPEZA <- BEBIDA  (15.73%/270, 31.88%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_LIMPEZA  (18.88%/324, 86.17%), (Y)
PRODUTOS_LIMPEZA <- PRODUTOS_ALIMENTICIOS  (18.88%/324, 34.88%), (Y)
BEBIDA <- PRODUTOS_HIGIENE  (19.00%/326, 72.93%), (Y)
PRODUTOS_HIGIENE <- BEBIDA  (19.00%/326, 38.49%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_HIGIENE  (21.68%/372, 83.22%), (Y)
PRODUTOS_HIGIENE <- PRODUTOS_ALIMENTICIOS  (21.68%/372, 40.04%), (Y)
PRODUTOS_ALIMENTICIOS <- BEBIDA  (32.63%/560, 66.12%), (Y)
BEBIDA <- PRODUTOS_ALIMENTICIOS  (32.63%/560, 60.28%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_LIMPEZA BEBIDA  (15.09%/259, 95.93%), (Y)
BEBIDA <- PRODUTOS_LIMPEZA PRODUTOS_ALIMENTICIOS  (15.09%/259, 79.94%), (Y)
PRODUTOS_LIMPEZA <- BEBIDA PRODUTOS_ALIMENTICIOS  (15.09%/259, 46.25%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_HIGIENE BEBIDA  (17.31%/297, 91.10%), (Y)
BEBIDA <- PRODUTOS_HIGIENE PRODUTOS_ALIMENTICIOS  (17.31%/297, 79.84%), (Y)
PRODUTOS_HIGIENE <- BEBIDA PRODUTOS_ALIMENTICIOS  (17.31%/297, 53.04%), (Y)
```

Figura 25: Conjunto de regras obtidas da segunda base de dados utilizando o algoritmo *AprioriGen* v.2 (saidabase2v2exp11.rules).

Com esta base também foi executado um segundo experimento com a segunda versão do algoritmo *AprioriGen*, definindo um valor de 4% para suporte mínimo e 5% para confiança mínima.

- Experimento 12 – Para a execução do Algoritmo *AprioriGen* versão 2:

aprioriGenV2.exe -c5 -s4 -o -a -p%.2f%% -w base2.data saidabase2v2exp12.rules base2tax.tax

Na Figura 26 é apresentado o arquivo de saída (.rules) com as regras obtidas após a execução da segunda versão do algoritmo *AprioriGen* referente ao Experimento 12. Nesse experimento é apresentado somente uma pequena amostra do conjunto de regras, devido a grande quantidade de regras obtidas (218 regras).

```
PRODUTOS_ALIMENTICIOS <-  (54.14%/929, 54.14%), (Y)
BEBIDA <- PRODUTOS_ALIMENTICIOS  (32.63%/560, 60.28%), (Y)
PRODUTOS_ALIMENTICIOS <- MARGDORIANA BEBIDA  (4.08%/70, 97.22%), (Y)
BEBIDA <- MARGDORIANA PRODUTOS_ALIMENTICIOS  (4.08%/70, 89.74%), (Y)
MARGDORIANA <- BEBIDA PRODUTOS_ALIMENTICIOS  (4.08%/70, 12.50%), (Y)
PRODUTOS_HIGIENE <- LEITE_MOCA PRODUTOS_LIMPEZA BEBIDA  (6.53%/112, 96.55%), (Y)
PRODUTOS_LIMPEZA <- LEITE_MOCA PRODUTOS_HIGIENE BEBIDA  (6.53%/112, 86.15%), (Y)
PRODUTOS_ALIMENTICIOS <- SAL_CISNE  (5.48%/94, 91.26%), (Y)
SAL_CISNE <- PRODUTOS_ALIMENTICIOS  (5.48%/94, 10.12%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_LIMPEZA BEBIDA  (15.09%/259, 95.93%), (Y)
BEBIDA <- PRODUTOS_LIMPEZA PRODUTOS_ALIMENTICIOS  (15.09%/259, 79.94%), (Y)
PRODUTOS_LIMPEZA <- BEBIDA PRODUTOS_ALIMENTICIOS  (15.09%/259, 46.25%), (Y)
PRODUTOS_ALIMENTICIOS <- PRODUTOS_HIGIENE BEBIDA  (17.31%/297, 91.10%), (Y)
PRODUTOS_LIMPEZA <- LEITE_MOCA PRODUTOS_HIGIENE BEBIDA  (6.53%/112, 86.15%), (Y)
LEITE_MOCA <- PRODUTOS_LIMPEZA PRODUTOS_HIGIENE BEBIDA  (6.53%/112, 49.78%), (Y)
PRODUTOS_ALIMENTICIOS <- MARGDORIANA  (4.55%/78, 92.86%), (Y)
MARGDORIANA <- PRODUTOS_ALIMENTICIOS  (4.55%/78, 8.40%), (Y)
LEITE_MOCA <- CREME_DE_LEITE_NESTLE  (5.13%/88, 60.69%), (N)
CREME_DE_LEITE_NESTLE <- LEITE_MOCA  (5.13%/88, 46.32%), (N)
BEBIDA <- MAIONESE_HELLMANN_S PRODUTOS_HIGIENE  (4.31%/74, 91.36%), (Y)
PRODUTOS_HIGIENE <- MAIONESE_HELLMANN_S BEBIDA  (4.31%/74, 82.22%), (Y)
MAIONESE_HELLMANN_S <- PRODUTOS_HIGIENE BEBIDA  (4.31%/74, 22.70%), (Y)
LEITE_MOCA <- BEBIDA  (9.32%/160, 18.89%), (Y)
```

Figura 26: Amostra do conjunto de regras obtidas utilizando o algoritmo *AprioriGen* v.2 com suporte mínimo igual a 4% e confiança mínima igual a 5% (saidabase2v2exp12.rules).

A seguir, são discutidos e analisados os resultados obtidos com os experimento utilizando o algoritmo *AprioriGen* versão 2.

A.3 Análise dos Resultados

Analisando o resultado das regras obtidas com o Experimento 9, Figura 23, utilizando o algoritmo *AprioriGen* versão 2 (saidabase1v2exp9.rules), obteve-se como resultado duas regras que não possuem itens no antecedente, não sendo possível ter uma avaliação. No entanto, diminuindo o valor da confiança mínima para 50% foram obtidas mais duas regras: clothes $\Rightarrow$ footwear e footwear $\Rightarrow$ clothes (Figura 24). Essas regras mostram que a segunda versão do algoritmo *AprioriGen* está funcionando como esperado, de modo a incluir nas transações somente o ancestral (item geral) de máxima generalização dos itens específicos contidos na taxonomia, ocorrendo a troca dos itens filhos (itens específicos) pelos itens pais (itens gerais) de máxima generalização nas transações.

Com o Experimento 11, Figura 25, utilizando a segunda base de dados (base2.data), foram obtidas 22 regras (saidabase2v2exp11.rules), apresentando uma redução de 29% em

relação ao Experimento 6, Figura 15, que utiliza a primeira versão algoritmo *AprioriGen*, sendo em sua maioria Regras de Associação Generalizadas.

Referências Bibliográficas

Adamo, J.-M. (2001). *Data Mining for Association Rules and Sequential Patterns*. New York, NY: Springer-Verlag. 2, 12, 14

Adomavicius, G. & A. Tuzhilin (2001). Expert-driven validation of rule-based user models in personalization applications. *Data Mining and Knowledge Discovery 5*(1/2), 33–58. 38

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. 4

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. 2, 3, 6, 7, 8, 11, 23, 43

Baixeries, J., G. Casas, & J. L. Balcázar (2000). Frequent sets, sequences, and taxonomies: New, efficient algorithmic proposals. *Technical Report* LSI-00-78-R, Departament de LSI – Universitat Politècnica de Catalunya. Disponível em: `http://citeseer.ist.psu.edu/428618.html` [02/03/2007]. 38

Borgelt, C. & R. Kruse (2002). Induction of association rules: Apriori implementation. Em *Proceedings of the 15th Conference on Computational Statistics*, Germany, Berlim. 22, 23, 25, 26, 29

Carvalho, V. O., S. O. Rezende, & M. Castro (2006). Regras de associação generalizadas: Obtenção e avaliação. Em *Proceedings of the II Workshop em Algoritmos e Aplicações de Mineração de Dados (WAAMD-2006) – SBBD/SBES*, pp. 81–88. 38, 40, 42, 43

Chen, X., X. Zhou, R. Scherl, & J. Geller (2003). Using an interest ontology for improved support in rule mining. Em Y. Kambayashi, M. K. Mohania, & W. Wöß (eds.), *5th International Conference on Data Warehousing and Knowledge Discovery, DaWaK 2003*, Volume 2737, pp. 320–329. Springer-Verlag. 37

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. 2

Domingues, M. A. (2004). Generalização de regras de associação. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação – USP – São Carlos. 38

Domingues, M. A. & S. O. Rezende (2005). Using taxonomies to facilitate the analysis of the association rules. Em *Proceedings of ECML/PKDD'05 – The Second International Workshop on Knowledge Discovery and Ontologies, KDO 2005*, pp. 59–66. 38

Fayyad, U. M., G. Piatetsky-Shapiro, & P. Smyth (1996). 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. 1

Galindo, M. G. L., P. S. S. Coelho, & G. Lachtermacher (2002). Usando taxonomias para criar regras de associação: Aplicação em uma base de dados transacional. Em *XI Latin-Iberian American Congress of Operations Research, CLAIO 2002*, Volume 1, pp. 102–110. 37

Giha, F. E., Y. P. Singh, & H. T. Ewe (2006). Mining generalized customer profiles. Em *International Conference on Artificial Intelligence and Machine Learning, AIML 2006*, Volume 1, pp. 141–147. 37

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, VLDB 1995*, pp. 420–431. Disponível em: `http://citeseer.nj.nec.com/han95discovery.html` [13/07/2004]. 38

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 1998*, pp. 74–82. Disponível em: `http://citeseer.nj.nec.com/hipp98new.html` [13/07/2004]. 38

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. 2, 13

Pohle, D.-K. C. (2003). Integrating and updating domain knowledge with data mining. Em M. H. Scholl & T. Grust (eds.), *Proceedings of the International Conference on Very Large Data Bases, VLDB 2003, PhD Workshop*, Volume 76. CEUR Workshop Proceedings (`CEUR-WS.org`). 5p. Disponível em: `CEUR-ws.org/Vol-76/pohle.pdf` [23/01/2007]. 43

Pramudiono, I. & M. Kitsuregawa (2004). FP-tax: Tree structure based generalized association rule mining. Em *Proceedings of the 9th ACM SIGMOD – Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD 2004*, New York, NY, USA, pp. 60–63. ACM Press. 38

Rajkumar, N., M. R. Karthik, & S. N. Sivanandam (2003). Fast algorithm for mining multilevel association rules. Em *IEEE TENCON 2003 Conference on Convergent Technologies for the Asia-Pacific Region*, Volume 2, pp. 688–692. 38

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. 2

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/~hilario/icdm-01/DM-KM-Final/Semenova.pdf`. Acesso em: 11/12/2002. 2

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. 14

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. 13

Srikant, R. & R. Agrawal (1995). Mining generalized association rules. Em *Proceedings of the 21th International Conference on Very Large Data Bases, VLDB 1995*, pp. 407–419. 15, 16, 19, 37, 38, 44, 46

Srikant, R. & R. Agrawal (1997a). Mining generalized association rules. Relatório Técnico, IBM Research Division. Disponível em: `http://www.almaden.ibm.com/cs/projects/iis/hdb/Publications/papers/vldb95_tax_rj.pdf`. Acesso em: 28/02/2007. iii, 26, 29, 30, 35, 46

Srikant, R. & R. Agrawal (1997b). Mining generalized association rules. *Future Generation Computer Systems 13*(2–3), 161–180. 13, 14, 15

Sriphaew, K. & T. Theeramunkong (2004). Fast algorithms for mining generalized frequent patterns of generalized association rules. *IEICE Transactions on Information & Systems E87-D*(3), 761–770. 38

Thakur, R. S., R. C. Jain, & K. R. Pardasani (2006). Mining level-crossing association rules from large databases. *Journal of Computer Science 2*(1), 76–81. 38

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 1998*. 8p. Disponível em: `http://citeseer.nj.nec.com/25197.html` [13/07/2004]. 38

Yen, S.-J. (2000). Mining generalized multiple-level association rules. Em *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD 2000*, Volume 1910, pp. 679–684. Springer-Verlag. 38

Yen, S.-J. & A. Chen (2001). A graph-based approach for discovering various types of association rules. *IEEE Transactions on Knowledge and Data Engineering 13*(5), 839–845. 38

Zhang, C. & S. Zhang (2002). *Association Rule Mining: Models and Algorithms*, Volume 2307. Springer. 3

Zhou, H., B. Ruan, J. Zhu, Y. Zhu, & B. Shi (2001). ARMiner: A data mining tool based on association rules. Em X. S. Wang, G. Yu, & H. Lu (eds.), *The Second International Conference on Web-Age Information Management, WAIM 2001*, Volume 2118, pp. 114–121. Springer-Verlag. 37