UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2569

Sintaxe Padrão para Representar Regras de Associação

Edson Augusto Melanda

Solange Oliveira Rezende

Nº 206

RELATÓRIOS TÉCNICOS

São Carlos - SP
Julho/2003

Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

Sintaxe Padrão para Representar Regras de Associação

Edson Augusto Melanda
Solange Oliveira Rezende

Nº 206

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Julho/2003

Sintaxe Padrão para Representar Regras de Associação

Edson Augusto Melanda
Solange Oliveira Rezende

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística
Laboratório de Inteligência Computacional
Caixa Postal 668, 13560-970 - São Carlos, SP, Brasil
e-mail: {melanda, solange}@icmc.usp.br

Resumo

O processo de Mineração de Dados inicia-se com a escolha das fontes de dados a serem utilizadas e a definição dos objetivos. Uma parcela desses dados é selecionada, pré-processada e submetida aos métodos e ferramentas adequados com o objetivo de encontrar padrões e/ou modelos que representem o conhecimento obtido. Depois de extraídos, os padrões são pós-processados e o conhecimento adquirido é avaliado quanto à sua qualidade e utilidade para determinar a viabilidade de sua utilização no apoio a algum processo de tomada de decisão.

Neste relatório técnico é apresentada a sintaxe padrão utilizada para representar as regras geradas por diferentes algoritmos de associação em um formato único. Além disso, nesse relatório técnico também é apresentada uma classe em Perl que converte as Regras de Associação geradas pelos aplicativos *Apriori*, *Magnum Opus*, *WEKA* e MineSet^{TM}para o formato padrão de Regras de Associação.

Palavras-Chave: Mineração de Dados, Pós-Processamento, Regras de Associação, Sintaxe Padrão.

Julho 2003

Sumário

1 Introdução **1**

2 Conceitos Gerais **2**
2.1 Regras de Associação 2
2.2 Matriz de Contingência 3

3 Formato Padrão das Regras **4**
3.1 Gramática da Sintaxe Padrão 5
3.1.1 Sintaxe Padrão para Regra de Associação 6
3.1.2 Sintaxe Padrão Compacta para Regras de Associação 7
3.2 Cálculo da Matriz de Contingência 8

4 Conversão de Formatos **12**
4.1 *Apriori* 13
4.2 *Magnum Opus* 14
4.3 *WEKA* 16
4.4 MineSet^{TM} 17

5 Considerações Finais **22**

A Apêndice **23**

Referências **35**

Lista de Figuras

1 Produções da Gramática para Regras de Associação 7
2 Regra de associação Gerada pelo Algoritmo do MineSet^{TM} 8
3 Regra de associação da Figura 2 na Sintaxe Padrão 8
4 Produções da Gramática para Regras de Associação no Formato Compacto 9
5 Regra de associação da Figura 2 na Sintaxe Padrão Compacta 10
6 Módulo de Conversão das Saídas dos Algoritmos de Regras de Associação . 12
7 Algumas Regras Geradas pelo *Apriori* Utilizando o Conjunto de Dados *Weather-Nominal* 14
8 Regras da Figura 7 Convertidas para o Formato Padrão Compacto de Regras de Associação 15
9 Algumas Regras Geradas pelo *Magnum Opus* Utilizando o Conjunto de Dados *Weather-Nominal* 15
10 Regras da Figura 9 Convertidas para o Formato Padrão Compacto de Regras de Associação 16
11 Algumas Regras Geradas pelo *WEKA* Utilizando o Conjunto de Dados *Weather-Nominal* 18
12 Regras da Figura 11 Convertidas para o Formato Padrão Compacto de Regras de Associação 19
13 Algumas Regras Geradas pelo MineSet^{TM} Utilizando o Conjunto de Dados *Weather-Nominal* 19
14 Regras da Figura 13 Convertidas para o Formato Padrão Compacto de Regras de Associação. 20
15 Regras da Figura 13 Convertidas para o Formato Padrão de Regras de Associação. 21

Lista de Tabelas

1 Matriz de contingência 4
2 Matriz de contingência com freqüências relaivas 4
3 Base de dados *Weather Nominal* 13

*Este documento foi elaborado com o LaTeX e a bibliografia gerada automaticamente pelo BibTeX.

1 Introdução

Dentre as técnicas de Mineração de Dados, Regras de Associação tem sido uma das que tem despertado maior interesse nos últimos anos (Baesens, Viaene, & Vanthienen 2000). Na área acadêmica diversas pesquisas vêm sendo desenvolvidas, principalmente na etapa de identificação dos padrões e, mais recentemente, na avaliação e interpretação do conhecimento extraido. Esta técnica também tem sido amplamente utilizada por organizações, com aplicações em áreas como comércio eletrônico e tradicional, seguros, saúde, geoprocessamento, estudos sócio-econômicos (Liu, Hsu, Chen, & Ma 2000; Clementini, Felice, & Koperski 2000; Semenova, Hegland, Graco, & Williams 2001).

Em termos gerais, a técnica de Regras de Associação permite identificar padrões intra-transações em uma base de dados de transações, ou seja, uma Regra de Associação caracteriza o quanto a presença de um conjunto de atributos nos registros de uma base de dados implica na presença de algum outro conjunto distinto de atributos nos mesmos registros.

Desse modo, a possibilidade de descoberta de todas as associações existentes nas transações de uma base de dados é o aspecto de maior relevância da técnica. Entretanto, esta característica determina a geração de um número elevado de regras, dificultando sobremaneira a interpretação do conjunto de regras pelo usuário. Além disso, em alguns casos pode levar à descoberta de conhecimento muito óbvio ou irrelevante que pouco acrescenta ao conhecimento já existente sobre o domínio. No processo de Extração de Conhecimento por Regras de Associação, a etapa Pós-processamento é importante para a resolução dessa problemática.

Um dos grandes entraves na elaboração de um sistema para Pós-processamento de Conhecimento é que os algoritmos de identificação de padrões geralmente armazenam o conhecimento em um formato proprietário, dificultando assim a integração e interpretação do conhecimento extraído por diferentes algoritmos.

Desse modo, é apresentada neste trabalho uma sintaxe padrão para representar Regras de Associação. Essa sintaxe objetiva unificar a forma de representação das regras obtidas e de algumas métricas associadas.

Este relatório está organizado com os conceitos centrais sobre Regras de Associação e matriz de contingência apresentados na Seção 2, na Seção 3 são descritas as gramáticas da Sintaxe Padrão e da Sintaxe Padrão Compacta e procedimento para cálculo da matriz de contingência a partir de métricas geradas pelos aplicativos de Mineração de Dados. Na Seção 4 são descritos os aplicativos utilizados e as peculiaridades do processo de conversão dos arquivos de regras para o Formato Padrão de Regras de Associação. As considerações finais são realizadas na Seção 5 e na seqüência são apresentadas a listagem do código da classe de conversão e as referências bibliográficas utilizadas.

2 Conceitos Gerais

Nesta seção é apresentada a técnica de Regras de Associação, com seus principais conceitos e definições e, também, a definição da matriz de contingência, a partir da qual diversas medidas para avaliação de regras podem ser derivadas.

2.1 Regras de Associação

A questão das Regras de Associação foi abordada primeiramente por (Agrawal, Imielinski, & Swami 1993). No referido trabalho, os autores trabalham o conceito com uma base de dados binária. Trabalhos subseqüentes (Agrawal & Srikant (1994), Adamo (2001), dentre outros) buscam ampliar e generalizar esta abordagem, considerando outros aspectos, tais como, atributos contínuos, taxonomia de atributos, bases de dados relacionais, bases de dados distribuídas e processamento paralelo.

De forma geral, uma Regra de Associação caracteriza o quanto a presença de um conjunto de atributos s nos registros de uma base de dados implica na presença de algum outro conjunto distinto de atributos u nos mesmos registros (Agrawal & Srikant 1994). O formato da regra pode ser apresentado como:

$$R : if \underbrace{< antecedente >}_{LHS} then \underbrace{< consequente >}_{RHS}$$

ou, de modo simplificado como:

$$R : LHS \Rightarrow RHS$$

sendo LHS e RHS, respectivamente, o lado esquerdo (*Left Hand Side*) e o lado direito (*Right Hand Side*) da regra, definidos por conjuntos disjuntos de atributos.

Segundo Agrawal & Srikant (1994) pode-se definir formalmente Regras de Associação da seguinte forma:

> Seja $A = \{a_1, a_2, ..., a_m\}$ um conjunto de atributos. Seja T um conjunto de transações, onde cada transação t_i é composta por um conjunto de atributos tal que $t_i \subseteq A$.
>
> A Regra de Associação é uma implicação na forma $LHS \Rightarrow RHS$, onde $LHS \subset A$, $RHS \subset A$ e $LHS \cap RHS = \varnothing$. A regra $LHS \Rightarrow RHS$ ocorre no conjunto de transações T com Confiança = **conf** se em **conf %** das transações em T em que ocorre LHS, ocorre também RHS. A regra $LHS \Rightarrow RHS$ tem Suporte = **sup** se em **sup %** das transações em T ocorre $LHS \cup RHS$.

As medidas Suporte e Confiança são as medidas mais empregadas em Regras de Associação, tanto no que se refere à etapa de Pós-processamento do conhecimento adquirido,

quanto à etapa de seleção dos subconjuntos de atributos durante o processo de geração das regra. Sendo assim, buscando facilitar a compreensão das medidas, são apresentadas as formas alternativas de descrição de Suporte e Confiança:

Suporte quantifica a incidência da regra no conjunto de dados, ou seja, indica a freqüência com que LHS e RHS ocorrem juntos no conjunto de dados. É equivalente à probabilidade $p(LHS, RHS)$, isto é, a probabilidade de que LHS e RHS ocorram simultaneamente.

Confiança indica a freqüência com que LHS e RHS ocorrem juntos em relação ao número total de registros em que LHS ocorre. É equivalente à probabilidade condicional $p(RHS|LHS)$, ou seja, representa a probabilidade de ocorrência de RHS, dado que LHS ocorre.

Vale ressaltar que nem todas as regras descobertas em um conjunto de transações são interessantes ou úteis. Como forma de prevenir esta ocorrência, algumas restrições devem ser estabelecidas. As mais comumente empregadas são Suporte mínimo e Confiança mínima. Neste caso, são consideradas exclusivamente regras com valores de Suporte e Confiança superiores aos valores mínimos definidos. O emprego destas restrições leva à redução do número de regras geradas e também do esforço (tempo) de processamento. Mesmo assim, o número de regras geradas ainda pode ser muito elevado, requerendo técnicas adequadas de Pós-processamento.

2.2 Matriz de Contingência

A matriz de contingência é um caso particular da matriz de confusão, utilizada em testes de hipóteses, quando são apenas dois elementos a serem comparados.

Considerando uma regra R e uma transação T_i, podemos aplicar a regra à transação e verificar se os atributos estão presentes ou não. Isso resulta em quatro situações possíveis:

1. Os atributos de LHS e RHS estão presentes na transação, LHS e RHS são verdadeiros;
2. Os atributos de LHS estão presentes na transação mas os de RHS não, LHS é verdadeiro mas, RHS é falso;
3. Os atributos de LHS não estão presentes na transação mas os de RHS estão, LHS é falso mas, RHS é verdadeiro;
4. Os atributos de LHS e RHS não estão presentes na transação, LHS e RHS são falsos;

A matriz de contingência para uma regra $LHS \Rightarrow RHS$ representa sua cobertura. Na Tabela 1, LHS denota o antecedente da regra, e $\overline{LHS}$ o complemento de LHS; de

forma análoga para o conseqüente RHS e $\overline{RHS}$. $n(LHS\ RHS)$ denota o número de transações no qual LHS e RHS são verdadeiros; $n(\overline{LHS}\ RHS)$ denota o número de exemplos no qual LHS é falso e RHS é verdadeiro; de forma similar para os demais elementos. Complementarmente N denota o número total de exemplos.

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

Tabela 1: Matriz de contingência

De forma genérica a freqüência relativa associada ao conjunto X pode ser calculada como $n(X)/N$ e denotada por $f(X)$. Assim como definida, a freqüência relativa pode ser utilizada como uma estimativa de probabilidade (Tabela 2). A notação $f(X|Y)$ segue sua definição usual em probabilidade, dada por:

$$f(X|Y) = \frac{f(XY)}{f(Y)} \tag{1}$$

sendo X e Y subconjuntos das N transações.

	RHS	$\overline{RHS}$	
LHS	$f(LHS\ \ RHS)$	$f(LHS\ \ \overline{RHS})$	$f(LHS)$
$\overline{LHS}$	$f(\overline{LHS}\ \ RHS)$	$f(\overline{LHS}\ \ \overline{RHS})$	$f(\overline{LHS})$
	$f(RHS)$	$f(\overline{RHS})$	1

Tabela 2: Matriz de contingência com freqüências relaivas

Os valores da matriz de contingência de uma regra são calculados pela comparação dessa regra com cada exemplo da base de dados. No caso das Regras de Associação, esse processo de cálculo pode ser extremamente custoso devido ao elevado número de regras. Um modo alternativo para a elaboração da matriz de contingência é apresentado na Seção 3.2.

3 Formato Padrão das Regras

Regras de Associação é uma técnica descritiva de Mineração de Dados, que utiliza regras como formato de representação das hipóteses geradas a partir da aplicação dos algoritmos. Apesar da forma de representação das hipóteses ser semelhante, cada algoritmo salva regras com sintaxes diferentes. Os algoritmos além das regras, fornecem as métricas Suporte e Confiança, geralmente acompanhadas de métricas complementares – *Lift*, Confiança Esperada, *Leverage*, entre outras – variando de algoritmo para algoritmo.

Devido às diferentes sintaxes de representação das regras geradas por esses algoritmos, qualquer medida ou estatística que necessite ser calculada sobre as regras teria de ser implementada várias vezes, uma para cada sintaxe. Do mesmo modo, a interpretação das regras pelo usuário fica dificultada, já que este terá que lidar com símbolos e formatos diferentes a cada conjunto de regras.

Para propiciar ao usuário um conjunto de informações padronizadas sobre cada regra e para evitar retrabalho nas implementações das métricas de avaliação, é definida neste relatório técnico uma sintaxe padrão para representação de Regras de Associação. Com base nessa sintaxe, padronizam-se as saídas dos algoritmos de associação, convertendo-as para um formato único em que as regras são não ordenadas.

Considerando que a maioria das medidas objetivas para avaliação de regras encontradas na literatura podem ser derivadas da matriz de contingência, o formato padrão para Regras de Associação contém as freqüências relativas $f(LHS \quad RHS)$, $f(LHS \quad \overline{RHS})$, $f(\overline{LHS} \quad \overline{RHS})$ e $f(\overline{LHS} \quad RHS)$, além do número total de exemplos considerados N. Os valores absolutos das medidas podem ser obtidos pela multiplicação das freqüências relativas por N. As freqüências marginais $f(LHS)$, $f(RHS)$, $f(\overline{LHS})$ e $f(\overline{RHS})$ também podem ser obtidas pela soma das linhas e colunas da matriz de contingência (Tabela 1, página 4). Os valores de freqüência relativa e número de exemplos, comporão a sintaxe padrão do seguinte modo:

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

Esse conjunto de informações é calculado a partir das métricas fornecidas pelos algoritmos de Regras de Associação, conforme apresentado na Seção 3.2.

3.1 Gramática da Sintaxe Padrão

Para a definição de uma sintaxe padrão para Regras de Associação, é especificada uma gramática a ser utilizada na conversão das regras geradas pelos algoritmos de associação para essa sintaxe. A definição de tal gramática levou em consideração a padronização definida pelo grupo do LABIC que trabalha com Extração de Conhecimento de Dados. Foi considerada também, a necessidade de otimização para o processamento dos arquivos de regras e a compreensibilidade das regras na sintaxe padrão para os usuários.

Nesse contexto, foi proposta a Sintaxe Padrão para Regras de Associação, que possui boa compreensibilidade e legibilidade, provendo fácil leitura e interpretação das regras pelo usuário. Contudo, considerando o elevado número de regras normalmente gerado, e conseqüentemente arquivos de regras de tamanho considrável, foi necessário o desenvolvimento de uma Sintaxe Padrão Compacta, que permite um processamento mais ágil e proporciona arquivos de regras de tamanho consideravelmente inferior (de 30% a 40% menor, conforme alguns experimentos), porém a compreensibilidade desta sintaxe é significativamente menor se comparada à Sintaxe Padrão, sendo necessária a manutenção das duas sintaxes.

3.1.1 Sintaxe Padrão para Regra de Associação

A gramática que define a Sintaxe Padrão para Regras de Associação, é definida como uma quádrupla $G = (V_n, V_t, P, S)$:

S: <regra associação>

V_n: { <regra associação>, <nro regra>, <regra_a>, <extensão>, <antecedente>, <conseqüente>, <fator>, <termo>, <valor>, <valor numérico>, <valor categórico>, <comparação>, <operador relacional>, <identificador>, <sinal>, <dígitos>,<dígito>, <letra> }

V_t:{ IF, THEN, AND, <=, >=, =, <, >, <>, +-, **{**, **}**, [,], +, -, **.**, **,**, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, _ }

P: é descrito usando a notação EBNF e representado na Figura 1.

$[\alpha] = \alpha|\lambda$

$\{\alpha\}$ = repetição da cadeia α zero ou mais vezes

$\alpha|\beta = \alpha$ ou β devem ser escolhidos

Não terminais aparecem entre < e >, e terminais aparecem em negrito.

As regras geradas são convertidas para a Sintaxe Padrão de Regras de Associação utilizando a gramática de conversão apresentada anteriormente.

O exemplo de regra apresentado na Figura 2 convertido para a Sintaxe Padrão de Regras de Associação pode ser observado na Figura 3. Deve-se lembrar que no formato original, as regras são armazenadas em arquivo-texto, com uma regra por linha. Na Sintaxe Padrão as regras ainda são armazenadas em arquivo-texto, só que não mais uma regra por linha.

Observe-se também que algumas informações fornecidas pela regra original (por exemplo: 2 2 7.1429 100) não são armazenadas na Sintaxe Padrão, uma vez que essas medidas normalmente são distintas para cada algoritmo de associação. As informações necessárias à avaliação de regras – matriz de contingência – são armazenadas de modo padronizado e independe do algoritmo que as geraram.

```
<regra associação> ::= <nro regra> , <regra_a> , <extensão>
<nro regra> ::= [ R <dígitos> ]
<regra_a> ::=  IF <antecedente>  THEN <conseqüente>
<extensão>::= [<valor numérico> , <valor numérico> , <valor numérico> ,
                                  <valor numérico> , <dígitos> ]
<antecedente> ::= TRUE | <fator>
<conseqüente> ::= <fator>
<fator> ::= <termo> | <termo> AND <fator>
<termo> ::= <identificador> | <comparação>
<comparação> ::= <identificador> <operador relacional> <valor numérico>
               | <identificador> <operador relacional> <valor categórico>
<operador relacional> ::=  <= | >= | = | < | > | <>
<valor categórico> ::= <identificador>
<valor numérico> ::= [ <sinal> ] <dígitos> [ . <dígitos> ]
             | [ <sinal> ] <dígitos> [ . <dígitos> ]  e <sinal> <dígitos>
<sinal> ::= + | -
<dígitos> ::= <dígito> { <dígito> }
<dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<identificador> ::= <letra> | <dígito> { <letra> | <dígito> }
<letra> ::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o
          | p | q | r | s | t | u | v | w | x | y | z | A | B | C | D
          | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S
          | T | U | V | W | X | Y | Z | _
```

Figura 1: Produções da Gramática para Regras de Associação

3.1.2 Sintaxe Padrão Compacta para Regras de Associação

De modo semelhante a gramática da Sintaxe Padrão, a gramática que define a Sintaxe Padrão Compacta para Regras de Associação, é definida como uma quádrupla $G = (V_n, V_t, P, S)$:

V_n: { <regra associação>, <nro regra>, <regra_a>, <extensão>, <antecedente>,

```
2 2 7.1429 100 42.86 2.3333  outlook=overcast and temperature=cool
                                          humidity=normal and play=yes
```

Figura 2: Regra de associação Gerada pelo Algoritmo do MineSet^{TM}

```
[R0547] IF outlook=overcast
        AND temperature=cool
        THEN humidity=normal
        AND play=yes
        [0.071429,0.000000,0.571400,0.357171,14]
```

Figura 3: Regra de associação da Figura 2 na Sintaxe Padrão

<conseqüente>, <fator>, <termo>, <valor>, <valor numérico>, <valor categórico>, <comparação>, <operador relacional>, <identificador>, <sinal>, <dígitos>,<dígito>, <letra> }

V_t: { &, <=, >=, =, <, >, <>, +-, **{**, **}**, [,], +, -, **.**, **,**, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, _ }

P: é descrito usando a notação EBNF e representado na Figura 4.

$[\alpha] = \alpha|\lambda$

$\{\alpha\}$ = repetição da cadeia α zero ou mais vezes

$\alpha|\beta = \alpha$ ou β devem ser escolhidos

Não terminais aparecem entre < e >, e terminais aparecem em negrito.

As regras geradas são convertidas para a Sintaxe Padrão Compacta de Regras de Associação utilizando a gramática de conversão apresentada anteriormente. A regra apresentada na Figura 2, convertida para a Sintaxe Padrão Compacta pode ser vista na Figura 5. Ressalta-se que todas as regras convertidas também são armazenadas em arquivo-texto e com uma regra por linha. Assim como na Sintaxe Padrão, também são desconsideradas algumas informações presentes no arquivo de regras original.

3.2 Cálculo da Matriz de Contingência

Diferentemente de outras técnicas de Mineração de Dados, a técnica de Regras de Associação gera um número elevado de regras (da ordem de milhares ou centenas de

```
<regra associação> ::= <nro regra> , <regra_a> , <extensão>
<nro regra> ::= [ R <dígitos> ]
<regra_a> ::= <antecedente> , <conseqüente>
<extensão>::= [<valor numérico> , <valor numérico> , <valor numérico> ,
                                  <valor numérico> , <dígitos> ]
<antecedente> ::= TRUE | <fator>
<conseqüente> ::= <fator>
<fator> ::= <termo> | <termo> & <fator>
<termo> ::= <identificador> | <comparação>
<comparação> ::= <identificador> <operador relacional> <valor numérico>
               | <identificador> <operador relacional> <valor categórico>
<operador relacional> ::= <= | >= | = | < | > | <>
<valor categórico> ::= <identificador>
<valor numérico> ::= [ <sinal> ] <dígitos> [ . <dígitos> ]
             | [ <sinal> ] <dígitos> [ . <dígitos> ] e <sinal> <dígitos>
<sinal> ::= + | -
<dígitos> ::= <dígito> { <dígito> }
<dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<identificador> ::= <letra> | <dígito> { <letra> | <dígito> }
<letra> ::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o
          | p | q | r | s | t | u | v | w | x | y | z | A | B | C | D
          | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S
          | T | U | V | W | X | Y | Z | _
```

Figura 4: Produções da Gramática para Regras de Associação no Formato Compacto

milhares de regras), com cada regra representando um conhecimento único e independente de outra regras.

A elaboração da matriz de contingência para essa elevada quantidade de regras, demanda tempo considerável, já que é necessário avaliar para cada regra todos os exemplos existentes na base de dados, ou seja, uma passagem pela base de dados para cada regra,

```
[R0547],outlook=overcast & temperature=cool,humidity=normal &
                play=yes, [0.071429,0.000000,0.571400,0.357171,14]
```

Figura 5: Regra de associação da Figura 2 na Sintaxe Padrão Compacta

assim, para uma conjunto de um milhão de regras seria necessário um milhão de passagens pela base, o que evidentemente tornaria o processo muito lento.

Como modo de contornar esta dificuldade, a alternativa adotada é determinar a matriz de contingência a partir de métricas fornecidas pelos algoritmos, derivadas da matriz de contingência, e que são calculadas durante o processo de geração das regras.

Suporte e Confiança são duas métricas fornecidas pelos algoritmos de Regras de Associação, contudo são insuficientes para elaborar a matriz de contingência, assim é necessário o emprego de uma outra métrica. A Confiança Esperada, o *Lift* (razão entre Confiança e Confiança Esperada) ou a Diferença de Confiança Esperada (diferença entre Confiança e Confiança Esperada), são métricas usualmente fornecidas pelos algoritmos em conjunto com Suporte e Confiança. Considerando que a Confiança Esperada pode ser obtida de maneira simples a partir das outras duas medidas mencionadas, esta foi adotada para ilustrar os cálculos da matriz de contingência.

Na seqüencia é apresentado como a matriz de contingência é elaborada a partir do Suporte, Confiança e Confiança Esperada. Estas medidas são definidas, respectivamente, pelas Equações 2, 3 e 4, como se segue:

$$Suporte = p(LHS \;\; RHS) = \frac{n(LHS \;\; RHS)}{N} \tag{2}$$

$$\begin{aligned} Confianca &= p(RHS|LHS) = \frac{p(LHS \;\; RHS)}{p(LHS)} \\ &= \frac{(\frac{n(LHS \;\; RHS)}{N})}{\frac{n(LHS)}{N}} = \frac{n(LHS \;\; RHS)}{n(LHS)} \end{aligned} \tag{3}$$

$$ConfEsp = p(RHS) = \frac{n(RHS)}{N} \tag{4}$$

As Equações 5 e 6 que representam o número de verdadeiros e o número de negativos, respectivamente, também auxiliarão na elaboração da matriz de contingência.

Das Equações 2 e 3 é determinado n(LHS):

$$\begin{aligned} n(LHS) &= p(LHS) \cdot N \\ &= N \cdot \frac{p(LHS\ \ RHS)}{p(RHS|LHS)} \\ &= N \cdot \frac{Suporte}{Confianca} \end{aligned} \tag{5}$$

Da Equação 4 é determinado n($\overline{RHS}$):

$$\begin{aligned} n(\overline{RHS}) &= N - n(RHS) \\ &= N - ConfEsp \cdot N \\ &= N \cdot (1 - ConfEsp) \end{aligned} \tag{6}$$

A partir das Equações 2 a 6 são definidos os valores da matriz de contingência, expressos pelas Equações 7 a 10:

- n($LHS\ RHS$): número de verdadeiros positivos, pode ser obtido a partir da Equação 2.

$$n(LHS\ \ RHS) = p(LHS\ \ RHS) \cdot N = Suporte \cdot N \tag{7}$$

- n($LHS\ \overline{RHS}$): número de falsos negativos, pode ser obtido a partir das Equações 5 e 7.

$$\begin{aligned} n(LHS\ \ \overline{RHS}) &= n(LHS) - n(LHS\ \ RHS) \\ &= N \cdot \frac{Suporte}{Confianca} - Suporte \cdot N \\ &= N \cdot Suporte \cdot \left(\frac{1}{Confianca} - 1 \right) \end{aligned} \tag{8}$$

- n($\overline{LHS}\ RHS$): número de falsos positivos, pode ser obtido a partir das Equações 4 e 7.

$$\begin{aligned} n(\overline{LHS}\ \ RHS) &= n(RHS) - n(LHS\ \ RHS) \\ &= ConfEsp \cdot N - Suporte \cdot N \\ &= N \cdot (ConfEsp - Suporte) \end{aligned} \tag{9}$$

- n($\overline{LHS}\ \ \overline{RHS}$): número de verdadeiros negativos, pode ser obtido a partir das Equações 6 e 8.

$$\begin{aligned} n(\overline{LHS}\ \ \overline{RHS}) &= n(\overline{RHS}) - n(LHS\ \ \overline{RHS}) \\ &= N \cdot (1 - ConfEsp) - N \cdot Suporte \cdot \left(\frac{1}{Confianca} - 1\right) \\ &= N \cdot \left(1 - ConfEsp - Suporte \cdot \left(\frac{1}{Confianca} - 1\right)\right) \quad (10) \end{aligned}$$

4 Conversão de Formatos

O processo de conversão das regras para a sintaxe padrão de Regras de Associação é realizado pela classe **Rass_Conv** que contém os métodos que convertem a saída de cada algoritmo de associação para a Sintaxe Padrão ou para a Sintaxe Padrão Compacta. A classe também permite a conversão entre as duas sintaxes. Na Figura 6 é apresentado o esquema funcional do módulo de conversão de sintaxes.

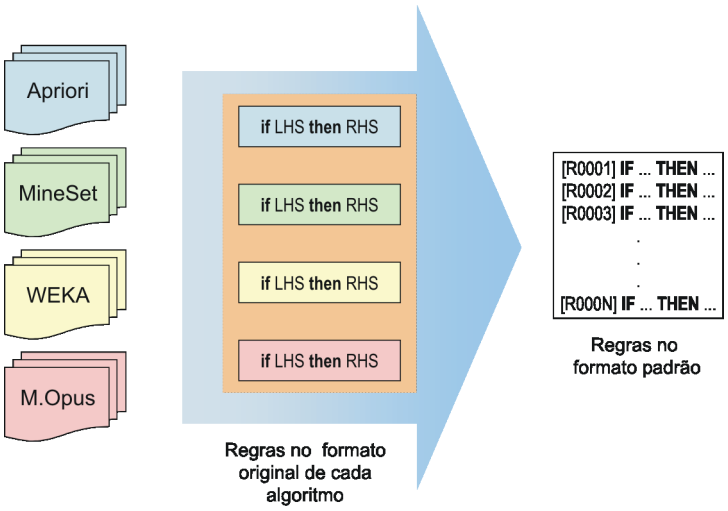

Figura 6: Módulo de Conversão das Saídas dos Algoritmos de Regras de Associação

No restante desta seção são descritos os sistemas e algoritmos de associação para os quais foram implementados os conversores, assim como exemplos de regras de associação no formato original de cada sistema e depois de convertidas para o formato padrão de Regras de Associação. Os aplicativos *Apriori* e *WEKA* são *freeware*, o MineSet^{TM}existe uma versão licenciada para o LABIC e o aplicativo *Magnum Opus* foi utilizado em sua versão demonstrativa.

Para facilitar a compreensão das regras geradas, os algoritmos foram executados uti-

lizando a base *Weather Nominal* apresentada na Tabela 3, que consta do conjunto de exemplos do aplicativo *WEKA*.

Tabela 3: Base de dados *Weather Nominal*

outlook	temperature	humidity	windy	play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

4.1 *Apriori*

O aplicativo *Apriori* é uma implementação de Chistian Borgelt do tradicional algoritmo de Regras de Associação, de mesmo nome desenvolvido por Agrawal, Imielinski, & Swami (1993). O aplicativo foi desenvolvido na filosofia GNU, sendo utilizado para a elaboração do módulo conversor a versão 4.08. Em sua versão 2.7, compos o módulo de Regras de Associação do software Clementine 7.0 da SPSS*. Como características principais desta implementação do Apriori, devem ser destacadas:

- utiliza definição de Suporte diferente da original, contudo é possível a utilização da definição original se Suporte, adicionando-se a opção `-o` na linha de comando;
- permite utilizar métricas complementares na geração de regras. As métricas são ativadas pela opção `-e` da linha de comando, seguida pelo número da métrica. Adicionalmente é necessário especificar o valor mínimo da métrica, dado pela opção `-d` seguido do referido valor;
- permite gerar além das regras apenas os *itemsets*;
- para que exista compatibilidade com o módulo conversor, o aplicativo deve ser executado com as opções: `-c# -s# -o -a -v -e1`.

*www.spss.com

A opção `-o` indica que devem ser utilizadas as definições originais de Suporte e Confiança definidas por Agrawal, Imielinski, & Swami (1993), a opção `-e1 -d0` indica que as regras além de satisfazer a condição de Suporte e Confiança mínimos, deve satisfazer a condição de "Diferença de Confiança Esperada" superior ao valor especificado na opção `-d` que, para não afetar os resultados, deve possuir o valor 0 (zero). A matriz de contingência poderá ser derivada das medidas Suporte, Confiança e Diferença de Confiança Esperada. Esse recurso é usado para eliminar a necessidade de percorrer a base de dados para construir a matriz de contingência para cada regra. A Diferença de Confiança Esperada representa a diferença entre a Confiança e a Confiança Esperada para uma regra, como já mencionado.

Na Figura 7 é apresentado um exemplo do arquivo de saída gerado pelo aplicativo *Apriori* e, na Figura 8, as regras desse exemplo na Sintaxe Padrão Compacta.

```
play=no <-  (35.7%, 35.7%, 0.0%)
temperature=mild <-  (42.9%, 42.9%, 0.0%)
windy=FALSE <-  (57.1%, 57.1%, 0.0%)
play=yes <-  (64.3%, 64.3%, 0.0%)
outlook=sunny <- temperature=cool  (7.1%, 25.0%, 10.7%)
temperature=cool <- outlook=sunny  (7.1%, 20.0%, 8.6%)
play=no <- temperature=cool  (7.1%, 25.0%, 10.7%)
temperature=cool <- play=no  (7.1%, 20.0%, 8.6%)
outlook=overcast <- temperature=hot humidity=normal  (7.1%, 100.0%, 71.4%)
temperature=hot <- outlook=overcast humidity=normal  (7.1%, 50.0%, 21.4%)
windy=FALSE <- temperature=hot outlook=overcast  (14.3%, 100.0%, 42.9%)
humidity=normal <- outlook=overcast play=yes  (14.3%, 50.0%, 0.0%)
play=yes <- temperature=hot outlook=overcast humidity=high  (7.1%, 100.0%, 35.7%)
humidity=high <- temperature=hot outlook=overcast play=yes  (7.1%, 50.0%, 0.0%)
play=yes <- outlook=rainy temperature=mild humidity=normal windy=FALSE  (7.1%, 100.0%, 35.7%)
windy=FALSE <- outlook=rainy temperature=mild humidity=normal play=yes  (7.1%, 100.0%, 42.9%)
humidity=normal <- outlook=rainy temperature=mild windy=FALSE play=yes  (7.1%, 50.0%, 0.0%)
```

Figura 7: Algumas Regras Geradas pelo *Apriori* Utilizando o Conjunto de Dados *Weather-Nominal*

4.2 *Magnum Opus*

O aplicativo *Magnum Opus*, comercializado pela Rule Quest[†], é a implementação do algoritmo Opus proposto por Webb (1995). Apresenta como principais características:

- adicionalmente ao Suporte, considera as métricas *Coverage*, *Lift*, *Leverage* e *Strength*;
- definição do número máximo de regras a serem geradas;
- permite definir o número máximo de atributos em LHS, mas gera regras com apenas um item em RHS;
- possui filtros para regras consideradas improdutivas, triviais e insignificantes;

[†]www.rulequest.com

```
TRUE,play=no,[0.357000,0.643000,0.000000,0.000000,14]
TRUE,temperature=mild,[0.429000,0.571000,0.000000,0.000000,14]
TRUE,windy=FALSE,[0.571000,0.429000,0.000000,0.000000,14]
TRUE,play=yes,[0.643000,0.357000,0.000000,0.000000,14]
temperature=cool,outlook=sunny,[0.071000,0.213000,0.644000,0.072000,14]
outlook=sunny,temperature=cool,[0.071000,0.284000,0.602000,0.043000,14]
temperature=cool,play=no,[0.071000,0.213000,0.644000,0.072000,14]
play=no,temperature=cool,[0.071000,0.284000,0.602000,0.043000,14]
temperature=hot & humidity=normal,outlook=overcast,[0.071000,0.000000,0.714000,0.215000,14]
outlook=overcast & humidity=normal,temperature=hot,[0.071000,0.071000,0.643000,0.215000,14]
temperature=hot & outlook=overcast,windy=FALSE,[0.143000,0.000000,0.429000,0.428000,14]
outlook=overcast & play=yes,humidity=normal,[0.143000,0.143000,0.357000,0.357000,14]
temperature=hot & outlook=overcast & humidity=high,play=yes,[0.071000,0.000000,0.357000,0.572000,14]
temperature=hot & outlook=overcast & play=yes,humidity=high,[0.071000,0.071000,0.429000,0.429000,14]
outlook=rainy & temperature=mild & humidity=normal & windy=FALSE,play=yes,
                                                                [0.071000,0.000000,0.357000,0.572000,14]
outlook=rainy & temperature=mild & humidity=normal & play=yes,windy=FALSE,
                                                                [0.071000,0.000000,0.429000,0.500000,14]
outlook=rainy & temperature=mild & windy=FALSE & play=yes,humidity=normal,
                                                                [0.071000,0.071000,0.429000,0.429000,14]
```

Figura 8: Regras da Figura 7 Convertidas para o Formato Padrão Compacto de Regras de Associação

- possui recursos, apesar de limitados, para tratar atributos contínuos.

Na Figura 9 é apresentado um exemplo do arquivo de saída gerado pelo aplicativo *Magnum Opus* e, na Figura 10, as regras desse exemplo na Sintaxe Padrão Compacta.

```
LHS,RHS,Coverage,(Coverage),Support,(Support),Strength,Lift,Leverage,(Leverage)
humidity=normal,play=yes,0.500,7,0.429,6,0.857,1.33,0.1071,1
humidity=normal,temperature=cool,0.500,7,0.286,4,0.571,2.00,0.1429,2
humidity=high,play=no,0.500,7,0.286,4,0.571,1.60,0.1071,1
humidity=normal & windy=FALSE,play=yes,0.286,4,0.286,4,1.000,1.56,0.1020,1
humidity=normal & play=yes,temperature=cool,0.429,6,0.214,3,0.500,1.75,0.0918,1
outlook=sunny & humidity=high,play=no,0.214,3,0.214,3,1.000,2.80,0.1378,1
outlook=rainy & windy=FALSE & play=yes,humidity=normal,0.214,3,0.143,2,0.667,1.33,0.0357,0
temperature=cool & humidity=normal & windy=FALSE,play=yes,0.143,2,0.143,2,1.000,1.56,0.0510,0
outlook=sunny & temperature=hot & humidity=high,play=no,0.143,2,0.143,2,1.000,2.80,0.0918,1
temperature=cool & humidity=normal & windy=FALSE & play=yes,outlook=rainy,
                                                                0.143,2,0.071,1,0.500,1.40,0.0204,0
outlook=overcast & temperature=cool & humidity=normal & windy=TRUE,play=yes,
                                                                0.071,1,0.071,1,1.000,1.56,0.0255,0
outlook=rainy & temperature=cool & humidity=normal & windy=TRUE,play=no,
                                                                0.071,1,0.071,1,1.000,2.80,0.0459,0
outlook=rainy & temperature=cool & windy=TRUE,play=no,0.071,1,0.071,1,1.000,2.80,0.0459,0
outlook=rainy & temperature=cool & humidity=normal & windy=FALSE,play=yes,
                                                                0.071,1,0.071,1,1.000,1.56,0.0255,0
outlook=rainy & temperature=cool & play=yes,windy=FALSE,0.071,1,0.071,1,1.000,1.75,0.0306,0
```

Figura 9: Algumas Regras Geradas pelo *Magnum Opus* Utilizando o Conjunto de Dados *Weather-Nominal*

```
[R0001],humidity=normal,play=yes,[0.429000,0.071583,0.284056,0.215361,14]
[R0002],humidity=normal,temperature=cool,[0.286000,0.214876,0.499624,-0.000500,14]
[R0003],humidity=high,play=no,[0.286000,0.214876,0.428249,0.070875,14]
[R0004],humidity=normal & windy=FALSE,play=yes,[0.286000,0.000000,0.358974,0.355026,14]
[R0005],humidity=normal & play=yes,temperature=cool,[0.214000,0.214000,0.500286,0.071714,14]
[R0006],outlook=sunny & humidity=high,play=no,[0.214000,0.000000,0.642857,0.143143,14]
[R0007],outlook=rainy & windy=FALSE & play=yes,humidity=normal,[0.143000,0.071393,0.427103,0.358504,14]
[R0008],temperature=cool & humidity=normal & windy=FALSE,play=yes,[0.143000,0.000000,0.358974,0.498026,14]
[R0009],outlook=sunny & temperature=hot & humidity=high,play=no,[0.143000,0.000000,0.642857,0.214143,14]
[R0010],temperature=cool & humidity=normal & windy=FALSE & play=yes,outlook=rainy,
                                                                [0.071000,0.071000,0.571857,0.286143,14]
[R0011],outlook=overcast & temperature=cool & humidity=normal & windy=TRUE,play=yes,
                                                                [0.071000,0.000000,0.358974,0.570026,14]
[R0012],outlook=rainy & temperature=cool & humidity=normal & windy=TRUE,play=no,
                                                                [0.071000,0.000000,0.642857,0.286143,14]
[R0013],outlook=rainy & temperature=cool & windy=TRUE,play=no,[0.071000,0.000000,0.642857,0.286143,14]
[R0014],outlook=rainy & temperature=cool & humidity=normal & windy=FALSE,play=yes,
                                                                [0.071000,0.000000,0.358974,0.570026,14]
[R0015],outlook=rainy & temperature=cool & play=yes,windy=FALSE,[0.071000,0.000000,0.428571,0.500429,14]
```

Figura 10: Regras da Figura 9 Convertidas para o Formato Padrão Compacto de Regras de Associação

4.3 *WEKA*

O ambiente *WEKA* (*Waikato Environment for Knowledge Analysis*), desenvolvido na Universidade de Waikato na Nova Zelândia[‡] dentro da filosofia GNU de código fonte aberto. Possui ferramentas para pré-processamento de dados, classificação, regressão, *clustering*, Regras de Associação e visualização.

Os diversos algoritmos de Aprendizado de Máquina, implementados na linguagem Java, são representados cada qual por uma classe, o que permite que as classes sejam organizadas em pacotes, que são simplesmente diretórios contendo uma coleção de classes relacionadas. Isso é bastante útil porque certos algoritmos compartilham muitas funcionalidades, e desse modo, várias classes em um pacote podem ser utilizadas por mais de um algoritmo (Witten & Frank 1999). O pacote **weka.associations** contém a implementação do algoritmo Apriori proposto por Agrawal, Imielinski, & Swami (1993).

Para que exista compatibilidade com a classe desenvolvida para conversão para a Sintaxe Padrão de Regras de Associação, é necessário que o módulo de Regras de Associação do *WEKA* seja executado com a opção *metricType* diferente de *Confidence* (Confinança), assim serão calculadas métricas adicionais que permitirão a elaboração da matriz de contingência sem a necessidade de consultas à base de dados.

As características a serem destacadas são:

- inclusão de novos algoritmos de maneira simplificada;
- adicionalmente ao Suporte e Confiança, permite considerar as métricas *Lift*, *Leverage* e *Conviction*;

[‡]www.waikato.ac.nz

- permite estabelecer o número máximo de regras a serem geradas;
- exige, para Regras de Associação, que os atributos sejam categóricos.

Na Figura 11 é apresentado um exemplo do arquivo de saída gerado pelo aplicativo *WEKA* e, na Figura 12, as regras desse exemplo na Sintaxe Padrão Compacta.

4.4 MineSet^{TM}

O aplicativo MineSet^{TM}da Silicon Graphics possui implementado, entre outros algoritmos de Mineração de Dados, um específico para Regras de Associação. As principais características deste aplicativo, referentes a Regras de Associação, são:

- além das métricas Suporte e Confiança, gera também as métricas Confiança Esperada e Lift;
- permite controlar o número de atributos por regra;
- salva as regras geradas em diversos formatos, dentre os quais MineSet^{TM} binário e Ascii (este utilizado pelo módulo conversor);
- discretiza automaticamente atributos contínuos.

Na Figura 13 é apresentado um exemplo do arquivo de saída gerado pelo aplicativo. Nas Figuras 14 e 15 são apresentadas as regras convertidas para o Formato Padrão Compacto e para o Formato Padrão de Regras de Associação, respectivamente.

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 100000 -T 1 -C 0.1 -D 0.02 -U 1.0 -M 0.02 -S -1.0
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.02
Minimum metric <lift>: 0.1
Number of cycles performed: 48

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12
Size of set of large itemsets L(2): 54
Size of set of large itemsets L(3): 96
Size of set of large itemsets L(4): 64
Size of set of large itemsets L(5): 14

Best rules found:

     1. humidity=normal play=no 1 ==> outlook=rainy temperature=cool windy=TRUE 1
                conf:(1) < lift:(14)> lev:(0.07) [0] conv:(182)
     2. temperature=cool play=no 1 ==> outlook=rainy humidity=normal windy=TRUE 1
                conf:(1) < lift:(14)> lev:(0.07) [0] conv:(182)
     3. outlook=rainy humidity=normal windy=TRUE 1 ==> temperature=cool play=no 1
                conf:(1) < lift:(14)> lev:(0.07) [0] conv:(182)
   420. humidity=normal play=yes 6 ==> outlook=rainy temperature=cool windy=FALSE 1
                conf:(0.17) < lift:(2.33)> lev:(0.04) [0] conv:(182)
   421. temperature=cool play=yes 3 ==> outlook=rainy humidity=normal windy=FALSE 1
                conf:(0.33) < lift:(2.33)> lev:(0.04) [0] conv:(168)
   422. outlook=rainy play=yes 3 ==> temperature=cool humidity=normal windy=FALSE 1
                conf:(0.33) < lift:(2.33)> lev:(0.04) [0] conv:(168)
   622. windy=TRUE 6 ==> outlook=sunny temperature=mild humidity=normal 1
                conf:(0.17) < lift:(2.33)> lev:(0.04) [0] conv:(182)
   623. outlook=sunny temperature=mild humidity=normal 1 ==> windy=TRUE 1
                conf:(1) < lift:(2.33)> lev:(0.04) [0] conv:(112)
   624. outlook=sunny humidity=normal 2 ==> temperature=mild windy=TRUE 1
                conf:(0.5) < lift:(2.33)> lev:(0.04) [0] conv:(154)
  1122. outlook=overcast temperature=mild humidity=high 1 ==> play=yes 1
                conf:(1) < lift:(1.56)> lev:(0.03) [0] conv:(70)
  1123. play=yes 9 ==> outlook=overcast temperature=hot humidity=normal 1
                conf:(0.11) < lift:(1.56)> lev:(0.03) [0] conv:(182)
  1124. outlook=overcast temperature=hot humidity=normal 1 ==> play=yes 1
                conf:(1) < lift:(1.56)> lev:(0.03) [0] conv:(70)
  1997. windy=FALSE play=yes 6 ==> outlook=sunny 1
                conf:(0.17) < lift:(0.47)> lev:(-0.08) [-1] conv:(126)
  1998. outlook=sunny 5 ==> windy=FALSE play=yes 1
                conf:(0.2) < lift:(0.47)> lev:(-0.08) [-1] conv:(112)
  2000. humidity=normal 7 ==> play=no 1
                conf:(0.14) < lift:(0.4)> lev:(-0.11) [-1] conv:(126)
```

Figura 11: Algumas Regras Geradas pelo *WEKA* Utilizando o Conjunto de Dados *Weather-Nominal*

```
[R0001],humidity=normal & play=no,outlook=rainy & temperature=cool & windy=TRUE,
                                                        [0.071429,0.000000,0.928571,0.000000,14]
[R0002],temperature=cool & play=no,outlook=rainy & humidity=normal & windy=TRUE,
                                                        [0.071429,0.000000,0.928571,0.000000,14]
[R0003],outlook=rainy & humidity=normal & windy=TRUE,temperature=cool & play=no,
                                                        [0.071429,0.000000,0.928571,0.000000,14]
[R0420],humidity=normal & play=yes,outlook=rainy & temperature=cool & windy=FALSE,
                                                        [0.071429,0.357143,0.571326,0.000102,14]
[R0421],temperature=cool & play=yes,outlook=rainy & humidity=normal & windy=FALSE,
                                                        [0.071429,0.142857,0.714081,0.071633,14]
[R0422],outlook=rainy & play=yes,temperature=cool & humidity=normal & windy=FALSE,
                                                        [0.071429,0.142857,0.714081,0.071633,14]
[R0622],windy=TRUE,outlook=sunny & temperature=mild & humidity=normal,
                                                        [0.071429,0.357143,0.571326,0.000102,14]
[R0623],outlook=sunny & temperature=mild & humidity=normal,windy=TRUE,
                                                        [0.071429,0.000000,0.570815,0.357756,14]
[R0624],outlook=sunny & humidity=normal,temperature=mild & windy=TRUE,
                                                        [0.071429,0.071429,0.713979,0.143164,14]
[R1122],outlook=overcast & temperature=mild & humidity=high,play=yes,
                                                        [0.071429,0.000000,0.358974,0.569597,14]
[R1123],play=yes,outlook=overcast & temperature=hot & humidity=normal,
                                                        [0.071429,0.571429,0.357346,-0.000204,14]
[R1124],outlook=overcast & temperature=hot & humidity=normal,play=yes,
                                                        [0.071429,0.000000,0.358974,0.569597,14]
[R1997],windy=FALSE & play=yes,outlook=sunny,[0.071429,0.357143,0.288247,0.283181,14]
[R1998],outlook=sunny,windy=FALSE & play=yes,[0.071429,0.285714,0.288754,0.354103,14]
[R2000],humidity=normal,play=no,[0.071429,0.428571,0.214286,0.285714,14]
```

Figura 12: Regras da Figura 11 Convertidas para o Formato Padrão Compacto de Regras de Associação

```
1 1 21.4286 60    35.71 1.68  outlook=sunny  play=no
1 1 28.5714 100   64.29 1.5556  outlook=overcast  play=yes
2 1 14.2857 66.67 64.29 1.037  outlook=rainy and temperature=mild  play=yes
2 1 14.2857 50    35.71 1.4  temperature=cool and humidity=normal  outlook=rainy
1 2 14.2857 50    21.43 2.3333  temperature=cool  outlook=rainy and humidity=normal
1 2 21.4286 60    42.86 1.4  outlook=rainy  windy=FALSE and play=yes
2 2  7.1429 50    21.43 2.3333  temperature=mild and humidity=normal  outlook=rainy and windy=FALSE
3 1  7.1429 100   64.29 1.5556  outlook=rainy and temperature=mild and humidity=normal  play=yes
2 2  7.1429 50    21.43 2.3333  temperature=mild and humidity=normal  outlook=rainy and play=yes
2 2  7.1429 50    14.29 3.5  outlook=rainy and humidity=high  temperature=mild and play=no
3 1  7.1429 50    35.71 1.4  outlook=rainy and temperature=cool and humidity=normal  play=no
1 3 14.2857 50    21.43 2.3333  temperature=hot  outlook=sunny and humidity=high and play=no
1 3 14.2857 50    14.29 3.5  outlook=overcast  temperature=hot and windy=FALSE and play=yes
4 1  7.1429 100   64.29 1.5556  outlook=rainy and temperature=mild and humidity=normal and windy=FALSE
                                                                                          play=yes
```

Figura 13: Algumas Regras Geradas pelo MineSet^{TM} Utilizando o Conjunto de Dados *Weather-Nominal*

```
[R0001],outlook=sunny,play=no,[0.214286,0.142857,0.500043,0.142814,14]
[R0002],outlook=overcast,play=yes,[0.285714,0.000000,0.357100,0.357186,14]
[R0003],outlook=rainy and temperature=mild,play=yes,[0.142857,0.071418,0.285682,0.500043,14]
[R0004],temperature=cool and humidity=normal,outlook=rainy,[0.142857,0.142857,0.500043,0.214243,14]
[R0005],temperature=cool,outlook=rainy and humidity=normal,[0.142857,0.142857,0.642843,0.071443,14]
[R0006],outlook=rainy,windy=FALSE and play=yes,[0.214286,0.142857,0.428543,0.214314,14]
[R0007],temperature=mild and humidity=normal,outlook=rainy and windy=FALSE,
                                                        [0.071429,0.071429,0.714271,0.142871,14]
[R0008],outlook=rainy and temperature=mild and humidity=normal,play=yes,
                                                        [0.071429,0.000000,0.357100,0.571471,14]
[R0009],temperature=mild and humidity=normal,outlook=rainy and play=yes,
                                                        [0.071429,0.071429,0.714271,0.142871,14]
[R0010],outlook=rainy and humidity=high,temperature=mild and play=no,
                                                        [0.071429,0.071429,0.785671,0.071471,14]
[R0011],outlook=rainy and temperature=cool and humidity=normal,play=no,
                                                        [0.071429,0.071429,0.571471,0.285671,14]
[R0012],temperature=hot,outlook=sunny and humidity=high and play=no,
                                                        [0.142857,0.142857,0.642843,0.071443,14]
[R0013],outlook=overcast,temperature=hot and windy=FALSE and play=yes,
                                                        [0.142857,0.142857,0.714243,0.000043,14]
[R0014],outlook=rainy and temperature=mild and humidity=normal and windy=FALSE,play=yes,
                                                        [0.071429,0.000000,0.357100,0.571471,14]
```

Figura 14: Regras da Figura 13 Convertidas para o Formato Padrão Compacto de Regras de Associação.

```
[R0001] IF      outlook=sunny
                        THEN play=no
                        [0.214286,0.142857,0.500043,0.142814,14]

[R0002] IF      outlook=overcast
                        THEN play=yes
                        [0.285714,0.000000,0.357100,0.357186,14]

[R0003] IF      outlook=rainy and temperature=mild
                        THEN play=yes
                        [0.142857,0.071418,0.285682,0.500043,14]

[R0004] IF      temperature=cool and humidity=normal
                        THEN outlook=rainy
                        [0.142857,0.142857,0.500043,0.214243,14]

[R0005] IF      temperature=cool
                        THEN outlook=rainy and humidity=normal
                        [0.142857,0.142857,0.642843,0.071443,14]

[R0006] IF      outlook=rainy
                        THEN windy=FALSE and play=yes
                        [0.214286,0.142857,0.428543,0.214314,14]

[R0007] IF      temperature=mild and humidity=normal
                        THEN outlook=rainy and windy=FALSE
                        [0.071429,0.071429,0.714271,0.142871,14]

[R0008] IF      outlook=rainy and temperature=mild and humidity=normal
                        THEN play=yes
                        [0.071429,0.000000,0.357100,0.571471,14]

[R0009] IF      temperature=mild and humidity=normal
                        THEN outlook=rainy and play=yes
                        [0.071429,0.071429,0.714271,0.142871,14]

[R0010] IF      outlook=rainy and humidity=high
                        THEN temperature=mild and play=no
                        [0.071429,0.071429,0.785671,0.071471,14]

[R0011] IF      outlook=rainy and temperature=cool and humidity=normal
                        THEN play=no
                        [0.071429,0.071429,0.571471,0.285671,14]

[R0012] IF      temperature=hot
                        THEN outlook=sunny and humidity=high and play=no
                        [0.142857,0.142857,0.642843,0.071443,14]

[R0013] IF      outlook=overcast
                        THEN temperature=hot and windy=FALSE and play=yes
                        [0.142857,0.142857,0.714243,0.000043,14]

[R0014] IF      outlook=rainy and temperature=mild and humidity=normal and windy=FALSE
                        THEN play=yes
                        [0.071429,0.000000,0.357100,0.571471,14]
```

Figura 15: Regras da Figura 13 Convertidas para o Formato Padrão de Regras de Associação.

5 Considerações Finais

Regras de Associação é uma técnica em grande destaque nos últimos anos na área de Mineração de Dados, apesar de ainda pouco explorada se comparada com outras técnicas de Aprendizado de Máquina e de Mineração de Dados, visto que a maioria trata de problemas de classificação e regressão.

Nesse sentido, este relatório técnico apresenta a definição de uma sintaxe padrão para representar Regras de Associação, e a implementação de uma classe para converter as saídas dos algoritmos de associação *Apriori*, MineSet^{TM}, *WEKA* e *Magnum Opus* para essa sintaxe.

Uma sintaxe padrão para representar Regras de Associação é importante pois permite a implementação de conversores de diferentes algoritmos para um formato único. Essa unificação viabiliza a implementação diversas medidas de pós-processamento uma única vez e a utilização destas com regras geradas pelos diferentes algoritmos.

Considerado o tamanho dos arquivos de regras possíveis de serem gerados e, os requisitos de compreensibilidade do conhecimento pelo usuário e de eficiência no processamento dos arquivos de regras, foi necessária a adoção de duas sintaxes: uma padrão – com elevada compreensibilidade; outra compacta – com arquivos cerca de 30% menores.

Por fim, com a definição dessas sintaxes será possível incrementar a potencialidade de ambientes para trabalhar problemas de associação.

A Apêndice

Neste apêndice é apresentado o código fonte, em Perl, do módulo que permite converter regras geradas pelos aplicativos *Apriori*, MineSet^{TM}, *WEKA* e *Magnum Opus* para o Formato Padrão e Formato Padrão Compacto de Regras de Associação, definidos neste relatório técnico.

```
#-----------------------------------------------------------------------------
#Classe de transformação dos formatos Apriori, MineSet, Magnum Opus e Weka para os
    formatos Padrão e Padrão cOMPACTO.
#
#Desenvolvido por Gilson Kenji Ywamoto e Edson Augusto Melanda - 2002/2003
#-----------------------------------------------------------------------------


package Rass_Conv;


require Exporter;
@ISA = qw(Exporter);
@EXPORT = qw(Apriori_to_Padrao, Mineset_to_Padrao, Magnumopus_to_Padrao, Weka_to_Padrao);

sub new
{
    my $class = shift;
    my $self = {};
    bless $self, $class;
    return $self;
}


#-----------------------------------------------------------------------------
# Conversão do formato Apriori para o Padrão
#
#Parâmetros de Entrada:
#-->$nomearq: indica o nome do arquivo de entrada, com a extensão '.apr'
#-->$opcao:   0 se o usuário deseja que o formato de saída seja o Padrão Compacto.
#          1 se o usuário deseja que o formato de saída seja o Padrão.
#-----------------------------------------------------------------------------

sub Apriori_to_Padrao
{
    my($self, $nomearq, $opcao) = @_;

    my($input_file, $output_file, @linhas, $total_regras, $id_regra);
    my($regra, @itens, @lados, @medidas);
    my(@suporte_vet,@temp);
    my($suporte, $confianca, $conf_esp);
    my($n_RHS, $n_notRHS, $n_LHS, $n_notLHS, $n_LHS_RHS, $n_notLHS_RHS, $n_LHS_notRHS,
        $n_notLHS_notRHS);

    $input_file = $nomearq;

    if($opcao == 0) #PADRÃO COMPACTO
    {
        $output_file = $nomearq.'.dcar';
    }
    else #PADRÃO
    {
        $output_file = $nomearq.'.dar';
    }

    open(IN, $input_file) || die "Could not open the input file $input_file\n";
    open(OUT, ">$output_file");
```

```
@linhas = <IN>; #@LINHAS POSSUI O CONTEÚDO DO ARQUIVO DE ENTRADA
close(IN);

$total_regras = 0;

$id_regra = 0; #indica qual regra está sendo trabalhada

foreach $regra (@linhas)
{

    @itens = split('  \(',$regra);

    @lados = split(' <- ',$itens[0]); #$LADOS[0] = RHS, $LADOS[1] = LHS

    $lados[0] =~ s/ \<-//; #RETIRA ' <- ' NOS CASOS ONDE NÃO HÁ RHS

    if ($lados[1] eq ""){
        $lados[1] = 'TRUE';}


        $itens[1] =~ s/\)//; #RETIRA )

    $itens[1] =~ s/%//;

    @medidas = split(', ',$itens[1]);

    @suporte_vet = split('\/',$medidas[0]);


    #MEDIDAS[0] = SUPORTE, MEDIDAS[1] = CONFIANCA, MEDIDAS[2] = DIF. CONF.


    $suporte = $suporte_vet[0]/100; #SUPORTE
    $confianca = $medidas[1]/100;    #CONFIANCA
    $conf_esp = $confianca - ($medidas[2]/100);  #CONFIANCA ESPERADA

    if ($total_regras == 0) #calcula o número total de exemplos
    {

        $total_regras = $suporte_vet[1]/$suporte;

        @temp = split('\.',$total_regras);

        $temp[1] = '0.'.$temp[1];

        if($temp[1]>0.5){#arredondando para cima (ex: 13.95 se tornará 14.95)
            $total_regras++;}
    }


    #valores entre 0 e 1
    $n_RHS = $conf_esp;  #N(RHS)
    $n_notRHS = 1 - $n_RHS; #N(notRHS)
    $n_LHS = ($suporte/$confianca); #N(LHS)
    $n_notLHS = 1 - $n_LHS; #N(notLHS)

    $n_LHS_RHS = $suporte; #N(LHS RHS)

    $n_notLHS_RHS = $n_RHS - $n_LHS_RHS; #N(notLHS RHS)
    $n_LHS_notRHS = $n_LHS - $n_LHS_RHS; #N(LHS notRHS)

    $n_notLHS_notRHS = $n_notRHS - $n_LHS_notRHS; #N(notLHS notRHS)

    $id_regra++;
```

```
                #formatando a saida de $id_regra
                if ($id_regra<10) {
                        print OUT "[R000$id_regra]"; }
                else {
                        if ($id_regra<100) {
                                print OUT "[R00$id_regra]"; }
                        else {
                                if ($id_regra<1000) {
                                        print OUT "[R0$id_regra]"; }
                                else{
                                        print OUT "[R$id_regra]"; }
                        }
                }


        if ($opcao == 0)#PADRÃO COMPACTO
        {
            $lados[1] =~ s/ / & /g;

            print OUT ",$lados[1],$lados[0],";


            printf OUT "[%0.6f,%0.6f,%0.6f,%0.6f,%d]\n",$n_LHS_RHS,$n_LHS_notRHS,
                $n_notLHS_notRHS,$n_notLHS_RHS,$total_regras;
        }
        else #PADRÃO
        {



            $lados[1] =~ s/ /\n\t\t\tAND /g;


            print OUT "\tIF\t";

            print OUT "$lados[1]\n\t\t\tTHEN_$lados[0]\n";

            printf OUT "\t\t\t[%0.6f,%0.6f,%0.6f,%0.6f,%d]\n\n",$n_LHS_RHS,$n_LHS_notRHS,
                $n_notLHS_notRHS,$n_notLHS_RHS,$total_regras;
        }

    }
    close(OUT);
}

#-----------------------------------------------------------------------------------------
# Conversão do formato MineSet para o Padrão
#
#Parâmetros de Entrada:
#-->$nomearq:    indica o nome do arquivo de entrada, com a extensão '.rules.out'
#-->$num_ex:     indica o número de exemplos (opcional).
#-->$arq_dados: indica o nome do arquivo que possui os dados (opcional).
#-->$opcao:      0 se o usuário deseja que o formato de saída seja o Padrão Compacto.
#            1 se o usuário deseja que o formato de saída seja o Padrão.
#-----------------------------------------------------------------------------------------

sub Mineset_to_Padrao
{
    my($self, $nomearq, $num_ex, $arq_dados , $opcao) = @_;

    my(@dados, $dado, $data_file);
    my($input_file, $output_file, @linhas, $total_regras, $id_regra);
    my($regra, @itens, @lados, @medidas);
    my($suporte, $confianca, $conf_esp);
    my($LHS, $RHS);
    my(@VET_LHS, $vetor, $i, @VET_RHS, @vetor_itens);
```

```
my($n_RHS, $n_notRHS, $n_LHS, $n_notLHS, $n_LHS_RHS, $n_notLHS_RHS, $n_LHS_notRHS,
    $n_notLHS_notRHS);


#calculando o número de exemplos
if ($num_ex > 0) #numero de exemplos passado com parametro
{
    $total_regras = $num_ex;
}
elsif(open(DADOS, $arq_dados))  #nome do arquivo com os exemplos
{
    @dados = <DADOS>;
    $total_regras = $#dados+1;
    close (DADOS);
}
else #tentará abrir o arquivo "arquivo.data" ou "arquivo-assoc.out"
{
    $data_file = $nomearq;
    $data_file =~ s/\.rules//;

    if (open(DADOS, $data_file))#arquivo "arquivo.data"
    {
        @dados = <DADOS>;
        $total_regras = $#dados+1;
        close (DADOS);
    }
    else #arquivo "arquivo-assoc.out"
    {

        $data_file = $nomearq;
        $data_file =~ s/\.rules\.data/\-assoc\.out/;
        open(DADOS, $data_file) || die "could not open the file $data_file";

        @dados = <DADOS>;
        close (DADOS);

        foreach $dado (@dados)
        {
            if($dado =~ / records read/)
            {
                @vetor = split(" ",$dado);
                $total_regras = $vetor[0];
            }
        }
    }

}


$input_file = $nomearq;

if ($opcao == 0) #PADRÃO COMPACTO
{
    $output_file = $nomearq.'.dcar';
}
else #PADRÃO
{
    $output_file = $nomearq.'.dar';
}

open(IN, $input_file) || die "Could not open the input file $input_file\n";
open(OUT, ">$output_file");

@linhas = <IN>; #@LINHAS POSSUI O CONTEÚDO DO ARQUIVO DE ENTRADA
close(IN);
```

```
$id_regra = 0; #indica qual regra está sendo trabalhada


foreach $regra (@linhas)
{

    @itens = split(';',$regra);

    #$itens[2] =  suporte, $itens[3] = confianca, $itens[4] = confianca esperada,
        $itens[5] = lift
    #$itens[6] = LHS, $itens[7] = RHS

    $LHS = $itens[6];
    $RHS = $itens[7];

    $RHS =~ s/\n//;


    #formatando o $LHS e o $RHS
    $i=-1;
    @VET_LHS = split(' and ',$LHS);


    foreach $vetor (@VET_LHS) #transformará os intervalos com "(" e "]" em ">" e "<="
    {
        $i++;
        if($vetor =~ / in /)
        {
            if($vetor =~ s/ in \(\.\.\. / \<= /g)
            # "AAA in (... YYY]" se transformará em "AAA <= YYY"
            {
                $vetor =~ s/]//;
            }
            else #(XXX ...
            {
                if($vetor =~ s/ \.\.\.]//)
                # "AAA in (XXX ...]" se transformará em "AAA > XXX"
                {
                    $vetor =~ s/ in \(/ \> /;
                }
                else
                # "AAA in (XXX ... YYY]" se transformará em "AAA > XXX and AAA <= YYY
                    "
                {
                    $vetor =~ s/in \(//;
                    $vetor =~ s/]//;
                    $vetor =~ s/\.\.\. //;
                    @vetor_itens = split(' ',$vetor);
                    $vetor = $vetor_itens[0]. ' > '.$vetor_itens[1]. ' and '.
                        $vetor_itens[0]. ' <= '.$vetor_itens[2];
                }
            }
        }

        if($i<$#VET_LHS)
        #insere '&' entre os dados de cada posicao de VET_LHS, mas não adiciona '&' à
            ultima posicao
        {
            $vetor = $vetor. ' &';
        }

    }


    $i=-1;
    @VET_RHS = split(' and ',$RHS);
    foreach $vetor (@VET_RHS) #transformará os intervalos com "(" e "]" em ">" e "<="
```

```
{
    $i++;
    if($vetor =~ / in /)
    {
        if($vetor =~ s/ in \(\.\.\. / \<= /g)
        # "AAA in (... YYY]" se transformará em "AAA <= YYY"
        {
            $vetor =~ s/]//;
        }
        else #(XXX ...
        {
            if($vetor =~ s/ \.\.\.]//)
            # "AAA in (XXX ...]" se transformará em "AAA > XXX"
            {
                $vetor =~ s/ in \(/ \> /;
            }
            else
            # "AAA in (XXX ... YYY]" se transformará em "AAA > XXX and AAA <= YYY
                "
            {
                $vetor =~ s/in \(//;
                $vetor =~ s/]//;
                $vetor =~ s/\.\.\. //;
                @vetor_itens = split(' ',$vetor);
                $vetor = $vetor_itens[0]. ' > '. $vetor_itens[1]. ' and '.
                    $vetor_itens[0]. ' <= '. $vetor_itens[2];
            }
        }
    }
    if($i<$#VET_RHS)
    #insere '&' entre os dados de cada posicao de VET_RHS, mas não adiciona '&' à
        ultima posicao
    {
        $vetor = $vetor. ' &';
    }
}


$suporte = $itens[2]/100;    #SUPORTE
$confianca = $itens[3]/100; #CONFIANCA
$conf_esp = $itens[4]/100;  #CONFIANCA ESPERADA

#medidas entre 0 e 1
$n_RHS = $conf_esp; #N(RHS)
$n_notRHS = 1 - $n_RHS; #N(notRHS)
$n_LHS = ($suporte/$confianca); #N(LHS)
$n_notLHS = 1 - $n_LHS; #N(notLHS)
$n_LHS_RHS = $suporte;  #N(LHS RHS)
$n_notLHS_RHS = $n_RHS - $n_LHS_RHS; #N(notLHS RHS)
$n_LHS_notRHS = $n_LHS - $n_LHS_RHS; #N(LHS notRHS)
$n_notLHS_notRHS = $n_notRHS - $n_LHS_notRHS; #N(notLHS notRHS)

$id_regra++;


        #formatando a saida de $id_regra
        if ($id_regra<10) {
            print OUT "[R000$id_regra]"; }
        else {
            if ($id_regra<100) {
                    print OUT "[R00$id_regra]"; }
                else {
                    if ($id_regra<1000) {
                            print OUT "[R0$id_regra]"; }
                        else{
                                print OUT "[R$id_regra]"; }
                }
```

```
        }

if ($opcao == 0) #PADRÃO COMPACTO
{

    if ($LHS =~ / in \(/) #se houve transformacao de '(' ou ']' em '>' ou '<='
    {
        for ($i=0; $i<=$#VET_LHS ; $i++)
        {
            $VET_LHS[$i] =~ s/ and / \& /g;
        }
        print OUT ",@VET_LHS,";
    }
    else
    {
        $LHS =~ s/ and / & /g;
        print OUT ",$LHS,";
    }

    if ($RHS =~ / in \(/) #se houve transformacao de '(' ou ']' em '>' ou '<='
    {
        for ($i=0; $i<=$#VET_RHS ; $i++)
        {
            $VET_RHS[$i] =~ s/ and / & /g;
        }
        print OUT "@VET_RHS,";
    }
    else
    {
        $RHS =~ s/ and / & /g;
        print OUT "$RHS,";
    }


    printf OUT "[%0.6f,%0.6f,%0.6f,%0.6f",$n_LHS_RHS,$n_LHS_notRHS,
        $n_notLHS_notRHS,$n_notLHS_RHS;
    print OUT ",$total_regras]\n";

}
else #PADRÃO
{

    $LHS =~ s/ and/\n\t\t\tAND/g;
    $RHS =~ s/ and/\n\t\t\tAND/g;


    print OUT "\tIF\t";

    if ($LHS =~ / in \(/) #se houve transformacao de '(' ou ']' em '>' ou '<='
    {
        for ($i=0; $i<$#VET_LHS ; $i++)
        {
            $VET_LHS[$i] =~ s/\&//g;
            $VET_LHS[$i] =~ s/ and / AND /g;
            print OUT "$VET_LHS[$i]\n\t\t\tAND ";
        }
        $VET_LHS[$i] =~ s/\&//g;
        $VET_LHS[$i] =~ s/ and / AND /g;
        print OUT "$VET_LHS[$i]\n\t\t\tTHEN ";
    }
    else
    {
        print OUT "$LHS\n\t\t\tTHEN ";
    }
```

```
            if ($RHS =~ / in \(/) #se houve transformacao de '(' ou ']' em '>' ou '<='
            {
                for ($i=0; $i<$#VET_RHS ; $i++)
                {
                    $VET_RHS[$i] =~ s/\&//g;
                    $VET_RHS[$i] =~ s/ and / AND /g;
                    print OUT "$VET_RHS[$i]\n\t\t\tAND ";
                }
                $VET_RHS[$i] =~ s/\&//g;
                $VET_RHS[$i] =~ s/ and / AND /g;
                print OUT "$VET_RHS[$i]\n";
            }
            else
            {
                print OUT "$RHS\n";
            }



            printf OUT "\t\t\t[%0.6f,%0.6f,%0.6f,%0.6f",$n_LHS_RHS,$n_LHS_notRHS,
                $n_notLHS_notRHS,$n_notLHS_RHS;
            print OUT ",$total_regras]\n\n";

        }
    }
    close(OUT);
}

#-----------------------------------------------------------------------------------------
# Conversão do formato Magnum Opus para o Padrão
#
#Parâmetros de Entrada:
#-->$nomearq:    indica o nome do arquivo de entrada, com a extensão '.mop'
#-->$opcao:      0 se o usuário deseja que o formato de saída seja o Padrão compacto.
#               1 se o usuário deseja que o formato de saída seja o Padrão.
#-----------------------------------------------------------------------------------------

sub Magnumopus_to_Padrao
{
    my($self, $nomearq, $opcao) = @_;

    my($input_file, $output_file, @linhas, $total_regras, $id_regra);
    my($regra, @itens, @medidas);
    my($suporte, $confianca, $conf_esp);
    my($n_RHS, $n_notRHS, $n_LHS, $n_notLHS, $n_LHS_RHS, $n_notLHS_RHS, $n_LHS_notRHS,
        $n_notLHS_notRHS);
    my(@temp);

    $input_file = $nomearq;

    if($opcao == 0) #PADRÃO COMPACTO
    {
        $output_file = $nomearq.'.dcar';
    }
    else #PADRÃO
    {
        $output_file = $nomearq.'.dar';
    }

    open(IN, $input_file) || die "Could not open the input file $input_file\n";
    open(OUT, ">$output_file");

    @linhas = <IN>; #@LINHAS POSSUI O CONTEÚDO DO ARQUIVO DE ENTRADA
    close(IN);

    $id_regra = -1; #indica qual regra está sendo trabalhada
```

```
$total_regras = 0;

foreach $regra (@linhas)
{

    $id_regra++;

    if($id_regra > 0) #ignora a primeira linha
    {

        @itens = split(',',$regra);
        #$itens[0] = LHS, $itens[1] = RHS, $itens[2] = cobertura, $itens[3] = numero
            de regras que possuem o LHS
        #$itens[4] = suporte, $itens[6] = confiança, $itens[7] = lift

        if ($total_regras == 0) #calcula o número total de exemplos
        {
            $total_regras = $itens[3]/$itens[2];
            @temp = split('\.',$total_regras);
            $temp[1] = '0.'.$temp[1];
            if ($temp[1] > 0.5){ #arredondando para cima (ex: 13.95 se tornará 14.95)
                $total_regras++;
            }
        }

        #MEDIDAS[0] = SUPORTE, MEDIDAS[1] = CONFIANCA, MEDIDAS[2] = DIF. CONF.


        $suporte = $itens[4];    #SUPORTE
        $confianca = $itens[6]; #CONFIANCA
        $conf_esp = $confianca/$itens[7];  #CONFIANCA ESPERADA

        #valores entre 0 e 1
        $n_RHS = $conf_esp;  #N(RHS)
        $n_notRHS = 1 - $n_RHS; #N(notRHS)
        $n_LHS = ($suporte/$confianca); #N(LHS)
        $n_notLHS = 1 - $n_LHS; #N(notLHS)

        $n_LHS_RHS = $suporte; #N(LHS RHS)

        $n_notLHS_RHS = $n_RHS - $n_LHS_RHS; #N(notLHS RHS)
        $n_LHS_notRHS = $n_LHS - $n_LHS_RHS; #N(LHS notRHS)

        $n_notLHS_notRHS = $n_notRHS - $n_LHS_notRHS; #N(notLHS notRHS)


                #formatando a saida de $id_regra
                if ($id_regra <10) {
                        print OUT "[R000$id_regra]"; }
                else {
                        if ($id_regra <100) {
                                print OUT "[R00$id_regra]"; }
                        else {
                                if ($id_regra <1000) {
                                        print OUT "[R0$id_regra]"; }
                                else{
                                        print OUT "[R$id_regra]"; }
                        }
                }


        if ($opcao == 0)#PADRÃO COMPACTO
        {

            print OUT ",$itens[0],$itens[1],";
```

```
                printf OUT "[%0.6f,%0.6f,%0.6f,%0.6f,%d]\n",$n_LHS_RHS,$n_LHS_notRHS,
                    $n_notLHS_notRHS,$n_notLHS_RHS,$total_regras;
            }
            else #PADRÃO
            {

                $itens[0] =~ s/ & /\n\t\t\tAND /g;
                $itens[1] =~ s/ & /\n\t\t\tAND /g;


                print OUT "\tIF\t";

                print OUT "$itens[0]\n\t\t\tTHEN $itens[1]\n";

                printf OUT "\t\t\t[%0.6f,%0.6f,%0.6f,%0.6f,%d]\n\n",$n_LHS_RHS,
                    $n_LHS_notRHS,$n_notLHS_notRHS,$n_notLHS_RHS,$total_regras;
            }
        }

    }
    close(OUT);
}


#-----------------------------------------------------------------------------------------
# Conversão do formato WEKA para o Padrão
#
#Parâmetros de Entrada:
#-->$nomearq:    indica o nome do arquivo de entrada, com a extensão '.weka'
#-->$opcao:      0 se o usuário deseja que o formato de saída seja o Padrão compacto.
#            1 se o usuário deseja que o formato de saída seja o Padrão.
#-----------------------------------------------------------------------------------------

sub Weka_to_Padrao
{
    my($self, $nomearq, $opcao) = @_;

    my($input_file, $output_file, @linhas, $total_regras, $id_regra);
    my($regra, @itens, @lados, @medidas);
    my($suporte, $confianca, $conf_esp);
    my($n_RHS, $n_notRHS, $n_LHS, $n_notLHS, $n_LHS_RHS, $n_notLHS_RHS, $n_LHS_notRHS,
        $n_notLHS_notRHS);
    my($num_RHS, $num_LHS, @LHS, @RHS);


    $input_file = $nomearq;

    if($opcao == 0) #PADRÃO COMPACTO
    {
        $output_file = $nomearq.'.dcar';
    }
    else #PADRÃO
    {
        $output_file = $nomearq.'.dar';
    }

    open(IN, $input_file) || die "Could not open the input file $input_file\n";
    open(OUT, ">$output_file");

    @linhas = <IN>; #@LINHAS POSSUI O CONTEÚDO DO ARQUIVO DE ENTRADA
    close(IN);

    $id_regra = 0; #indica qual regra está sendo trabalhada

    foreach $regra (@linhas)
```

```
{
    if ($regra =~ s/^Instances://) #calcula o número de regras
    {
        $total_regras = $regra;
        $total_regras =~ s/\n//g;
        $total_regras =~ s/ //g;
    }
    elsif ($regra =~ /\=\=\>/)
    {
        @itens = split('    ',$regra); #$ITENS[0] = REGRAS,  $ITENS[1] = MEDIDAS

        @lados = split(' \=\=\> ',$itens[0]); #$LADOS[0] = LHS, $LADOS[1] = RHS

        @LHS = split(' ',$lados[0]);
        $num_LHS = pop(@LHS);

        @RHS = split(' ',$lados[1]);
        $num_RHS = pop(@RHS);

        $itens[1] =~ s/\< //g;
        $itens[1] =~ s/\>//g;
        $itens[1] =~ s/\(//g;
        $itens[1] =~ s/\)//g;

        @medidas = split(' ',$itens[1]); #$MEDIDAS[1] = LIFT

        $medidas[1] =~ s/lift://g;

        $suporte = $num_RHS/$total_regras;
        $confianca = $num_RHS/$num_LHS;
        $conf_esp = $confianca/$medidas[1];

        #valores entre 0 e 1
        $n_RHS = $conf_esp;  #N(RHS)
        $n_notRHS = 1 - $n_RHS; #N(notRHS)
        $n_LHS = ($suporte/$confianca); #N(LHS)
        $n_notLHS = 1 - $n_LHS; #N(notLHS)

        $n_LHS_RHS = $suporte; #N(LHS RHS)

        $n_notLHS_RHS = $n_RHS - $n_LHS_RHS; #N(notLHS RHS)
        $n_LHS_notRHS = $n_LHS - $n_LHS_RHS; #N(LHS notRHS)

        $n_notLHS_notRHS = $n_notLHS - $n_notLHS_RHS; #N(notLHS notRHS)

                $LHS[0] =~ s/\.//;
                $id_regra = $LHS[0];


            #formatando a saida de $id_regra
            if ($id_regra<10) {
                    print OUT "[R000$id_regra]"; }
            else {
                    if ($id_regra<100) {
                            print OUT "[R00$id_regra]"; }
                    else {
                            if ($id_regra<1000) {
                                    print OUT "[R0$id_regra]"; }
                            else{
                                    print OUT "[R$id_regra]"; }
                    }
            }


        if ($opcao == 0)#PADRÃO COMPACTO
        {
```

```
                print OUT ",";

                for($i=1; $i<$#LHS ; $i++)
                {
                    print OUT "$LHS[$i] & ";
                }
                print OUT "$LHS[$i],";

                for($i=0; $i<$#RHS ; $i++)
                {
                    print OUT "$RHS[$i] & ";
                }
                print OUT "$RHS[$i],";

                printf OUT "[%0.6f,%0.6f,%0.6f,%0.6f",$n_LHS_RHS,$n_LHS_notRHS,
                    $n_notLHS_notRHS,$n_notLHS_RHS;
                print OUT ",$total_regras]\n";
            }
            else #PADRÃO
            {

                print OUT "\tIF\t";

                for($i=1; $i<$#LHS ; $i++)
                {
                    print OUT "$LHS[$i]\n\t\t\tAND ";
                }
                print OUT "$LHS[$i]\n\t\t\tTHEN ";

                for($i=0; $i<$#RHS ; $i++)
                {
                    print OUT "$RHS[$i]\n\t\t\tAND ";
                }
                print OUT "$RHS[$i]\n";

                printf OUT "\t\t\t[%0.6f,%0.6f,%0.6f,%0.6f",$n_LHS_RHS,$n_LHS_notRHS,
                    $n_notLHS_notRHS,$n_notLHS_RHS;
                print OUT ",$total_regras]\n\n";
            }
        }


    }
    close(OUT);
}


1; # End of package

#------------------------------------------------------------------------------
```

Referências

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

Agrawal, R., T. Imielinski, & A. N. Swami (1993, 26–28). Mining association rules between sets of items in large databases. In P. Buneman & S. Jajodia (Eds.), *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Washington, D.C., pp. 207–216.

Agrawal, R. & R. Srikant (1994, 12–15). Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, & C. Zaniolo (Eds.), *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pp. 487–499. Morgan Kaufmann.

Baesens, B., S. Viaene, & J. Vanthienen (2000, August). Post-processing of association rules. In *Proceedings of the special workshop on post-processing, The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'2000)*, Boston, MA, USA, pp. 2–8.

Clementini, E., P. D. Felice, & K. Koperski (2000, August). Mining multiple-level spatial association rules for objects with a broad boundary. *Data & Knowledge Engineering 34*, 251–270.

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.

Semenova, T., M. Hegland, W. Graco, & G. Williams (2001, November). Effectiveness of mining association rules for identifying trends in large health databases. In F. J. Kurfess & M. Hilario (Eds.), *Workshop on Integrating Data Mining and Knowledge Management, The 2001 IEEE International Conference on Data Mining - ICDM'01*, San Jose, California, USA, pp. 12. acessado em 19/11/2001.

Webb, G. I. (1995). OPUS: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research 3*, 431–465.

Witten, I. H. & E. Frank (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. São Francisco, Califórnia: Morgan Kaufmann. Disponível em: `http://www.cs.waikato.ac.nz/~ml/weka/book.html`.