Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

DESCRIÇÃO DO MÓDULO DE EXPLORAÇÃO DE REGRAS DE ASSOCIAÇÃO GENERALIZADAS RULEE-RAG

MAGALY LIKA FUJIMOTO
VERÔNICA OLIVEIRA DE CARVALHO
SOLANGE OLIVEIRA REZENDE

Nº 296

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Abril/2007

Descrição do Módulo de Exploração de Regras de Associação Generalizadas RulEE-RAG

Magaly Lika Fujimoto
Verônica Oliveira de Carvalho
Solange Oliveira Rezende

Departamento de Ciências de Computação
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - Campus de São Carlos
Caixa Postal 668
13560-970, São Carlos, SP
e-mail: {mlika, veronica, solange}@icmc.usp.br

Resumo:

A mineração de dados é um processo de natureza iterativa e interativa responsável por identificar padrões em grandes conjuntos de dados objetivando extrair conhecimento válido, útil e inovador a partir dos mesmos. Dentre as técnicas de mineração de dados que vem recebendo grande destaque nos últimos anos está a de regras de associação. Embora essa técnica seja muito útil por identificar todas as associações intrínsecas que estejam contidas nos dados, a mesma possui o inconveniente de gerar uma grande quantidade de regras dificultando a interpretação das mesmas por parte dos usuários. Na tentativa de obter conjuntos de regras mais gerais a fim facilitar a compreensão dos mesmos pelos usuários, taxonomias vem sendo utilizadas. Assim, as regras de associação generalizadas proporcionam uma visão mais geral do conhecimento descoberto, enquanto as regras específicas (menos gerais), podem ser exploradas para maiores detalhes. Neste contexto, neste relatório técnico é descrito um módulo para exploração de regras de associação generalizadas na etapa de pós-processamento do conhecimento.

Abr./2007

Sumário

1. Introdução

O processo de extração de conhecimento de base de dados ou mineração de dados (MD) tem como objetivo encontrar conhecimento a partir de um conjunto de dados para ser utilizado em um processo de tomada de decisão. Portanto, esse conhecimento deve ser avaliado quanto a sua validade, compreensibilidade e interessabilidade durante a etapa de pós-processamento. Um problema desta etapa é que muitos dos algoritmos de extração geram uma enorme quantidade de padrões (Padmanabhan & Tuzhilin, 2000).

Este problema geralmente ocorre na extração de regras de associação devido a sua característica de descobrir as associações existentes nas transações em uma base de dados. Esta característica determina a geração de um número surpreendente de regras, dificultando sobremaneira a interpretação do conjunto de regras pelo usuário. Segundo Hilderman & Hamilton (2000); Ma, Wong, & Liu (2000), interpretar o conhecimento adquirido de modo a obter um bom entendimento sobre o domínio de aplicação é uma das etapas mais importantes do processo de mineração de dados. Desse modo, é importante que hajam abordagens que reduzam a quantidade de regras obtidas de forma a facilitar a análise das mesmas pelos usuários.

Assim, Carvalho, Rezende, & Castro (2006, 2007) propõem uma abordagem que utiliza conhecimento de domínio, expresso via taxonomias, para obter um número reduzido de regras. Nesse conjunto reduzido de regras podem ser encontradas regras mais gerais, chamadas de regras de associação generalizadas, as quais fornecem uma visão mais geral do conhecimento descoberto, quanto regras menos gerais (específicas), as quais podem ser exploradas posteriormente para maiores detalhes.

É importante ressaltar que para auxiliar o usuário na compreensão e utilização do conhecimento descoberto, já existe um protótipo de um ambiente para exploração de regras, denominado RulEE (*Rule Exploration Environment*), projetado e implementado no Laboratório de Inteligência Computacional (LABIC*) do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP†) (Paula, 2003). O RulEE é voltado para exploração de regras durante a etapa de pós-processamento e disponibilização do conhecimento, devido à necessidade de identificação de conhecimento interessante e da participação de usuários especialistas do domínio na exploração do conhecimento. O RulEE viabiliza tanto a análise quanto a disponibilização de regras de classificação, regressão e associação. É um ambiente interativo no qual o usuário pode "navegar" no conjunto de regras obtido de modo a selecionar, por meio de medidas de avaliação e con-

*http://labic.icmc.usp.br.
†http://www.icmc.usp.br.

sultas SQL, as regras que se apresentam mais interessantes. O módulo RulEE-RAG, descrito neste relatório técnico, foi implementado e integrado ao ambiente RulEE.

Neste contexto, neste relatório técnico é descrito o módulo para exploração de regras de associação generalizadas RulEE-RAG, que auxilia na abordagem proposta por Carvalho, Rezende, & Castro (2006, 2007). Assim, este relatório está organizado da seguinte maneira: na Seção 2 é realizada uma breve descrição de regras de associação e regras de associação generalizadas. Na Seção 3 é apresentada a abordagem de pós-processamento regras de associação proposto por Carvalho, Rezende, & Castro (2006, 2007). Na Seção 4 é apresentado o funcionamento do módulo RulEE-RAG. Também são descritas as mudanças realizadas na base de dados e as classes implementadas. Finalmente, na Seção 5 são realizadas as considerações finais.

2. Regras de Associação e Regras de Associação Generalizadas

Associação é uma tarefa de mineração de dados classificada como descritiva (Weiss & Indurkhya, 1998; Rezende, Pugliesi, Melanda, & Paula, 2003). Essa tarefa visa descobrir o quanto um conjunto de itens presentes em um registro de uma base de dados implica na presença de algum outro conjunto distinto de itens no mesmo registro (Agrawal & Srikant, 1994). Portanto, com a extração de regras de associação é possível encontrar tendências que possam ser usadas para entender e explorar padrões de comportamento dos dados.

O formato de uma regra de associação pode ser representado como uma implicação na forma $LHS \Rightarrow RHS$, em que LHS e RHS são, respectivamente, o lado esquerdo (*Left Hand Side*) e o lado direito (*Right Hand Side*) da regra, definidos por conjuntos disjuntos de itens. 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 $LHS \cap RHS = \oslash$. A regra $LHS \Rightarrow RHS$ ocorre no conjunto de transações T com confiança $conf$ se em $conf\%$ das transações

de T em que LHS ocorre, RHS também ocorre. A regra $LHS \Rightarrow RHS$ tem suporte sup se em $sup\%$ das transações em T ocorrem $LHS \cup RHS$.

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

Suporte - quantifica a incidência de um *itemset* X ou de uma regra no conjunto de dados, ou seja, indica a 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:

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

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:

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

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

Em outras palavras, o suporte representa as freqüências dos padrões e a confiança a força da implicação, ou seja, em pelo menos $c\%$ das vezes que o antecedente ocorrer nas transações, o conseqüente também deve ocorrer (Zhang & Zhang, 2002). Usualmente, os valores de suporte e confiança mínimos são definidos pelo usuário antes da realização da mineração das regras de associação. Um problema ao se definir esses valores é que se

eles forem altos são gerados, em geral, regras triviais e, se forem baixos, em geral, são gerados um grande volume de regras, dificultando a análise por parte do usuário. Uma forma de superar essas dificuldades é generalizar as regras, ou seja, tornar mais gerais os conceitos específicos, expressando um conhecimento mais amplo da realidade e facilitando a sua compreensão, sem perder as regras específicas. Assim, para que a generalização das regras ocorra, é necessário algum conhecimento sobre o domínio da aplicação, podendo ser expresso, por exemplo, via taxonomias.

As taxonomias refletem uma caracterização coletiva ou individual de como os itens podem ser hierarquicamente classificados (Adamo, 2001). Eventualmente, múltiplas taxonomias podem estar presentes simultaneamente, refletindo a existência de diversos pontos de vista ou a possibilidade de classificações distintas para o mesmo conjunto de itens. Na Figura 1 é apresentado um pequeno exemplo de uma taxonomia. Nesse exemplo pode-se verificar que: camiseta é uma roupa leve, bermuda é uma roupa leve, roupa leve é um tipo de roupa, sandália é um tipo de calçado, etc.

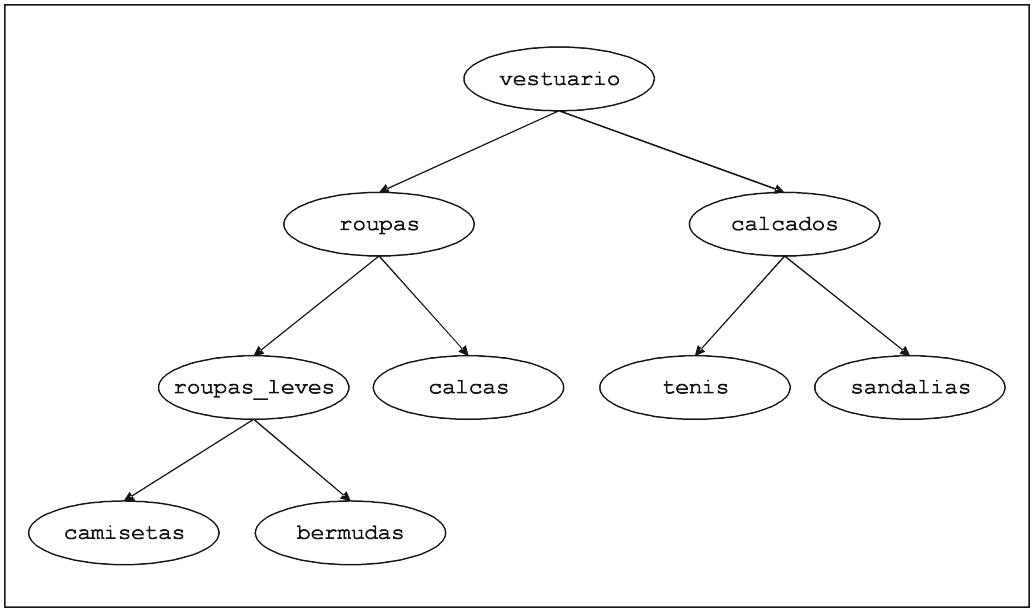


Figura 1: Exemplo de uma taxonomia para vestuário

Uma regra de associação usando taxonomias pode ser definida como (Srikant & Agrawal, 1995; Huang & Wu, 2002; Yang, 2005):

> 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 (chamado

itemset) tal que $t_i \subseteq A$. Seja $\mathcal{T}$ um grafo direcional acíclico sobre os itens, representando um conjunto de taxonomias. Se há uma aresta em $\mathcal{T}$ de um item $\bar{x}$ para um item x, $\bar{x}$ é dito ser pai de x e x é dito ser filho de $\bar{x}$. Se há um caminho de $\hat{x}$ para x no fecho transitivo de $\mathcal{T}$, diz-se que $\hat{x}$ é ancestral de x e x descendente de $\hat{x}$. Um item sem nenhum descendente é chamado de terminal; caso contrário de não terminal. O conjunto de itens A contém tanto os itens terminais como os não terminais. Diz-se que um *itemset* $\bar{X}$ é pai de um *itemset* X (e X filho de $\bar{X}$) se $\bar{X}$ é obtido pela substituição de um único item em X por um de seus itens pais. Diz-se que um *itemset* $\hat{X}$ é ancestral de um *itemset* X (e X descendente de $\hat{X}$) se $\hat{X}$ é obtido pela substituição de um ou mais itens em X pelos seus itens ancestrais. Formalmente, $\hat{X}$ é um *itemset* ancestral de X se para todo $x \in X$, tanto $x \in \hat{X}$ ou existe um ancestral $\hat{x}$ de x tal que $\hat{x} \in \hat{X}$. É dito que uma transação t_i suporta um item x se x está em t_i ou x é um ancestral de algum item em t_i. Uma transação t_i suporta um *itemset* X se t_i suporta todo item em X.

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 $LHS \cup RHS$.

Assim como no caso das regras de associação, as regras de associação generalizadas também utilizam as medidas de suporte e confiança durante o processo de geração de regras. Nesse caso, embora o suporte de um item terminal x na taxonomia seja definido de maneira semelhante, o suporte para um item y não terminal na taxonomia é definido em Adamo (2001) como: $sup(y) = sup(\cup\ des(y))$, em que $des(y)$ representa o conjunto de descendentes de y.

3. Abordagem para Generalização de Regras de Associação

Nesta seção é descrita uma abordagem de pós-processamento de regras de associação (*APRA*) via a utilização de taxonomias de domínio (conhecimento de domínio) (Carvalho, Rezende, & Castro, 2006, 2007). O objetivo dessa abordagem é pós-processar um conjunto de regras de associação (RA) de forma a obter um conjunto de regras mais reduzido e

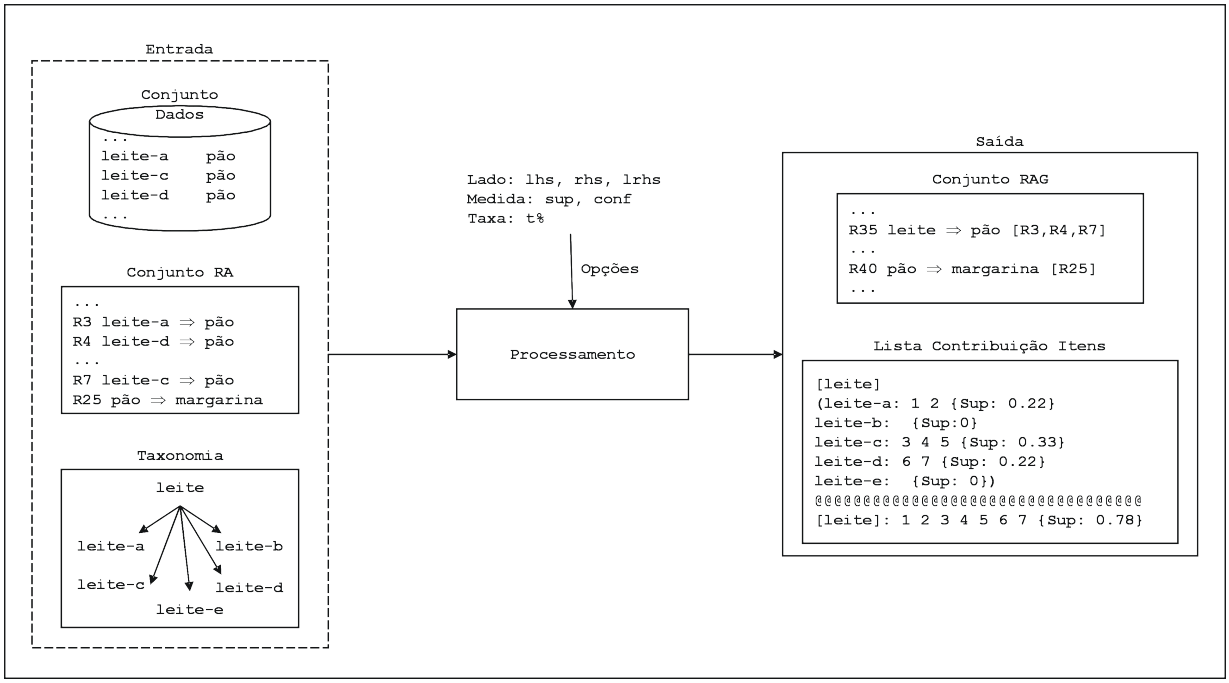


Figura 2: Visão geral da abordagem de pós-processamento de regras de associação (APRA)

expressivo, a fim de facilitar a compreensão do mesmo pelo usuário.

A *APRA* é apresentada na Figura 2. Considera-se que os elementos contidos no pontilhado estão disponíveis, a saber: um conjunto de regras de associação formado somente por regras específicas (regras compostas somente por itens contidos na base de dados), um conjunto de dados utilizado para gerar as regras específicas e as taxonomias. Com base nessas entradas a abordagem obtém um conjunto de regras de associação generalizadas (RAG) composto por regras específicas que não puderam ser generalizadas (por exemplo, regra R40 da Figura 2) e por regras generalizadas obtidas pelo agrupamento de algumas regras específicas via a utilização das taxonomias fornecidas (por exemplo, regra R35 da Figura 2 – regra obtida pelo agrupamento das regras *leite-a* $\Rightarrow$ *pão* (R3), *leite-d* $\Rightarrow$ *pão* (R4) e *leite-c* $\Rightarrow$ *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 uma taxonomia fornecida 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 2). Enquanto a generalização *lhs* indica a relação entre uma categoria/classe de itens e itens específicos, a *rhs* indica a relação entre os itens específicos e uma categoria/classe de itens. Já a generalização *lrhs* indica a relação entre categorias/classes de

itens.

Na *APRA*, regras generalizadas podem ser obtidas sem a utilização de todos os itens contidos na taxonomia, ou seja, a abordagem transforma regras específicas em regras gerais mesmo que a regra generalizada correspondente a um subconjunto de regras específicas tenha sido gerada sem todos os itens específicos do item geral contido na taxonomia. Por exemplo: suponha que a regra *leite* $\Rightarrow$ *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* $\Rightarrow$ *pão* irá existir mesmo que não exista uma regra 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 nos itens gerais. Por exemplo, para a o conjunto generalizado acima (conjunto RAG), a listagem apresentada na Figura 2 é gerada.

Essa é uma das vantagens da *APRA*: poder aproveitar taxonomias que contenham conhecimento de um mesmo domínio. Considere uma taxonomia que contenha conhecimento sobre produtos alimentícios. Qualquer base de dados que contenha informações a respeito desses produtos poderá utilizar a mesma taxonomia no processo de generalização, uma vez que para cada regra generalizada identifica-se em uma listagem o suporte de cada um dos itens específicos. Isso significa que se um determinado item possui 0% de suporte (caso dos itens leite-b e leite-e da Figura 2) ele não estava presente nas transações e, portanto, não contribuiu para o processo de generalização.

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 2) for t% maior do que o maior valor da mesma medida selecionada nas regras específicas (opção *Taxa* da Figura 2).

A fim de viabilizar a *APRA* foi desenvolvido o algoritmo $APRA_{alg}$, apresentado no Algoritmo 1.

O $APRA_{alg}$ supõe a existência de um conjunto de regras de associação na sintaxe padrão (.apr.dcar) (item R do Algoritmo 1) obtido, a priori, por métodos tradicionais de extração, de um conjunto de dados (.apr.data) (item D do Algoritmo 1) utilizado para extrair o conjunto de regras e de um conjunto de taxonomias (.tax) (item T do Algoritmo 1), conforme mostra a Figura 3. A partir da especificação desses conjuntos, o $APRA_{alg}$ obtém um conjunto de regras de associação generalizadas (rules_gen.txt) (item *RGen* do Algoritmo 1) e uma listagem de contribuição de itens (taxonomy_elements.txt) (item *Contrib* do Algoritmo 1). Uma descrição de cada um dos arquivos é apresentada a seguir.

Algoritmo 1 *$APRA_{alg}$ – algoritmo referente a viabilização da APRA.*

Entrada: Base de dados D, conjunto R de regras de associação na sintaxe padrão, conjunto de taxonomias T, lado L da regra a ser generalizado (lhs, rhs, $lrhs$), medida M a ser utilizada na generalização (sup, $conf$), taxa t da medida M.
Saída: Conjunto $RGen$ de regras de associação generalizadas e listagem $Contrib$ contendo a participação de cada um dos itens específicos nos itens gerais.

1: $Contrib$:= calcula-contribuicao-itens(D,T);
2: $RGen := R$;
3: **se** (($L = lhs$) OR ($L = rhs$)) **então**
4: $SC1$:= gera-subconjuntos-iniciais(R,$\bar{L}$);
5: **para todo** ($\widehat{SC1} \geqslant 2$, $\widehat{SC1} \subseteq SC1$) **faça**
6: $NATax := 1$;
7: **enquanto** ($NATax \leqslant NMTax$) **faça**
8: substitui-itens($\widehat{SC1}$,L,$NATax$);
9: elimina-itens-repetidos($\widehat{SC1}$,L);
10: ordena-lexicograficamente($\widehat{SC1}$,L);
11: $SC2$:= gera-subconjuntos($\widehat{SC1}$,L);
12: **para todo** ($\widehat{SC2} \geqslant 2$, $\widehat{SC2} \subseteq SC2$) **faça**
13: r := regra($\widehat{SC2}$);
14: regra-valida := avalia-criterios-generalizacao(r);
15: **se** regra-valida **então**
16: calcula-tabela-contingencia(r,D);
17: regra-valida := verifica-criterio-medida(r,M,t);
18: **se** regra-valida **então**
19: $RGen := RGen \cup \{r\}$;
20: $RGen$:= remove-regras-origem(r,$RGen$);
21: **fim-se**
22: **fim-se**
23: **fim-para**
24: $NATax := NATax + 1$;
25: **fim-enquanto**
26: **fim-para**
27: **fim-se**
28: **se** ($L = lrhs$) **então**
29: $TempRules := R$;
30: $NATax := 1$;
31: **enquanto** ($NATax \leqslant NMTax$) **faça**
32: substitui-itens($TempRules$,L,$NATax$);
33: elimina-itens-repetidos($TempRules$,L);
34: ordena-lexicograficamente($TempRules$,L);
35: $SC1$:= gera-subconjuntos($TempRules$,L);
36: **para todo** ($\widehat{SC1} \geqslant 2$, $\widehat{SC1} \subseteq SC1$) **faça**
37: r := regra($\widehat{SC1}$);
38: regra-valida := avalia-criterios-generalizacao(r);
39: **se** regra-valida **então**
40: calcula-tabela-contingencia(r,D);
41: regra-valida := verifica-criterio-medida(r,M,t);
42: **se** regra-valida **então**
43: $RGen := RGen \cup \{r\}$;
44: $RGen$:= remove-regras-origem(r,$RGen$);
45: **fim-se**
46: **fim-se**
47: **fim-para**
48: $NATax := NATax + 1$;
49: **fim-enquanto**
50: **fim-se**
51: $RGen$:= remove-regras-repetidas($RGen$);
52: $RGen$:= sintaxe-padrao($RGen$);

O arquivo ***.apr.data*** é composto por um conjunto de transações, onde cada linha representa uma transação. Na Figura 4, a primeira linha representa uma transação de compra, na qual os seguintes produtos foram adquiridos: leite_batavo, nescau, pao e margarina.

O arquivo ***.apr.dcar*** é composto por um conjunto de regras de associação expresso na

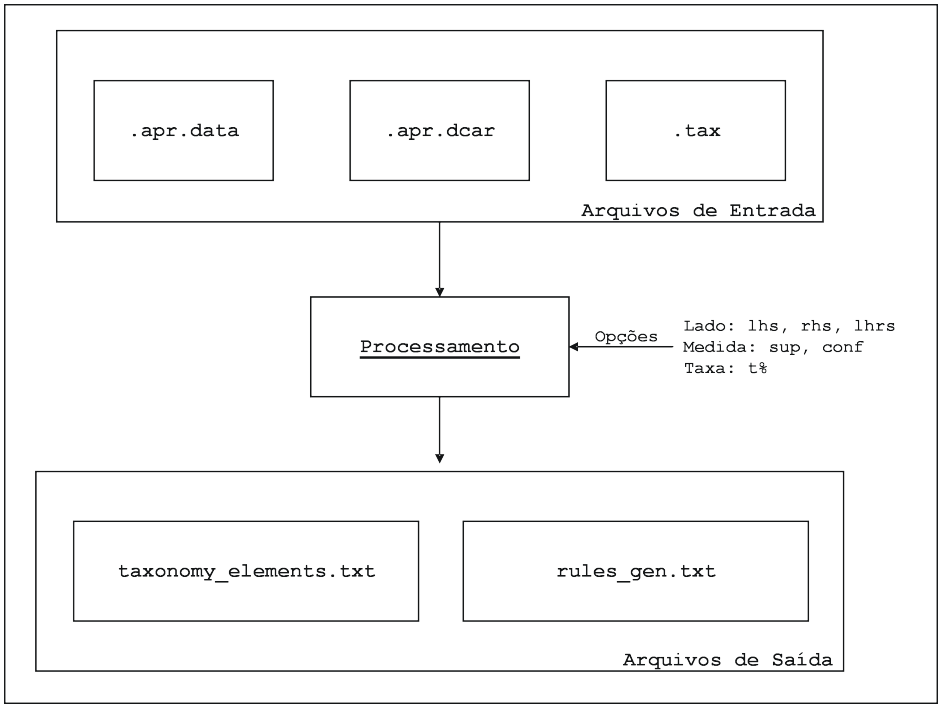


Figura 3: Relacionamento entre os arquivos de entrada e saída do $APRA_{alg}$

```
leite_batavo    nescau  pao     margarina
leite_batavo    nescau  pao     margarina
leite_nilza     nescau  pao     margarina
leite_nilza     nescau  pao     margarina
leite_nilza     nescau  pao     margarina
leite_parmalat  nescau  pao     margarina
leite_parmalat  acucar  cafe
macarrao        molho_tomate    cebola
molho_tomate    lazanha coca_cola
```

Figura 4: Exemplo de um arquivo de dados (`.apr.data`)

sintaxe padrão de regras de associação (Melanda & Rezende, 2003), o qual encontra-se exemplificado na Figura 5. É posível observar que a sintaxe consiste na apresentação do número da regra enter colchetes, seguidos pelo antecedente, pelo conseqüente e pelos valores da tabela de contigência entre colchetes, sendo que os itens são separados por vírgulas.

O arquivo ***.tax*** é composto por um conjunto de taxonomias. O conjunto de taxonomias da Figura 6 é formado por uma única taxonomia de dois níveis. No formato utilizado, os items mais específicos das taxonomias aparecem primeiro na especificação do arquivo. Além disso, para cada nível de abstração identifica-se o nível ao qual o item pertence. Por exemplo, o item leite da Figura 6 representa a primeira abstração (identificador (1)) relacionada aos tipos de leite existentes e o item produtos_matinais a segunda abstração

```
[R0001],TRUE,leite_nilza,[0.333333,0.666667,0.000000,0.000000,9]
[R0002],TRUE,nescau,[0.666667,0.333333,0.000000,0.000000,9]
[R0003],TRUE,pao,[0.666667,0.333333,0.000000,0.000000,9]
[R0004],TRUE,margarina,[0.666667,0.333333,0.000000,0.000000,9]
[R0005],leite_nilza,nescau,[0.333333,0.000000,0.333333,0.333333,9]
[R0006],nescau,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9]
[R0007],leite_nilza,pao,[0.333333,0.000000,0.333333,0.333333,9]
[R0008],pao,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9]
[R0009],leite_nilza,margarina,[0.333333,0.000000,0.333333,0.333333,9]
[R0010],margarina,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9]
[R0011],nescau,pao,[0.666667,0.000000,0.333333,0.000000,9]
[R0012],pao,nescau,[0.666667,0.000000,0.333333,0.000000,9]
[R0013],nescau,margarina,[0.666667,0.000000,0.333333,0.000000,9]
[R0014],margarina,nescau,[0.666667,0.000000,0.333333,0.000000,9]
[R0015],pao,margarina,[0.666667,0.000000,0.333333,0.000000,9]
[R0016],margarina,pao,[0.666667,0.000000,0.333333,0.000000,9]
[R0017],leite_nilza & nescau,pao,[0.333333,0.000000,0.333333,0.333333,9]
[R0018],leite_nilza & pao,nescau,[0.333333,0.000000,0.333333,0.333333,9]
[R0019],nescau & pao,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9]
[R0020],leite_nilza & nescau,margarina,[0.333333,0.000000,0.333333,0.333333,9]
[R0021],leite_nilza & margarina,nescau,[0.333333,0.000000,0.333333,0.333333,9]
[R0022],nescau & margarina,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9]
[R0023],leite_nilza & pao,margarina,[0.333333,0.000000,0.333333,0.333333,9]
[R0024],leite_nilza & margarina,pao,[0.333333,0.000000,0.333333,0.333333,9]
[R0025],pao & margarina,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9]
[R0026],nescau & pao,margarina,[0.666667,0.000000,0.333333,0.000000,9]
[R0027],nescau & margarina,pao,[0.666667,0.000000,0.333333,0.000000,9]
[R0028],pao & margarina,nescau,[0.666667,0.000000,0.333333,0.000000,9]
[R0029],leite_nilza & nescau & pao,margarina,[0.333333,0.000000,0.333333,0.333333,9]
[R0030],leite_nilza & nescau & margarina,pao,[0.333333,0.000000,0.333333,0.333333,9]
[R0031],leite_nilza & pao & margarina,nescau,[0.333333,0.000000,0.333333,0.333333,9]
[R0032],nescau & pao & margarina,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9]
```

Figura 5: Exemplo de um arquivo de regras de associação na sintaxe padrão (`.apr.dcar`)

(identificador (2)). Assim, para cada item especificado no arquivo deve-se identificar a qual nível de abstração o mesmo pertence. Como se pode observar, a taxonomia armazena as seguintes informações: leite_batavo é um tipo de leite, leite_molico é um tipo de leite, leite_nilza é um tipo de leite, etc; nescau é um tipo de achocolatado, tody é um tipo de achocolatado; leite e achocolatado são tipos de produtos_matinais. Caso o especialista queira especificar várias taxonomias simultaneamente no arquivo, basta informar primeiramente as abstrações de nível 1 de todas as taxonomias, depois as de nível 2 e assim sucessivamente, como mostra a Figura 7. Nesse caso, têm-se duas taxonomias: uma relacionada a produtos alimentícios e outra relacionada a produtos de vestuário.

```
leite(leite_batavo,leite_molico,leite_nilza,leite_parmalat,leite_salute)(1).
achocolatado(nescau,tody)(1).
produtos_matinais(leite,achocolatado)(2).
```

Figura 6: Exemplo de um arquivo contendo uma taxonomia (`.tax`)

O arquivo ***taxonomy_ elements.txt***, apresentado na Figura 8, é composto por uma lis-

```
leite(leite_batavo,leite_molico,leite_nilza,leite_parmalat,leite_salute)(1).
achocolatado(nescau,tody)(1).
roupas(camisetas,bermudas,calcas)(1).
calcados(tenis,sandalia,chinelo)(1).
produtos_matinais(leite,achocolatado)(2).
vestuario(roupas,calcados)(2).
```

Figura 7: Exemplo de um arquivo contendo duas taxonomias (`.tax`)

tagem contendo a participação de cada um dos itens específicos nos itens gerais. Observe que para cada um dos itens gerais contidos na taxonomia da Figura 6 é listado todos os itens específicos que o compõem, as transações as quais os mesmos pertencem e a freqüência (suporte) de cada um. Por exemplo, o item leite_batavo, referente aos dados apresentados na Figura 4, está presente nas transações 1 e 2 e possui, portanto, suporte de 22.22% (2/9). No caso dos itens leite_molico e leite_salute, por exemplo, os mesmos não estão presentes em nenhuma das transações e, portanto, possuem suporte de 0.0% (0/9). Após a seqüência de @'s encontram-se as transações as quais cada um dos itens gerais pertencem, assim como suas respectivas freqüências (suporte).

```
Participacao dos itens especificos nos itens gerais contidos na taxonomia
Copyright (c) Veronica Oliveira de Carvalho
Date: Wed Jan 25 17:06:09 2006
Etapa: PosProc
Parametros: Lado: lhs, Medida: sup, greater, 0%


[leite]
(leite_batavo: 1 2 {Sup: 0.222222222222222}
leite_molico:  {Sup: 0}
leite_nilza: 3 4 5 {Sup: 0.333333333333333}
leite_parmalat: 6 7 {Sup: 0.222222222222222}
leite_salute:  {Sup: 0})

[achocolatado]
(nescau: 1 2 3 4 5 6 {Sup: 0.666666666666667}
tody:  {Sup: 0})

[produtos_matinais]
(leite: 1 2 3 4 5 6 7 {Sup: 0.777777777777778}
achocolatado: 1 2 3 4 5 6 {Sup: 0.666666666666667})


@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@


[achocolatado]: 1 2 3 4 5 6 {Sup: 0.666666666666667}

[leite]: 1 2 3 4 5 6 7 {Sup: 0.777777777777778}

[produtos_matinais]: 1 2 3 4 5 6 7 {Sup: 0.777777777777778}
```

Figura 8: Exemplo de um arquivo contendo a listagem de contribuição de itens (`taxonomy_elements.txt`)

O arquivo ***rules_ gen.txt*** é composto por um conjunto de regras de associação generalizadas expresso na sintaxe padrão de regras de associação, o qual encontra-se exemplificado na Figura 9. Entretanto, como se pode observar, este conjunto possui uma diferença em relação ao conjunto da Figura 5: após a matriz de contingência da regra generalizada encontram-se os identificadores das regras que deram origem a respectiva regra. Por exemplo, a regra [R0022] da Figura 9 originou-se das regras específicas [R0023], [R0026] e [R0029] da Figura 5.

```
Regras de associacao generalizadas
Copyright (c) Veronica Oliveira de Carvalho
Date: Wed Jan 25 17:06:09 2006
Etapa: PosProc
Parametros: Lado: lhs, Medida: sup, greater, 0%


[R0001],TRUE,leite_nilza,[0.333333,0.666667,0.000000,0.000000,9],[R0001]
[R0002],TRUE,nescau,[0.666667,0.333333,0.000000,0.000000,9],[R0002]
[R0003],TRUE,pao,[0.666667,0.333333,0.000000,0.000000,9],[R0003]
[R0004],TRUE,margarina,[0.666667,0.333333,0.000000,0.000000,9],[R0004]
[R0005],leite_nilza,nescau,[0.333333,0.000000,0.333333,0.333333,9],[R0005]
[R0006],nescau,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9],[R0006]
[R0007],pao,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9],[R0008]
[R0008],margarina,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9],[R0010]
[R0009],pao,nescau,[0.666667,0.000000,0.333333,0.000000,9],[R0012]
[R0010],margarina,nescau,[0.666667,0.000000,0.333333,0.000000,9],[R0014]
[R0011],pao,margarina,[0.666667,0.000000,0.333333,0.000000,9],[R0015]
[R0012],margarina,pao,[0.666667,0.000000,0.333333,0.000000,9],[R0016]
[R0013],leite_nilza & pao,nescau,[0.333333,0.000000,0.333333,0.333333,9],[R0018]
[R0014],nescau & pao,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9],[R0019]
[R0015],leite_nilza & margarina,nescau,[0.333333,0.000000,0.333333,0.333333,9],[R0021]
[R0016],margarina & nescau,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9],[R0022]
[R0017],margarina & pao,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9],[R0025]
[R0018],margarina & pao,nescau,[0.666667,0.000000,0.333333,0.000000,9],[R0028]
[R0019],leite_nilza & margarina & pao,nescau,[0.333333,0.000000,0.333333,0.333333,9],[R0031]
[R0020],margarina & nescau & pao,leite_nilza,[0.333333,0.333333,0.333333,0.000000,9],[R0032]
[R0021],produtos_matinais,margarina,[0.666667,0.111111,0.222222,0.000000,9],[R0013,R0009,R0020]
[R0022],pao & produtos_matinais,margarina,[0.666667,0.000000,0.333333,0.000000,9],[R0023,R0026,R0029]
[R0023],margarina & produtos_matinais,pao,[0.666667,0.000000,0.333333,0.000000,9],[R0030,R0024,R0027]
[R0024],produtos_matinais,pao,[0.666667,0.111111,0.222222,0.000000,9],[R0011,R0007,R0017]
```

Figura 9: Exemplo de um arquivo de regras de associação generalizadas (`rules_gen.txt`)

4. Módulo de Exploração de Regras de Associação Generalizadas RulEE-RAG

O projeto do módulo RulEE-RAG foi desenvolvido considerando as necessidades da abordagem proposta por Carvalho, Rezende, & Castro (2006, 2007). A parte da abordagem para a qual o módulo foi desenvolvido se refere à exploração interativa de regras de associação generalizadas. O módulo tem como entrada: o identificador do conjunto de regras de origem; o arquivos de dados; o arquivo de taxonomias; o arquivo de contribuição dos itens específicos nos itens gerais; e o arquivo de regras generalizadas representadas na sintaxe padrão definida em Melanda & Rezende (2003).

Além dos arquivos de entrada do módulo, foram gerados os artefatos de diagrama de casos de uso e um esquema para o módulo RulEE-RAG. Assim, na Figura 10 é mostrado o diagrama de caso de uso para o módulo RulEE-RAG e na Figura 11 são apresentadas as classes com que a classe *RAG* se relaciona. A classe *RAG* representa uma exploração de um conjunto de regras de associação generalizadas de um determinado projeto. A classe Ruleset encapsula um conjunto de regras e a classe Rule representa uma regra. A classe Session representa uma sessão, informando, por exemplo, o usuário responsável. Já a classe Project representa um projeto existente.

Após finalizada a etapa de projeto do módulo RulEE-RAG, iniciou-se a etapa de implementação. Na próxima seção pode ser observada a utilização do módulo RulEE-RAG, e nas seções seguintes são detalhadas as tabelas e as classes criadas para sua implementação.

4.1 Utilização do RulEE-RAG

O módulo de regras de associação generalizadas (RulEE-RAG) foi implementado utilizando a estrutura proporcionada pelo ambiente RulEE[‡] (Paula, 2003). Assim, após a autenticação do usuário no RulEE, o módulo pode ser acessado por meio de um *link* no menu lateral, visualizado na Figura 12. O módulo RulEE-RAG permite ao usuário escolher entre inserir um novo conjunto de regras de associação generalizadas e explorar um conjunto existente (Figura 13).

Selecionando a opção de inserir, o usuário deve fornecer o identificador do conjunto de regras específicas (referente ao arquivo .apr.dcar da Figura 3, página 9) utilizado no processo de pós-processamento (considera-se que o mesmo encontra-se disponível no ambiente) e realizar o carregamento dos arquivos de dados (arquivo .apr.data da Figura 3)

[‡]http://143.107.231.137/rulee/index.html.

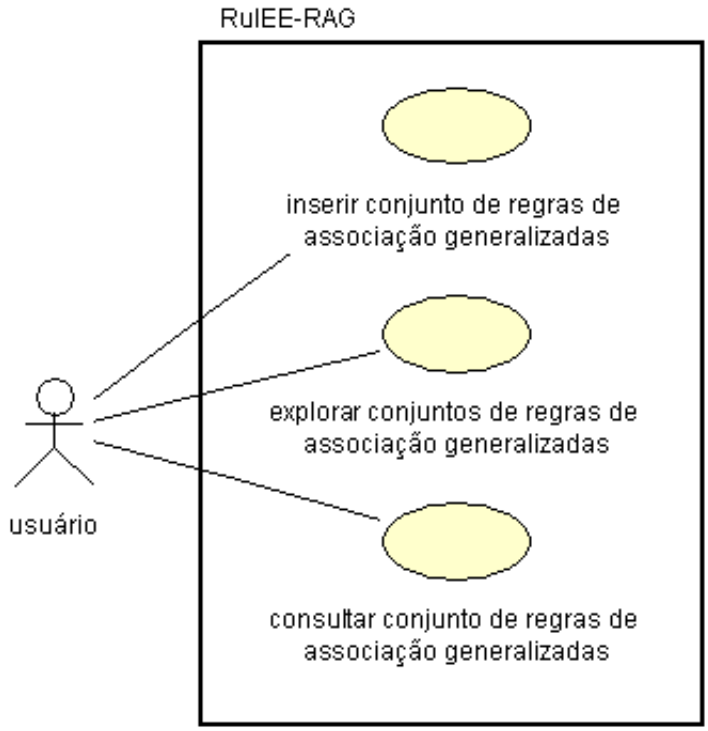


Figura 10: Diagrama de casos de uso do módulo RulEE-RAG

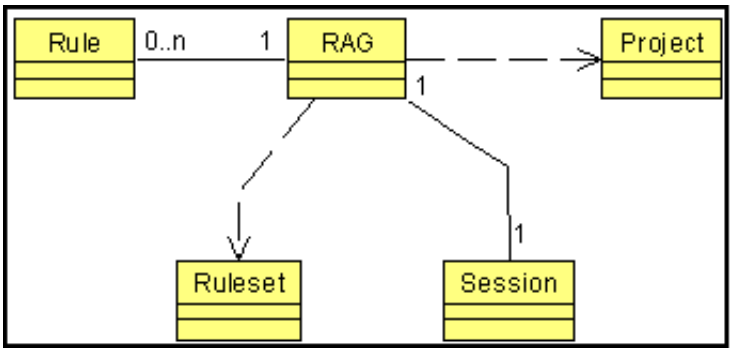


Figura 11: Esquema do módulo RulEE-RAG

e taxonomias (arquivo .tax da Figura 3), além dos arquivos de regras obtidas (arquivo rules_gen.txt da Figura 3) e a contribuição dos itens específicos nos itens gerais (arquivo taxonomy_elements.txt da Figura 3), gerados pelo $APRA_{alg}$ (Figura 14). Finalizada a inserção, o usuário pode escolher entre prosseguir para a etapa de exploração do conjunto de regras ou sair do módulo.

Para que o usuário possa explorar um conjunto de regras, ele deve selecionar o projeto e o identificador do conjunto desejado. Para cada conjunto selecionado, o módulo exibe a data do pós-processamento e os parâmetros utilizados durante o processamento, forne-

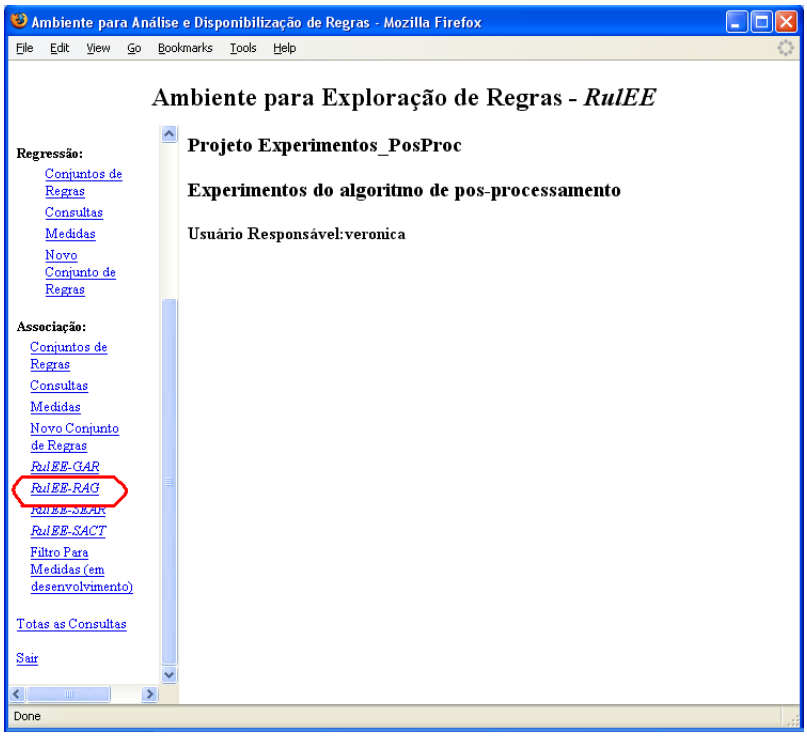


Figura 12: *Link* de acesso para o módulo RulEE-RAG

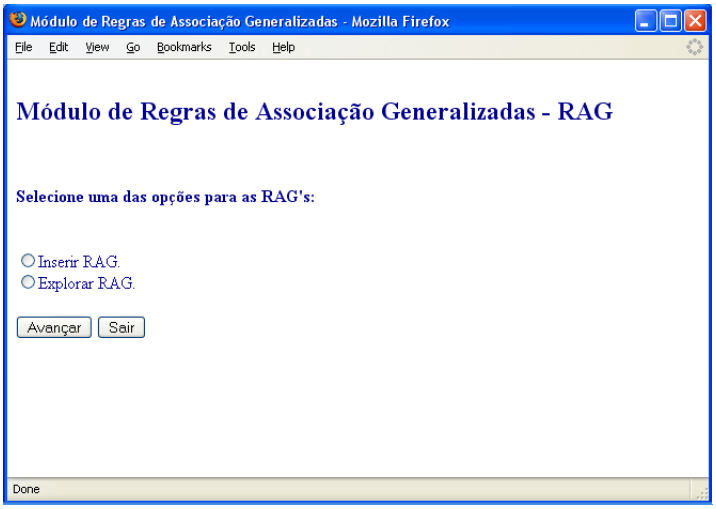


Figura 13: Interface inicial do RulEE-RAG

cendo a opção de realizar consultas ou visualizar todas as regras do conjunto em questão (Figura 15).

Ao selecionar a opção de consulta, o módulo solicita ao usuário os atributos, as restrições e a forma de ordenação para o conjunto de regras. Inicialmente, a opção de escolha do usuário em relação aos atributos, as restrições e a forma de ordenação, se restringem

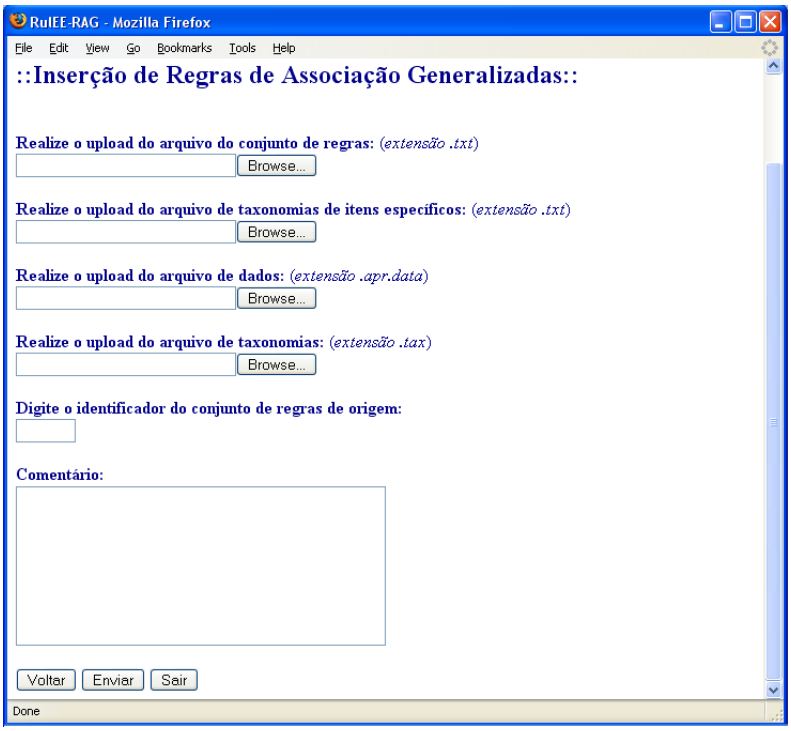


Figura 14: Inserção de um conjunto de regras de associação generalizadas

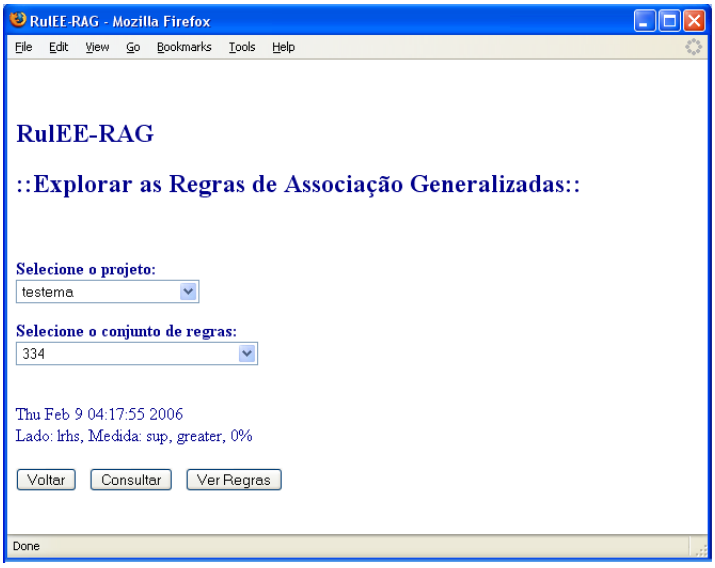


Figura 15: Exploração de um conjunto de regras de associação generalizadas

às medidas de avaliação disponíveis no ambiente RulEE e alguns atributos do sistema (identificador da regra, número da regra, identificador do conjunto de regras, a regra, antecedente da regra, conseqüente da regra e flag que indica se a regra é generalizada ou não) (Figura 16). Assim, o módulo permite ao usuário realizar diversos tipos de consultas,

como, selecionar apenas regras generalizadas ou específicas.

A partir das informações obtidas do usuário é montada uma consulta SQL, a qual pode ser modificada pelo usuário no modo avançado, o que permite ao usuário realizar consultas mais complexas do que as inicialmente possíveis. Ao salvar a consulta, o resultado é exibido, como ilustrado na Figura 17. No item (a) é possível observar algumas informações disponibilizadas em relação à consulta, além do resultado obtido. Já no item (b), as opções fornecidas pelo módulo, que são: atualizar os dados; editar a consulta; criar uma cópia, ou seja, salvar a consulta como sendo outra, possibilitando a realização de novas consultas sem que as antigas sejam sobrescritas; criar um comentário e explorar detalhadamente ou de forma geral as regras resultantes, no caso do usuário ter consultado apenas as regras generalizadas.

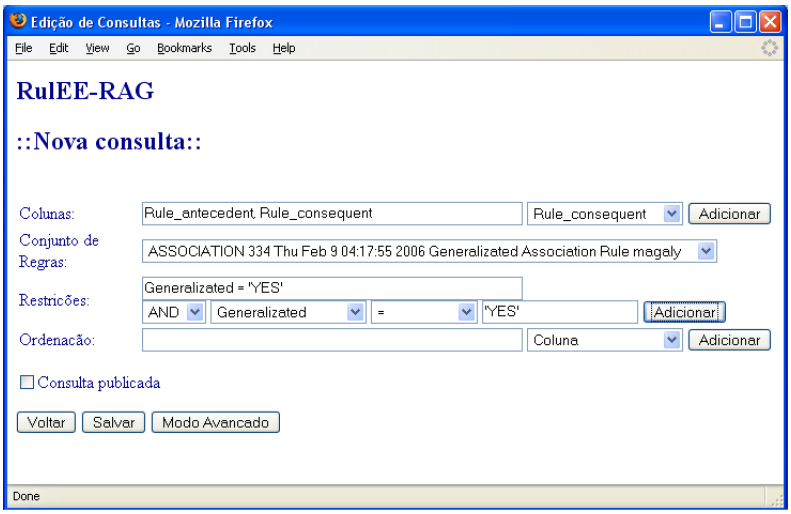
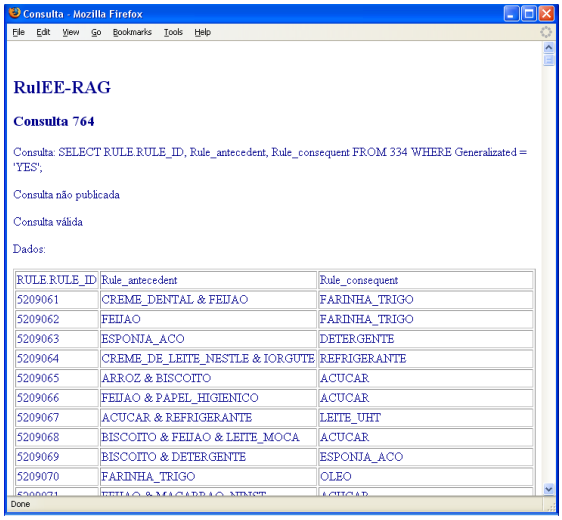


Figura 16: Consulta SQL em um conjunto de regras de associação generalizadas

Caso o usuário escolha a opção visualizar as regras, o módulo as exibe fazendo a distinção entre as regras específicas e as generalizadas (formadas por duas ou mais regras específicas), pois as regras específicas podem apenas ser visualizadas, enquanto que as regras generalizadas podem ser exploradas (Figura 18). No item (a) da figura é possível observar as regras específicas e, no item (b), as regras generalizadas e suas opções de exploração. Essas opções permitem ao usuário explorar de maneira detalhada ou geral, todas as regras generalizadas ou apenas as selecionadas.

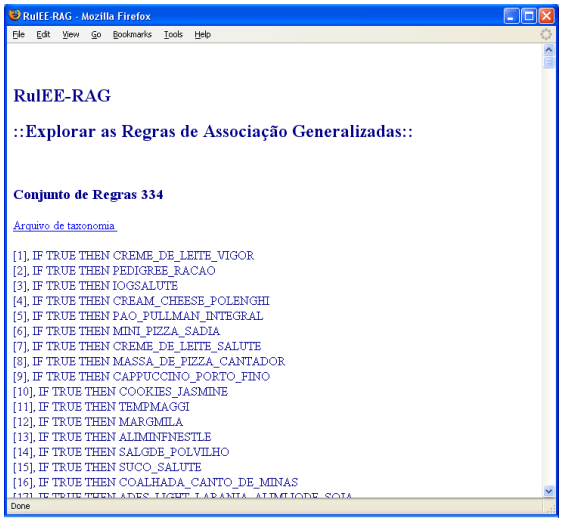
Na exploração detalhada o módulo exibe o número da regra generalizada; a regra generalizada; os valores de algumas medidas; as regras específicas que originaram a regra generalizada, com seus respectivos valores das medidas de suporte e confiança, entre outras, e a contribuição de cada item específico da taxonomia para cada item geral contido



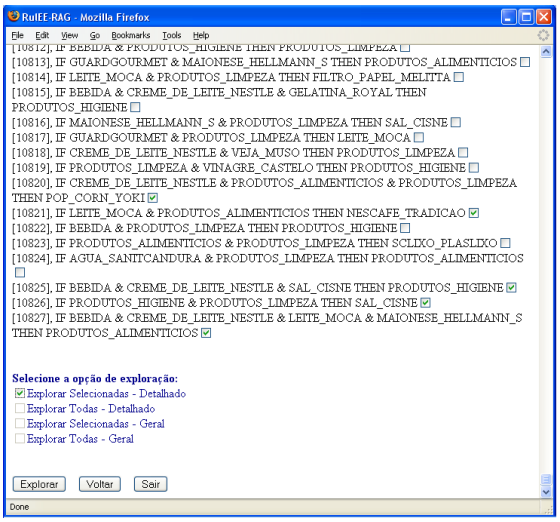
(a) Parte inicial da janela

(b) Parte final da janela

Figura 17: Página que exibe o resultado de uma consulta realizada



(a) Regras Específicas



(b) Regras Generalizadas

Figura 18: Página que exibe as regras do conjunto selecionado

na regra generalizada (Figura 19). Já na exploração geral, a única diferença em relação à exploração detalhada é que não é exibida a contribuição de itens para cada regra generalizada. A contribuição é exibida uma única vez no final de cada página para todas as regras exploradas, exibindo os itens generalizados e os itens específicos de todas as regras generalizadas exploradas.

Apresentada uma visão geral do funcionamento do módulo RuleE-RAG, nas próximas seções são descritas detalhes de implementação, como as tabelas das bases de dados e as classes implementadas.

Figura 19: Exploração detalhada das regras selecionadas)

4.2 Repositório

A base de dados do RulEE foi alterada com a criação de duas novas tabelas, CONTRIB_ITEM_SPEC e SET_TAX. Além disso, foram adicionados os atributos SET_TAX_ID e CONTRIB_ITEM_SPEC_FILE na tabela RULESET. A seguir são descritas as tabelas criadas para o módulo RulEE-RAG.

Tabela: CONTRIB_ITEM_SPEC
Descrição: armazena a contribuição de cada item específico relacionado a um item geral. Tabela utilizada no módulo para regras de associação generalizadas (RAG).

Colunas:

Nome da Coluna	Definição	Descrição
RULESET_ID	INTEGER(11) NOT NULL	Identificador do conjunto de regras.
SET_TAX_ID	VARCHAR(20) NOT NULL	Identificador da taxonomia.
GENERALIZATION_ITEM_ID	VARCHAR(100) NOT NULL	Identificador do item generalizado.
GENERALIZATION_ITEM	VARCHAR(200) NOT NULL	Item específico.
SUP	DECIMAL(11,4) NOT NULL	Suporte do item específico.
ITEM_PAI	ENUM('YES','NO') NULL	Indica se é um item específico ou geral.

Restrições de Integridade:

Nome da Restrição	Definição
PK_ CONTRIB_ITEM_SPEC	PRIMARY KEY (RULESET_ID, SET_TAX_ID, GENERALIZATION_ITEM_ID, GENERALIZATION_ITEM)
CONTRIBITEMSPEC_RULESETID_INDEX	INDEX CONTRIBITEMSPEC_RULESETID_INDEX (RULESET_ID)
CONTRIBITEMSPEC_SETTAX_INDEX	INDEX CONTRIBITEMSPEC_SETTAX_INDEX (SET_TAX_ID, GENERALIZATION_ITEM_ID)
FK_CONTRIBITEMSPEC_RULESETID	FOREING KEY (RULESET_ID) REFERENCES RULESET
FK_CONTRIBITEMSPEC_SETTAX	FOREING KEY (GENERALIZATION_ITEM_ID, SET_TAX_ID) REFERENCES SET_TAX

Tabela: SET_TAX

Descrição: armazena taxonomias utilizadas para generalizar itens no módulo RuleERAG.

Colunas:

Nome da Coluna	Definição	Descrição
SET_TAX_ID	VARCHAR(20) NOT NULL	Identificador da taxonomia.
GENERALIZATION_ITEM_ID	VARCHAR(100) NOT NULL	Identificador do item generalizado.
GENERALIZATION_ITEM	VARCHAR(200) NOT NULL	Item específico.
NIVEL_GENERALIZATION_ITEM	INTEGER NOT NULL	Nível do item na taxonomia.

Restrições de Integridade:

Nome da Restrição	Definição
PK_SETTAX	PRIMARY KEY (SET_TAX_ID, GENERALIZATION_ITEM_ID)

4.3 Classes e Métodos

Para viabilizar a implementação do módulo RulEE-RAG no ambiente RulEE foram necessários: alterar a classe *Ruleset*, adicionando dois métodos, e criar a classe *RAG*. Os métodos e suas respectivas classes estão descritas a seguir.

4.4 Classe Ruleset

Esta classe encapsula um conjunto de regras de classificação, regressão e associação. Os métodos implementados para viabilizar a implementação do módulo RulEE-RAG estão descritos a seguir.

Método: NewAssociationGeneralizadedRuleset
Funcionalidade: inserir um conjunto de regras de associação generalizadas na base de dados.
Parâmetros de entrada:

Parâmetro	Tipo	Descrição
$RULESET_ID	INTEGER(11)	Código do conjunto de regras.
$RULESET_ID_ORIGEM	INTEGER(11)	Código do conjunto de regras que originou a regra generalizada.
$SET_TAX_ID	VARCHAR(20)	Código da taxonomia.
$RULESET_FILE	VARCHAR(200)	Caminho para para o arquivo com o conjunto de regras.
$SPEC_FILE	VARCHAR(200)	Caminho para o arquivo com a contribuição de cada item específico associado a um item geral.
$COMMENT_FILE	VARCHAR(200)	Indica o arquivo de comentário do usuário que inseriu o conjunto de regras.
$DATA_FILE	VARCHAR(200)	Caminho para o arquivo com os dados utilizados na extração do conjunto de regras.
$TAX_FILE	VARCHAR(200)	Caminho para o arquivo com o conjunto de taxonomias.
$PROJECT_CODE	VARCHAR(20)	Código do projeto.
$SESSION_ID	INTEGER(11)	Código da sessão que está sendo utilizada.

Tipo de retorno: ponteiro para a classe.

4.5 Classe RAG

Esta classe foi projetada e implementada com o intuito de armazenar todos os métodos que auxiliam na exploração de regras de associação generalizadas do módulo RuleE-RAG.

Método: Assign
Funcionalidade: conecta na classe.
Parâmetros de entrada:

Parâmetro	Tipo	Descrição
$SESSION_ID	INTEGER(11)	Código da sessão que está sendo utilizada.

Tipo de retorno: ponteiro para a classe.

Método: SaveSetTax
Funcionalidade: salva os dados da taxonomia.
Parâmetros de entrada:

Parâmetro	Tipo	Descrição
$SET_TAX_ID	VARCHAR(20)	Código da taxonomia.
$SESSION_ID	INTEGER(11)	Código da sessão que está sendo utilizada.
$RULESET_ID	INTEGER(11)	Código do conjunto de regras.

Tipo de retorno: retorna 0 se ocorreu erro e 1 se não ocorreu erro.

Método: SaveContribItemEspec
Funcionalidade: salva os dados da contribuição dos itens específicos.
Parâmetros de entrada:

Parâmetro	Tipo	Descrição
$RULESET_ID	INTEGER(11)	Código do conjunto de regras.
$SET_TAX_ID	VARCHAR(20)	Código da taxonomia.

Tipo de retorno: retorna 0 se ocorreu erro e 1 se não ocorreu erro.

Método: GetRules
Funcionalidade: encontra todas as regras de um determinado conjunto de regras generalizadas.

Parâmetros de entrada:

Parâmetro	Tipo	Descrição
$RULESET_ID	INTEGER(11)	Código do conjunto de regras.

Tipo de retorno: *array* com as regras encontradas (RULE_ID (identificador da regra), RULE_NUMBER (número da regra), ANTECEDENT (antecedente da regra), CONSEQUENT (conseqüente da regra) e GENERALIZATED (atributo booleano que indica se regra foi generalizada ou não)).

5. Considerações Finais

O objetivo deste relatório técnico foi apresentar uma descrição detalhada do módulo de exploração de regras de associação generalizadas do ambiente RulEE, denominado RulEE-RAG. Esse módulo foi desenvolvido para auxiliar a abordagem para pós-processamento regras de associação desenvolvida por Carvalho, Rezende, & Castro (2006, 2007).

Assim, neste relatório foram descritas as regras de associação, as regras de associação generalizadas, a abordagem proposta por Carvalho, Rezende, & Castro (2006, 2007) e o módulo RulEE-RAG. Além disso, foram detalhadas as alterações realizadas na base de dados e as classes implementadas.

Referências

Adamo, J.-M. (2001). *Data Mining for Association Rules and Sequential Patterns*. Springer-Verlag.

Agrawal, R. & R. Srikant (1994). Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, & C. Zaniolo (Eds.), *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB'94*, pp. 487–499. Disponível em: http://citeseer.ist.psu.edu/agrawal94fast.html [26/02/2007].

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

Carvalho, V. O., S. O. Rezende, & M. Castro (2007). Obtaining and evaluating generalized association rules. In *Proceedings of the 9th International Conference on Enterprise Information Systems*. In press.

Hilderman, R. & H. J. Hamilton (2000). Applying objective interestingness measures in data mining systems. In D. A. Zighed, J. Komorowski, & J. Zytkow (Eds.), *Proceedings of the 4th European Symposium on Principles of Data Mining and Knowledge Discovery, PKDD 2000*, Volume 1910 of *Lecture Notes in Artificial Intelligence*, pp. 432–439. Springer-Verlag.

Huang, Y.-F. & C.-M. Wu (2002). Mining generalized association rules using pruning techniques. In *Proceedings of the 2002 IEEE International Conference on Data Mining, ICDM 2002*, Washington, DC, USA, pp. 227–234. IEEE Computer Society.

Ma, Y., C. K. Wong, & B. Liu (2000). Effective browsing of the discovered association rules using the web. In *ACM SIGKDD-2000 Workshop on Post-Processing in Machine Learning and Data Mining*. Disponível em: http://www.cs.uic.edu/∼liub/publications/papers_topics.html [02/03/2007].

Melanda, E. A. & S. O. Rezende (2003). Sintaxe padrão para representar regras de associação. Relatório Técnico 206, Instituto de Ciências Matemáticas e de Computação – USP – São Carlos. Disponível em: ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_206.pdf [26/02/2007].

Padmanabhan, B. & A. Tuzhilin (2000). Small is beautiful: Discovering the minimal set of unexpected patterns. In *Proceedings of the International Conference*

in Knowledge Discover and Data Mining (KDD-00), pp. 54–63. Disponível em: http://citeseer.ist.psu.edu/tuzhilin00small.html [06/02/2007].

Paula, M. F. (2003). Ambiente para exploração de regras. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação – USP – São Carlos.

Rezende, S. O., J. B. Pugliesi, E. A. Melanda, & M. F. Paula (2003). Mineração de dados. In S. O. Rezende (Ed.), *Sistemas Inteligentes: Fundamentos e Aplicações* (1 ed.)., Capítulo 12, pp. 307–335. Manole.

Srikant, R. & R. Agrawal (1995). Mining generalized association rules. In *Proceedings of the 21th International Conference on Very Large Data Bases, VLDB 1995*, pp. 407–419.

Weiss, S. M. & N. Indurkhya (1998). *Predictive Data Mining: A Pratical Guide*. Morgan Kaufmann Publishers Inc.

Yang, L. (2005). Pruning and visualizing generalized association rules in parallel coordinates. *IEEE Transactions on Knowledge and Data Engineering 17*(1), pp. 60–70.

Zhang, C. & S. Zhang (2002). *Association Rule Mining: Models and Algorithms*, Volume 2307 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag.