

# O Algoritmo de Associação Frequent Pattern-Growth na Shell Orion Data Mining Engine

Júlio César Borba Nandi<sup>1</sup>, Ruano Marques Pereira<sup>1</sup>, Gabriel Felipe<sup>1</sup>, Kristian Madeira<sup>1</sup>, Ana Claudia Garcia Barbosa<sup>1</sup>, Merisandra Côrtes de Mattos Garcia<sup>1</sup>

<sup>1</sup>Grupo de Pesquisa em Inteligência Computacional Aplicada - Curso de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC – Brasil

{julio-bor, ruanopereira, gabrielheavy}@hotmail.com,  
{kma, agb, mem}@unesc.net

**Resumo.** *O data mining automatiza o processo da extração de conhecimentos por meio de ferramentas e algoritmos computacionais. Dentre as tarefas de data mining, a descoberta de regras de associação tem recebido atenção por parte de pesquisadores tanto da área acadêmica como das organizações. Este artigo demonstra a implementação em uma ferramenta de data mining do algoritmo FP-Growth, desenvolvido para superar as limitações presentes no algoritmo Apriori e melhorar a eficiência da descoberta de regras de associação por meio de uma abordagem diferente, sem a geração do conjunto de candidatos.*

**Palavras-chave:** *Data Mining, Associação, Algoritmo FP-Growth, FP-tree .*

**Abstract.** *The data mining automates the process of extracting knowledge through computational tools and algorithms. Among the tasks of data mining, the discovery of association rules has received attention from researchers in academia as both organizations. This article demonstrates the implementation into a tool for data mining FP-Growth algorithm, developed to overcome the limitations present in Apriori algorithm and improve the efficiency of discovery of association rules using a different approach without generating the set of candidate.*

**Keywords:** *Data Mining, Association, FP-Growth Algorithm, FP-Tree.*

## 1. Introdução

Constantes avanços no armazenamento de dados, no conhecimento sobre as mais diversas áreas, e na disponibilidade de armazenamento com custo mais acessível, proporcionaram a coleta de grandes quantidades de dados. De fato a capacidade de reunir e armazenar os conjuntos de dados é superior a de analisar e entender os mesmos. Portanto a criação de técnicas e ferramentas computacionais é necessária para proporcionar agilidade na extração de conhecimento útil dos volumes de dados em rápida expansão [Tan, Steinbach e Kumar 2009].

Para atender esta necessidade, surgiu a área denominada Descoberta de Conhecimento em Bases de Dados, *Knowledge Discovery in Databases* (KDD), esse

processo inclui limpeza, integração, seleção e transformação de dados; data mining; avaliação de padrões e apresentação do conhecimento [Han e Kamber 2006]. Sendo o *data mining* a mais destacada etapa desse processo, pois disponibiliza técnicas e ferramentas computacionais necessárias para proporcionar agilidade na extração de novos conhecimentos e na confirmação dos já existentes, contribuindo nas tomadas de decisões [Goldschmidt e Passos 2005].

A descoberta de regras de associação tem recebido atenção por parte de pesquisadores tanto da área acadêmica como das organizações. Na área acadêmica, as pesquisas desenvolvidas estão gerando bons resultados. Esses resultados estão sendo utilizados pelas organizações em aplicações práticas [Han e Kamber 2006].

Neste artigo apresenta-se a implementação do algoritmo *FP-Growth* no módulo de associação do protótipo de uma ferramenta de *data mining*, denominada *Shell Orion*.

## 2. O Algoritmo *FP-Growth*

Algoritmos tradicionais de associação adotam uma abordagem igual ou semelhante a do algoritmo Apriori, que se baseia na seguinte regra: se qualquer padrão de comprimento  $k$  não é frequente na base de dados, seu comprimento  $(k + 1)$  não será frequente. A ideia é, por meio de um processo iterativo, gerar o conjunto de padrões de candidatos de comprimento  $(k + 1)$  a partir do conjunto de padrões de frequência de comprimento  $k$  (para  $k \geq 1$ ), e verificar suas frequências de ocorrência na base de dados. No entanto, a geração de conjunto candidato é ainda dispendiosa especialmente quando há um grande número de padrões e/ou estes são longos, ou seja, padrões formados por um número expressivo de itens. Também é dispendioso percorrer repetidas vezes a base de dados para verificar e testar todos os conjuntos de candidatos e seus padrões correspondentes [Han et al 2004].

Com o objetivo de superar essas limitações e melhorar a eficiência da descoberta de regras de associação diversos métodos e algoritmos alternativos têm sido desenvolvidos. Como por exemplo, o algoritmo *FP-Growth* que utiliza uma abordagem diferente do Apriori, sem a geração do conjunto de candidatos. Também não concorda com o paradigma de gerar e testar do Apriori, ao contrário disso, codifica o conjunto de dados em uma estrutura de dados compacta em forma de árvore chamada *Frequent Pattern tree (FP-tree)* e extrai os conjuntos de itens frequentes diretamente desta estrutura. Isso possibilita uma melhor eficiência na geração das regras de associação, pois evita constantes acessos na base de dados [Tan, Steinbach e Kumar 2009].

## 3. O Algoritmo *FP-Growth* na Tarefa de Associação da *Shell Orion Data Mining Engine*

A implementação do algoritmo *FP-Growth* teve início com a construção dos diagramas de caso de uso, atividades e sequência utilizando a linguagem de modelagem UML. Posteriormente foi desenvolvida a demonstração matemática do funcionamento do algoritmo com a finalidade de facilitar o entendimento do mesmo.

O algoritmo necessita de no mínimo dois parâmetros de entrada [Tan, Steinbach e Kumar 2009]:

- a) **suporte:** é a métrica utilizada pelo algoritmo para encontrar todos os *N itemsets*. O suporte de uma regra de associação  $X, A \Rightarrow B$ , é a porcentagem



Na construção da estrutura *FP-tree*, o algoritmo *FP-Growth* cria uma tendência. Isso ocorre devido a forma como as informações, que estão contidas nas transações que estão sendo lidas, são organizadas. Essa tendência, padrão frequente de crescimento, dá nome ao algoritmo e contribui para um melhor aproveitamento dos ramos inseridos na estrutura e a torna compacta; também evita a dispendiosa etapa da geração e teste de candidatos, pois não necessita que um *itemset* descoberto seja armazenado para consultas em iterações posteriores.

Ao término da segunda leitura da base de dados, todas as transações estarão processadas e inseridas na estrutura *FP-tree* que estará corretamente criada. Ela irá representar a base de dados original e será passada como parâmetro para outro método do algoritmo *FP-Growth* responsável pela descoberta dos  $N$  *itemsets* frequentes.

O método consiste no desenvolvimento de uma estratégia baseada na técnica de dividir para conquistar, onde o problema é fracionado em subproblemas. O algoritmo faz isso considerando cada item da tabela de cabeçalhos. O primeiro item a ser processado é o último, o segundo o penúltimo item e assim sucessivamente até o término do processamento de todos os itens. Para cada *itemset* presente na tabela de cabeçalhos, bases de padrões condicionais são geradas, as quais são utilizadas para a construção das *FP-tree* condicionais, que relacionam os caminhos frequentes que se conectam aos nós correspondentes ao *itemset* em questão. Uma vez criadas, as *FP-tree* condicionais são utilizadas para encontrar os padrões frequentes que apresentam o *itemset* como sufixo [Han, Pei e Yin 2000].

Tendo como base o conjunto de  $N$  *itemsets* frequentes encontrados por meio da *FP-tree*, o algoritmo *FP-Growth* gera as regras de associação. A partir deste momento começa a fase de pós-processamento das regras geradas, na *Shell Orion*, objetivando a seleção das regras úteis e fortes. O algoritmo *FP-Growth* irá primeiro calcular a confiança (fórmula 1):

$$conf(A \Rightarrow B) = prob(A | B) = \frac{sup(A \cup B)}{sup(A)} = \frac{n(A \cup B)}{n(A)} * 100 \quad (1)$$

Depois irá calcular respectivamente [Guillet e Hamilton 2007]:

- a) **lift**: é utilizada para avaliar dependências, ou seja, esta medida de qualidade indica o quanto mais frequente torna-se o conseqüente da regra quando o antecedente ocorre (fórmula 2);

$$Lift(A \Rightarrow B) = \frac{Conf(A \Rightarrow B)}{Sup(B)} \quad (2)$$

- b) **leverage**: esta medida é utilizada com o objetivo de se descobrir o valor da diferença entre o suporte real e o esperado de uma regra de associação (fórmula 3);

$$Leverage(A \Rightarrow B) = Sup(A \Rightarrow B) - (Sup(A) \times Sup(B)) \quad (3)$$

- c) **convicção**: tem por objetivo avaliar uma regra de associação como uma implicação, ou seja, o sentido da seta “ $\Rightarrow$ ” (fórmula 4).

$$Conv(A \Rightarrow B) = \frac{Sup(A) \times Sup(\neg B)}{Sup(A \cup \neg B)} \quad (4)$$

Somente as regras de associação que satisfizerem todas as medidas de qualidade serão consideradas relevantes e úteis.

### 3.1 Implementação

O algoritmo *FP-Growth* foi implementado no módulo de associação da *Shell Orion Data Mining Engine*, por meio da linguagem de programação Java e do ambiente de programação integrado *NetBeans 7.3.1*.

Para a execução do algoritmo é necessário selecionar os atributos da tabela que serão analisados e informar os valores mínimos para o suporte, a confiança, o *lift*, o *leverage* e a convicção (Figura 2).

FP-Growth

**Parâmetros do Algoritmo**

Suporte Mínimo: 20%

Minerar atributos no Formato Basket ☐

**Medidas de Qualidade**

Confiança Mínima: 60.00%

Lift Mínimo: 100.00

Leverage Mínimo: 0.00

Convicção Mínima: 0.90

**Atributos de Entrada**

Disponíveis

- CLASS
- HANDICAPPED\_INFANTS
- WATER\_PROJECT\_COST\_SHARING
- ADOPTION\_OF\_THE\_BUDGET\_RESOLUTION
- PHYSICIAN\_FEE\_FREEZE
- EL\_SALVADOR\_AID
- RELIGIOUS\_GROUPS\_IN\_SCHOOLS

Selecionados

Executar Cancelar Sobre

Figura 2. Seleção dos parâmetros e atributos de entrada

Os resultados obtidos pelo algoritmo *FP-Growth* na *Shell Orion*, podem ser analisados por meio de relatórios. A Figura 3 mostra um relatório textual gerado pelo algoritmo contendo os valores informados para os parâmetros de entrada; os atributos de entrada selecionados; os tempos de execução do algoritmo e o tempo total (execução do algoritmo mais montagem do relatório); número de transações presentes na base de dados; as regras de associação geradas com o percentual e a quantidade de suporte; e os valores das medidas de qualidade. Também é possível exportar os resultados obtidos para um arquivo texto por meio do botão *Salvar resumo*.

Na Figura 3 tem-se outro relatório gerado pela *Shell Orion* contendo informações dos resultados do algoritmo *FP-Growth*. Neste são listados os *N itemsets* frequentes encontrados com o percentual e a quantidade de suporte. Decidiu-se criar um relatório em separado para os *N itemsets* a fim de facilitar a análise do usuário e não sobrecarregar o relatório principal que contém as regras de associação.

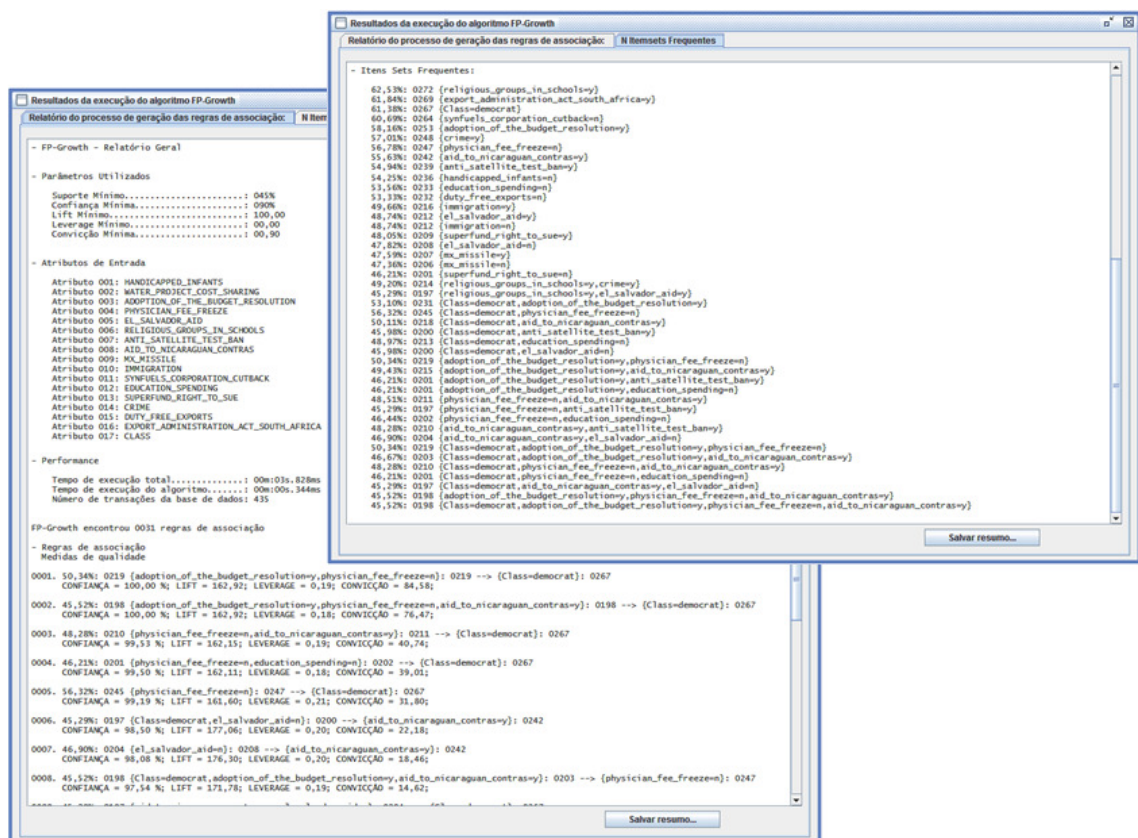


Figura 3. Relatórios gerados

## 3.2 Resultados Obtidos

A base de dados utilizada na avaliação do algoritmo *FP-Growth* na *Shell Orion* contém dados clínicos de pacientes com hepatite. Esta base de dados foi selecionada no UCI Machine Learning Repository que é um repositório gratuito de bases de dados mantido pelo Departamento de Informação e Ciências da Computação da Universidade da Califórnia. A base de dados possui 155 transações distribuídas em duas classes, a dos pacientes que morreram composta por 32 registros e a outra com os que viveram com 123 casos. Dos atributos, 12 são binários, enquanto 6 atributos apresentam valores contínuos, totalizando-se 20 atributos na base de dados.

Os testes realizados abrangem a análise das regras de associação identificadas, a análise dos tempos de processamento e a comparação dos tempos de processamento entre a *Shell Orion* e a ferramenta Weka 3.6.10, pois esta última é uma das ferramentas frequentemente utilizada em pesquisas na área de *data mining*.

### 3.2.1 Regras de Associação Identificadas pelo Algoritmo FP-Growth

Os testes de mesa<sup>1</sup> necessários para se testar a implementação do algoritmo, foram feitos com o exemplo mostrado na modelagem matemática, por isso a primeira análise

<sup>1</sup> Procedimento que simula a execução de um algoritmo utilizando apenas papel e caneta, por meio do teste de mesa é possível verificar as mudanças no conteúdo das variáveis de um algoritmo instrução a



dos resultados feita nessa fase da pesquisa foi uma comparação com o resultado gerado pelo *FP-Growth* na ferramenta Weka.

Executou-se o algoritmo utilizando-se os seguintes parâmetros e atributos de entrada: suporte 70%, confiança 90%, *lift* 100, *leverage* 0 (zero) e convicção 1.01. Na Figura 4 é possível verificar as regras geradas pelo algoritmo *FP-Growth*.

```
FP-Growth encontrou 0007 regras de associação
- Regras de associação
  Medidas de qualidade
0001. 77,42%: 0120 {ASCITES=2}: 0130 --> {VARICES=2}: 0132
    CONFIANÇA = 92,31 %; LIFT = 108,39; LEVERAGE = 0,06; CONVICÇÃO = 1,75;
0002. 72,90%: 0113 {Class=2}: 0123 --> {ASCITES=2}: 0130
    CONFIANÇA = 91,87 %; LIFT = 109,54; LEVERAGE = 0,06; CONVICÇÃO = 1,80;
0003. 70,97%: 0110 {SPLEEN_PALPABLE=2}: 0120 --> {VARICES=2}: 0132
    CONFIANÇA = 91,67 %; LIFT = 107,64; LEVERAGE = 0,05; CONVICÇÃO = 1,62;
0004. 72,26%: 0112 {Class=2}: 0123 --> {VARICES=2}: 0132
    CONFIANÇA = 91,06 %; LIFT = 106,92; LEVERAGE = 0,05; CONVICÇÃO = 1,52;
0005. 77,42%: 0120 {VARICES=2}: 0132 --> {ASCITES=2}: 0130
    CONFIANÇA = 90,91 %; LIFT = 108,39; LEVERAGE = 0,06; CONVICÇÃO = 1,64;
0006. 76,77%: 0119 {ANTIVIRALS=2}: 0131 --> {SEX=1}: 0139
    CONFIANÇA = 90,84 %; LIFT = 101,30; LEVERAGE = 0,01; CONVICÇÃO = 1,04;
0007. 70,32%: 0109 {SPLEEN_PALPABLE=2}: 0120 --> {SEX=1}: 0139
    CONFIANÇA = 90,83 %; LIFT = 101,29; LEVERAGE = 0,01; CONVICÇÃO = 1,03;
```

**Figura 4. Regras encontradas com os parâmetros informados**

Estudou-se a ferramenta Weka em fóruns, tutoriais disponíveis na internet e no livro publicado por Witten, Frank e Hall no ano de 2011. Configurou-se na Weka os mesmos parâmetros usados na *Shell Orion*. No primeiro teste não gerou nenhuma regra de associação.

Após uma nova série de testes, estudos e exploração na ferramenta Weka, constatou-se que o algoritmo *FP-Growth* implementado na Weka trabalha somente com índices binários, isto é, com atributos que possuem dois valores (SIM = + / NÃO = -). E gera as regras considerando somente os valores positivos ou negativos dos atributos na base de dados. Descobriu-se que por *default* a ferramenta vem selecionada para gerar regras com os atributos que possuem índice positivo. Alterou-se para gerar com os atributos que possuem índice negativo.

Desta vez, a ferramenta Weka gerou cinco regras de associação. Comparando-se estas regras (Figura 5) com as regras geradas pela *Shell Orion* verifica-se que as cinco regras geradas na Weka são iguais as cinco primeiras regras geradas na *Shell Orion*. Inclusive os valores para as medidas de qualidade também são os mesmos diferenciando-se somente na forma como ambas mostram o resultado.

#### Weka

```
=== Associator model (full training set) ===
```

```
FPGrowth found 5 rules
```

1. [ASCITES=2]: 130 ==> [VARICES=2]: 120 <conf:(0.92)> lift:(1.08) lev:(0.06) conv:(1.75)
2. [Class=2]: 123 ==> [ASCITES=2]: 113 <conf:(0.92)> lift:(1.1) lev:(0.06) conv:(1.8)
3. [SPLEEN\_PALPABLE=2]: 120 ==> [VARICES=2]: 110 <conf:(0.92)> lift:(1.08) lev:(0.05) conv:(1.62)
4. [Class=2]: 123 ==> [VARICES=2]: 112 <conf:(0.91)> lift:(1.07) lev:(0.05) conv:(1.52)
5. [VARICES=2]: 132 ==> [ASCITES=2]: 120 <conf:(0.91)> lift:(1.08) lev:(0.06) conv:(1.64)

**Figura 5. Execução da ferramenta Weka para atributos com índice negativo**

Uma vez que o algoritmo FP-Growth foi desenvolvido para gerar regras de associação sem para isso gerar candidato, o que resulta em um melhor aproveitamento dos recursos computacionais disponíveis, resolveu-se realizar uma comparação com o resultado do algoritmo Apriori também implementado na Weka. O resultado tem que ser o mesmo, ou no mínimo parecido, porque tanto o Apriori quanto o FP-Growth usam meios diferentes, mas chegam a um mesmo fim, que é a descoberta de regras de associação úteis e fortes. A Figura 6 mostra um resumo dos resultados apresentados.

<p><b>Shell Orion</b></p> <p>FP-Growth encontrou 0007 regras de associação</p> <p>- Regras de associação</p> <p>Medidas de qualidade</p> <p>0001. 77,42%: 0120 {ASCITES=2}: 0130 --&gt; {VARICES=2}: 0132 CONFIANÇA = 92,31 %; LIFT = 108,39; LEVERAGE = 0,06; CONVICÇÃO = 1,75;</p> <p>0002. 72,90%: 0113 {Class=2}: 0123 --&gt; {ASCITES=2}: 0130 CONFIANÇA = 91,87 %; LIFT = 109,54; LEVERAGE = 0,06; CONVICÇÃO = 1,80;</p> <p>0003. 70,97%: 0110 {SPLEEN_PALPABLE=2}: 0120 --&gt; {VARICES=2}: 0132 CONFIANÇA = 91,67 %; LIFT = 107,64; LEVERAGE = 0,05; CONVICÇÃO = 1,62;</p> <p>0004. 72,26%: 0112 {Class=2}: 0123 --&gt; {VARICES=2}: 0132 CONFIANÇA = 91,06 %; LIFT = 106,92; LEVERAGE = 0,05; CONVICÇÃO = 1,52;</p> <p>0005. 77,42%: 0120 {VARICES=2}: 0132 --&gt; {ASCITES=2}: 0130 CONFIANÇA = 90,91 %; LIFT = 108,39; LEVERAGE = 0,06; CONVICÇÃO = 1,64;</p> <p>0006. 76,77%: 0119 {ANTIVIRALS=2}: 0131 --&gt; {SEX=1}: 0139 CONFIANÇA = 90,84 %; LIFT = 101,30; LEVERAGE = 0,01; CONVICÇÃO = 1,04;</p> <p>0007. 70,32%: 0109 {SPLEEN_PALPABLE=2}: 0120 --&gt; {SEX=1}: 0139 CONFIANÇA = 90,83 %; LIFT = 101,29; LEVERAGE = 0,01; CONVICÇÃO = 1,03;</p>	<p><b>Weka</b></p> <p>=== Associator model (full training set) ===</p> <p>FPGrowth found 5 rules</p> <p>1. [ASCITES=2]: 130 ==&gt; [VARICES=2]: 120 &lt;conf:(0.92)&gt; lift:(1.08) lev:(0.06) conv:(1.75)</p> <p>2. [Class=2]: 123 ==&gt; [ASCITES=2]: 113 &lt;conf:(0.92)&gt; lift:(1.1) lev:(0.06) conv:(1.8)</p> <p>3. [SPLEEN_PALPABLE=2]: 120 ==&gt; [VARICES=2]: 110 &lt;conf:(0.92)&gt; lift:(1.08) lev:(0.05) conv:(1.62)</p> <p>4. [Class=2]: 123 ==&gt; [VARICES=2]: 112 &lt;conf:(0.91)&gt; lift:(1.07) lev:(0.05) conv:(1.52)</p> <p>5. [VARICES=2]: 132 ==&gt; [ASCITES=2]: 120 &lt;conf:(0.91)&gt; lift:(1.08) lev:(0.06) conv:(1.64)</p>	<p><b>Weka Apriori</b></p> <p>Best rules found:</p> <p>1. ASCITES=2 130 ==&gt; VARICES=2 120 conf:(0.92)</p> <p>2. Class=2 123 ==&gt; ASCITES=2 113 conf:(0.92)</p> <p>3. SPLEEN_PALPABLE=2 120 ==&gt; VARICES=2 110 conf:(0.92)</p> <p>4. Class=2 123 ==&gt; VARICES=2 112 conf:(0.91)</p> <p>5. VARICES=2 132 ==&gt; ASCITES=2 120 conf:(0.91)</p> <p>6. ANTIVIRALS=2 131 ==&gt; SEX=1 119 conf:(0.91)</p> <p>7. SPLEEN_PALPABLE=2 120 ==&gt; SEX=1 109 conf:(0.91)</p>
---	--	---

**Figura 6. Regras geradas na Shell Orion e Weka**

Observa-se que a *Shell Orion* gerou as mesmas regras de associação que o algoritmo Apriori implementado na Weka, com os mesmos percentuais de suporte e confiança. Comparando com o *FP-Growth* da Weka verifica-se que as cinco primeiras regras da *Shell Orion* são idênticas as cinco regras geradas na Weka. As regras seis e



sete possuem índices positivos e negativos por isso são descartadas pelo algoritmo *FP-Growth* na Weka, mas não são descartadas pelo algoritmo Apriori na mesma ferramenta.

### 3.2.2 Tempos de processamento *Shell Orion* versus Weka 3.6.10

A base de dados original foi replicada gradativamente para se obter conjuntos de diversos tamanhos, fez-se isso por meio do crescimento geométrico que torna possível a execução do algoritmo em diferentes tamanhos de cargas (centenas a milhares) dentro de um pequeno intervalo de iterações:

$$C_k = N \times 2^{k-1}$$

A fim de comparar o desempenho em relação ao tempo de processamento entre as duas ferramentas, realizou-se também a coleta dos tempos gastos pela Weka para executar as mesmas tarefas nas mesmas condições (Tabela 3).

**Tabela 2. Tempos de processamento da *Shell Orion* e da Weka**

Quantidade de transações	<i>Shell Orion</i>	Weka
435	00m: 00s: 015ms	00m: 00s: 008ms
870	00m: 00s: 031ms	00m: 00s: 011ms
1740	00m: 00s: 061ms	00m: 00s: 012ms
3480	00m: 00s: 093ms	00m: 00s: 025ms
6960	00m: 00s: 171ms	00m: 00s: 045ms
13920	00m: 00s: 343ms	00m: 00s: 081ms
27840	00m: 00s: 701ms	00m: 00s: 163ms
55680	00m: 01s: 604ms	00m: 00s: 335ms
111360	00m: 03s: 634ms	00m: 00s: 675ms

Para as análises dos tempos coletados utilizou-se o pacote estatístico *Statistical Package for the Social Sciences* (SPSS). Aplicou-se o teste de normalidade *Shapiro-Wilk*, objetivando verificar se a distribuição dos valores da amostra era normal. O teste resultou em um *p-value* menor que o nível de significância preestabelecido (0,05) constatando que a distribuição não foi normal.

Optou-se pela transformação logarítmica dos tempos. Após o cálculo do *log* natural dos tempos, aplicou-se mais uma vez o teste *Shapiro-Wilk* e desta vez obteve-se uma distribuição gaussiana em ambos os grupos.

Para comparar a homogeneidade das variâncias observadas, utilizou-se o teste *F* de *Levene*, que aceitou a hipótese nula ( $p = 0,673$ ), o resultado comprovou que as variâncias são homogêneas, isso acarretou na utilização do teste *t* de *Student* para amostras independentes objetivando a comprovação das médias dos tempos gastos para execução na *Shell Orion* e Weka.

A realização do teste  $t$  de *Student* revelou que a diferença do tempo de processamento entre a *Shell Orion* e a Weka, nesta mesma base de dados, não foi significativa ( $p = 0,116$ ), aceitando a hipótese nula ( $H_0$ ), portanto, conclui-se que a variação ocorre ao acaso.

#### **4. Considerações Finais**

Este artigo apresentou o algoritmo *FP-Growth*, que utiliza uma abordagem diferente para resolver o problema da descoberta de regras de associação, sem para isso gerar conjunto de candidatos, implementado na tarefa de associação da *Shell Orion Data Mining Engine*, contribuindo com a ampliação das suas funcionalidades.

Analisando os resultados obtidos, verifica-se que a *Shell Orion* gerou corretamente as regras de associação, pois apresentou resultados satisfatórios. Com relação ao tempo de processamento a Weka apresentou valores um pouco melhores, porém, do ponto de vista estatístico, a variação apresentada não foi significativa.

#### **Referências**

- Goldschmidt, R. e Passos, E. (2005) Data mining: uma guia prático. Rio de Janeiro, Elsevier.
- Guillet, F. and Hamilton, H. J. (2007) Quality Measures in Data Mining. Studies in Computational Intelligence 43, Springer, 2007.
- Han, J., Pei, J. and Yin, Y. (2000) “Mining Frequent Patterns Without Candidate Generation”, In: Inf. Conf. on Management of Data (SIGMOD’00), Dallas, Texas, p. 1-12.
- Han, J. and Kamber, M. (2006) Data mining: concepts and techniques. San Francisco, Morgan Kaufmann.
- Tan, Pan-Ning, Steinbach, M. e Kumar, V. (2009) Introdução ao Datamining: mineração de dados, Rio de Janeiro, Ciência Moderna.
- Witten, I. H., Frank, E. and Hall, M. A. (2011) Data mining practical machine learning tools and techniques, Burlington, Morgan Kaufmann, 3. ed.