UMA ANÁLISE COMPARATIVA ENTRE ALGORITMOS ESTATÍSTICOS DE MINERAÇÃO DE DADOS

José Carlos Júnior, Eric Rommel, Daniel Silva
Departamento de Exatas e Tecnologia – Centro Universitário de João Pessoa (UNIPÊ)
58.053-000 – João Pessoa – PB – Brasil,
{jcapjunior, ericrommel, danielpk.lima}@gmail.com

Ryan Ribeiro de Azevedo, Wendell Campos Veras
Centro de Informática-Universidade Federal de Pernambuco
Av. Prof. Luis Freire, s/n, Cidade Universitária - Recife-PE – Brasil
+55 81 21268430, 50740-540
{rra2,wcv}@cin.ufpe.br

Resumo: *A informação é o ativo mais importante para os negócios das corporações, tornando-se algo essencial para ganho de competitividade entre as empresas de pequeno, médio e grande porte. As estratégias assumidas para tal ganho, devem basear-se em informações concretas, visando uma diminuição substancial na ocorrência de erros. Avanços tecnológicos têm facilitado a obtenção dessas informações através de processos de descoberta do conhecimento em banco de dados (Knowledge Discovery in Database – KDD), capaz de identificar padrões e descobrir informações relevantes que auxiliem na formação de posturas estratégicas de marketing, na busca e conquista de clientes, na descoberta de falhas em linhas de produção, dentre outras.*

Palavras-chave: Descoberta de Conhecimento, KDD, *Data Mining*, Algoritmos de Mineração.

Introdução

A previsão na tecnologia é abordada de maneira interessante e diversificada, na maioria das vezes baseando-se em idéias e fantasias, não obstante ser o que move a tecnologia para as constantes evoluções tecnológicas atuais e futuras. A informação armazenada em coleções organizadas e inter-relacionadas de dados persistentes é alvo de inúmeras perquirições na busca da descoberta de conhecimentos ocultos na mineração dos dados concretos, reais e verdadeiros, essencial para a sobrevivência ou crescimento das pequenas, médias e grandes corporações, seja comercial ou acadêmica.

O discernimento dos processos de descoberta de conhecimento, tendo como fundamento as variadas técnicas de descoberta das regras de associações existentes, e a utilização de algumas técnicas computacionais de mineração de dados, é essencial para a correta análise e entendimento do problema.

Este trabalho teve como base metodológica uma pesquisa exploratória e descritiva, na qual se buscou fazer um comparativo entre os algoritmos *Tertius* e *Predictive Apriori,* destacando uma análise mais detalhada e completa dos algoritmos utilizados, assim como as evoluções propostas e os aspectos da integração destas tecnologias com o KDD, conceituando, esclarecendo e despertando o interesse das empresas sobre a importância da sua utilização nas tomadas de decisões.

As demais seções deste artigo estão estruturadas da seguinte forma: a Seção 2 foca nos principais conceitos do processo de descoberta de conhecimento; a Seção 3 foca nos conceitos e benefícios das regras de associação; a Seção 4 foca na apresentação e funcionamento do algoritmo *Tertius*, baseado na Inteligência Artificial; a Seção 5 foca na apresentação e funcionamento do algoritmo *Predictive Apriori*, baseado na Estatística. São apresentados na Seção 6 os experimentos e os resultados. Por fim, na Seção 7 apresenta-se às conclusões e trabalhos futuros.

2. Descoberta de Conhecimento em Bases de Dados

2.1. Fases do KDD

Após a fase inicial, o foco passa a ser a escolha ou seleção da massa de dados a ser minerada, podendo ser um conjunto de dados ou um subconjunto de variáveis onde a extração será realizada. A fase de *Data Cleaning* e Pré-Processamento têm por objetivo assegurar a qualidade dos dados envolvidos no KDD realizando operações básicas como a remoção de ruídos, que podem ser, por exemplo, atributos nulos. A fase seguinte consiste na Seleção e Transformação dos dados em que serão selecionados os atributos realmente interessantes ao usuário, além de transformados utilizando o padrão ideal para aplicar algoritmos de mineração.

Após a realização das fases anteriores, a Mineração de Dados (*Data Mining*) é iniciada. Esta fase é a mais importante do KDD, sendo realizada através da

escolha do método e do algoritmo mais compatível com o objetivo da extração, a fim de encontrar padrões nos dados que sirva de subsídios para descobrir conhecimentos ocultos. Na Seção 3.2. um aprofundamento desta fase será realizado.

A Avaliação ou Pós-Processamento é a fase que identifica, entre os padrões extraídos na etapa de *Data Mining,* os padrões interessantes ao critério estabelecido pelo usuário, podendo voltar à fase inicial para novas iterações. É apresentado na Figura 1 de uma maneira simplificada todas as fases do KDD.

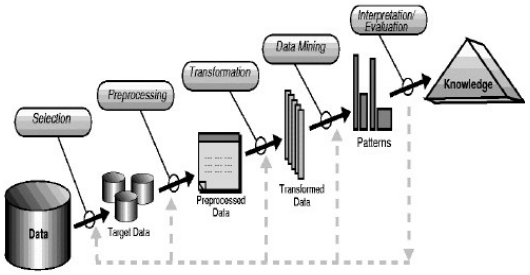

Figura 1: Fases do Processo de KDD [3].

2.2. Mineração de dados

Data Mining ou Mineração de Dados é o processo de pesquisa em grandes quantidades de dados para extração de conhecimento, utilizando técnicas de Inteligência Computacional para procurar relações de similaridade ou discordância entre dados, com o objetivo de encontrar padrões, irregularidades e regras, com o intuito de transformar dados, aparentemente ocultos, em informações relevantes para a tomada de decisão e/ou avaliação de resultados.

A mineração de dados possibilita a busca em grandes bases de dados de informações desconhecidas, permitindo aos gestores uma maior agilidade nas tomadas de decisões. Uma empresa que utiliza *Data Mining* é capaz de criar parâmetros para entender o comportamento do consumidor, identificando afinidades entre as escolhas de produtos e serviços, prevendo hábitos de compras e analisando comportamentos habituais para detecção de fraudes [8].

2.2.1. Métodos de *Data Mining*

De acordo com [3], existem diversos métodos de *Data Mining* para encontrar respostas ou extrair conhecimento em repositórios de dados, sendo os mais importantes para o KDD: Classificação, Modelos de Relacionamento entre Variáveis, Análise de Agrupamento, Sumarização, Modelo de Dependência, Regras de Associação e Análise de Séries Temporais. Deve-se ressaltar que a maioria desses métodos serve de base para técnicas das áreas de aprendizado de máquina, reconhecimento de padrões e estatística. Um breve resumo dos métodos mais importantes é descrito a seguir.

A Classificação associa ou classifica um item a uma ou várias classes categóricas pré-definidas, utilizando comumente uma técnica estatística chamada análise discriminante, objetivando envolver a descrição gráfica ou algébrica das características diferenciais das observações de várias populações, além da classificação em uma ou mais classes pré-determinadas.

Os Modelos de Relacionamento entre Variáveis associam um item a uma ou mais variáveis de predição de valores reais, consideradas variáveis independentes ou exploratórias. Regressão linear simples, múltipla e modelos lineares por transformação são as técnicas mais utilizadas para verificar a existência do relacionamento funcional entre duas variáveis quantitativas.

A Análise de Agrupamento, ou *Cluster*, associa um item a uma ou várias classes categóricas (ou *clusters*), determinando as classes pelos dados, independentemente da classificação pré-definida. Os clusters são definidos por meio do agrupamento de dados baseados em medidas de similaridade ou modelos probabilísticos, visando detectar a existência de diferentes grupos dentro de um determinado conjunto de dados e, em caso de sua existência, determinar quais são eles.

A Sumarização determina uma descrição com dispersão reduzida para um dado subconjunto no pré-processamento dos dados, freqüentemente utilizadas na análise de descobrimento de dados. Podemos citar como exemplos simples de sumarização de dados, as medidas de posição e variabilidade.

O Modelo de Dependência descreve dependências entre variáveis. Modelos de dependência existem em dois níveis: estruturado e quantitativo. O nível estruturado especifica, geralmente em forma de gráfico, quais variáveis são localmente dependentes. O nível quantitativo especifica o grau de dependência, usando alguma escala numérica.

As Regras de Associação determinam relações entre campos de um banco de dados, contribuindo para a tomada de decisão, recebendo assim, na Seção 3.3, um maior aprofundamento. Por fim, a Análise de Séries Temporais determinam características seqüenciais, como dados com dependência no tempo. Seu objetivo é modelar o estado do processo extraindo e registrando desvios e tendências no tempo.

2.3. Regras de Associação

O processo de extração de regras de associação foi proposto inicialmente por [1], e representa um padrão ocorrido em combinações de itens com determinada freqüência em uma base de dados [5].

As regras de associação foram definidas através da seguinte formalização [2]: seja $I = \{I_1, I_2, I_3,..., I_n\}$ um conjunto de itens e T um conjunto de transações, sendo cada transação T' um conjunto de itens, de forma que

$T' \subseteq I$, e cada T' será associado a um único identificador $T'IT$, tal que $x \subseteq T'$, $x \subseteq I$. Desta maneira a regra de associação gerada terá como forma $x \rightarrow y$, ou seja, x será o antecedente e y o conseqüente da regra de associação criada.

Outra abordagem para o conceito de regras de associação leva em consideração os conceitos de Confiança e Suporte [6], em que este estima a probabilidade da ocorrência de um conjunto de itens I em uma transação T', e aquele, para um conjunto Z de itens, refere-se à porcentagem de transações de uma base de dados que contem esses itens de Z.

No Quadro 1 [10], tem-se a fórmula utilizada no Suporte, para uma regra $X \rightarrow Y$, onde X e Y são conjuntos de itens, tendo por numerador a quantidade de transações em que ambos, X e Y, estão ocorrendo simultaneamente, e por denominador o número total de transações.

A Confiança é evidenciada utilizando a mesma regra $X \rightarrow Y$, mas seu denominador representa o valor total de transações em que o item X ocorre [10], visto no Quadro 2.

$$Suporte = \frac{Freqüência\ de\ X\ e\ Y}{Total\ de\ T}$$

Quadro 1: Suporte

$$Confiança = \frac{Freqüência\ de\ X\ e\ Y}{Freqüência\ de\ X}$$

Quadro 2: Confiança

3. Algoritmos de Mineração de Dados

3.1. Algoritmo *Tertius*

De acordo com [4], o Algoritmo *Tertius* utiliza descoberta de lógica de primeira ordem para encontrar cláusulas lógicas nas bases de dados das quais são criadas todas as possíveis instâncias, com estimativas otimistas, usando algoritmos de Busca pela Melhor Escolha (*Best-First Search*), com a abordagem *A**, percorrendo o espaço de possibilidades das regras de associação através de heurísticas e realizando ainda um refinamento de operadores não redundantes para descartar buscas desnecessárias. [7] explica que o *Tertius* descobre regras de acordo com uma métrica de confirmação, procurando regras com múltiplas condições, como o *Apriori* [2], mas diferente no fato destas condições serem *OU* lógico e não *E* lógico. A Tabela 1 mostra essa diferença num exemplo prático.

Tabela 1: Diferença entre o *Apriori* e o *Tertius* [7]

Algoritmo	Exemplo de regra	Suporte
Apriori	*play = no* → *humidity = high*	4
Tertius	*play = no* → humidity = *high* *OR humidity = normal*	

Uma busca, como solução do problema, é o processo que gera e analisa seqüências de ações para alcançar um objetivo, percorrendo um caminho entre o estado inicial e o estado final, sendo chamado de qualidade da solução o custo desse caminho. A soma do tempo gasto com o consumo de memória resulta no custo da busca. A soma desses dois, custo do caminho e custo da busca, estima o custo total.

Numa busca heurística, estima-se qual o melhor nó da fronteira a ser expandido com base em funções heurísticas, utilizando como estratégia de busca a melhor escolha tendo como direção, do estado inicial para o objetivo e do objetivo para o estado inicial.

No Algoritmo A*, o custo total da solução é minimizado associando economia, completude e decidibilidade, tendo como função de avaliação $f(n) = g(n) + h(n)$, onde $g(n)$ é a distância de n ao nó inicial e $h(n)$ é a distância estimada de n ao nó final. Se h é admissível, $f(n)$ nunca irá superestimar o custo real da melhor solução através de n. A função heurística h é monotônica se $h(n) \geq h(sucessor(n))$, isso vale para a maioria das funções heurísticas.

Transferindo-se esse comportamento para a função de avaliação $f(n) = g(n) + h(n)$, temos que $f(sucessor(n)) \geq f(n)$, uma vez que $g(n)$ é não decrescente. Em outras palavras, o custo de cada nó gerado no mesmo caminho nunca diminui. Quando $h(n)$ não é monotônica, para se garantir a monotonicidade de $f(n)$, temos: quando $f(sucessor(n)) < f(n)$ usa-se $f(sucessor(n)) = max(f(n), g(sucessor(n)) + h(sucessor(n)))$.

No Quadro 3, temos o algoritmo *Tertius* representado numa linguagem de alto nível.

```
agenda ← {empty rule}
while agenda is not empty
        rule ← first rule of the agenda
        if canRefine(rule)
                refine rule
                for each child
                        if canCalculateOptimistic(child)
                                calculate optimistic estimate
                        else remove child
                sort remaining children according to optimistic estimate
                for each remaining child
                        if canExplore(child)
                                if canStoreInNodes(child)
                                        add child to the agenda
                                if canCalculateConfirmation(child)
                                        calculate confirmation
                                        add child to remaining
                        else break
                sort remaining children according to confirmation
                for each remaining child
                        if canStoreInResults(child)
                                add child to the results
                        else break
                sort agenda according to optimistic estimate
        else break
```

Quadro 3: *Tertius* em Linguagem de Alto Nível

A primeira coluna do resultado obtido com a aplicação do algoritmo *Tertius*, Quadro 4, indica o número da regra obtida. A segunda coluna informa o grau de confirmação da regra e enquanto a terceira coluna, a freqüência relativa desta mesma regra. Os próximos campos formam a regra encontrada.

```
7. /* 0,223743 0,000210 */ grupo = CAMA and idade
= 38_42 ==> valor = 901_MAX or sexo = M
```

Quadro 4: Exemplo de uma linha do resultado do Algoritmo *Tertius*

É com base nessas informações que um analista especializado pode realizar suas considerações e efetivar suas próximas estratégias. Ao final do resultado, o *Tertius* descreve também a quantidade de hipóteses consideradas e a quantidade explorada., e ainda a duração do processamento.

3.2. Algoritmo *Predictive Apriori*

Utilizado para encontrar associações de grande valor, conveniência ou interesse entre itens de dados, o Algoritmo *Predictive Apriori* foi definido por [9] com base no algoritmo *Apriori*, que dá importância ao Suporte e à Confiança na geração de regras associativas, combinando-os em uma medida única, chamada de *Predictive Accuracy*, para encontrar as *n* melhores regras de associação já ordenadas em sua exibição.

Para eliminar regras que não se manifestam com freqüência nas bases de dados é usado um estimador chamado Suporte, sendo este uma porcentagem de registros onde aparecem juntos *X* e *Y*, sobre o total de registros. Um alto Suporte representa uma maior quantidade de regras, porém, com pouca precisão nas regras geradas em relação aos futuros dados.

Já a Confiança, indica o grau de acerto da regra, com base na porcentagem de registros que contêm *X* e *Y* sobre o total de registros que possuem *X*. Confiança alta possibilita a geração de regras confiáveis para poucos registros, podendo apresentar regras otimistas em relação à realidade. Este otimismo pode ser corrigido através do cálculo do Suporte.

O algoritmo *Predictive Apriori*, através de uma distribuição binomial, busca uma relação entre Suporte e Confiança de maneira a potencializar a ocasião favorável de uma íntegra previsão de dados não analisados, sendo estes dados, futuros ou não, utilizados na mineração dos dados. Para cada regra, a medida *Predictive Accuracy*, $c(x \rightarrow y)$, servirá para selecionar as *n* principais regras geradas em ordem decrescente. Esta medida fundamenta-se na probabilidade de uma previsão correta para a regra, ajustando Suporte e Confiança e o aumento gradativo do Suporte mínimo para retornar as *n* regras de associação selecionadas.

O cálculo da *Predictive Accuracy* esperada [9] é representado no Quadro 5, sendo $\hat{c}$ a Confiança, *r* um registro da base de dados *D*, sendo originado por um processo *P* e *s(X)* o Suporte:

$$E(c(r)|\hat{c}(r), s(X)) = \frac{\int cB[c, s(X)](\hat{c}(r))P(c)dc)}{\int B[c, s(X)](\hat{c}(r))P(c)dc)}$$

Quadro 5: *Predictive Accuracy*

Nos Quadros 6 e 7, têm-se a representação em linguagem de alto nível do algoritmo *Predictive Apriori* e do algoritmo *Rule Gen*, sendo este utilizado por aquele. É possível visualizar também outros estimadores utilizados na sua estrutura, como a geração dos itens *sets*, a seleção no espaço de hipóteses por ajuste dinâmico, geração de regras de associação e a remoção de regras redundantes.

1. **Input:** n (desired number of association rules), database with items $a_1, \ldots, a_k$.
2. **Let** $\tau = 1$.
3. **For** $i = 1 \ldots k$ **Do:** Draw a number of association rules $[x \Rightarrow y]$ with i items at random. Measure their confidence (provided $s(x) > 0$). Let $\pi_i(c)$ be the distribution of confidences.
4. **For** all c, **Let** $\pi(c) = \frac{\sum_{i=1}^{k} \pi_i(c)\binom{k}{i}(2^i-1)}{\sum_{i=1}^{k}\binom{k}{i}(2^i-1)}$.
5. **Let** $X_0 = \{\emptyset\}$; **Let** $X_1 = \{\{a_1\}, \ldots, \{a_k\}\}$ be all item sets with one single element.
6. **For** $i = 1 \ldots k-1$ **While** ($i = 1$ or $X_{i-1} \neq \emptyset$).
   (a) **If** $i > 1$ **Then** determine the set of candidate item sets of length i as $X_i = \{x \cup x' | x, x' \in X_{i-1}, |x \cup x'| = i\}$. Generation of X_i can be optimized by considering only item sets x and $x' \in X_{i-1}$ that differ only in the element with highest item index. Eliminate double occurrences of item sets in X_i.
   (b) Run a database pass and determine the support of the generated item sets. Eliminate item sets with support less than τ from X_i.
   (c) **For** all $x \in X_i$ **Call RuleGen**(x).
   (d) **If** *best* has been changed, **Then Increase** τ to be the smallest number such that $E(c|1, \tau) > E(c(best[n])|\hat{c}(best[n]), s(best[n]))$ (refer to Equation 6). **If** $\tau >$ database size, **Then Exit**.
   (e) **If** τ has been increased in the last step, **Then** eliminate all item sets from X_i which have support below τ.
7. **Output** $best[1] \ldots best[n]$, the list of the n best association rules.

Quadro 6: Algoritmo *Predictive Apriori*: Descoberta das *n* regras de associação preditas

10. **Let** γ be the smallest number such that $E(c|\gamma/s(x), s(x)) > E(c(best[n])|\hat{c}(best[n]), s(best[n]))$.
11. **For** $j = 1 \ldots k - |x|$ (number of items not in x)
    (a) **If** $j = 1$ **Then Let** $Y_1 = \{a_1, \ldots a_k\} \setminus x$.
    (b) **Else Let** $Y_j = \{y \cup y' | y, y' \in Y_{j-1}, |y \cup y'| = j\}$ analogous to the generation of candidates in step 6a.
    (c) **For** all $y \in Y_j$ **Do**
        i. Measure the support $s(x \cup y)$. **If** $s(x \cup y) \leq \gamma$, **Then** eliminate y from Y_j and **Continue** the for loop with the next y.
        ii. Calculate predictive accuracy $E(c([x \Rightarrow y])|s(x \cup y)/s(x), s(x))$ according to Equation 6.
        iii. **If** the predictive accuracy is among the n best found so far (recorded in *best*), **Then** update *best*, remove rules in *best* that are subsumed by other, at least equally accurate rules (utilize Theorem 1 and test for $x \subseteq x' \wedge y \supseteq y'$), and **Increase** γ to be the smallest number such that $E(c|\gamma/s(x), s(x)) \geq E(c(best[n])|\hat{c}(best[n]), s(best[n]))$.
12. **If** any subsumed rule has been erased in 11(c)iii, **Then** recur from step 10.

Quadro 7: Algoritmo *RuleGen*: Pesquisa eficientemente por todas as regras com corpo *x*

Um exemplo dos resultados com a execução do *Predictive Apriori* é exibido no Quadro 8, informando no primeiro campo o número da regra obtida. A última coluna informa o valor de precisão da regra (*accuracy*). Entre a primeira e a última coluna está a regra encontrada pelo algoritmo.

```
4. grupo=A.SERVICO idade=38_42 151 ==> sexo=M
151    acc:(0.99487)
```

Quadro 8: Resultado de uma linha do *Predictive Apriori*

5. Experimentos e Resultados

A base de dados utilizada nos experimentos e avaliação dos algoritmos Predictive Apriori e *Tertius* é proveniente de uma empresa atuante na área de varejo, atacado, consórcio, empréstimo e garantia estendida, optante por um acordo NDA (*Non-Disclosure Agreement*). O *software* de mineração de dados utilizado para realizar a geração de padrões úteis foi o *WEKA* (http://www.cs.waikato.ac.nz/ml/weka/), tendo sido desenvolvida na linguagem *Java* pela Universidade de *Waikato* na Nova Zelândia, o qual trabalha com diversas técnicas de *Data Mining*, além de ser um *software* livre e de fácil manuseio.

Após análise do banco de dados, foi selecionada uma amostra de 102.000 registros identificando as vendas realizadas no período de 01 de janeiro de 2008 a 30 de abril de 2008, de cerca de 10 filiais situadas no estado da Paraíba. A Tabela 2 destaca os atributos da base de dados necessários ao processo de mineração de dados.

Tabela 2: Atributos submetidos à mineração.

Atributo	Descrição	Valores Distintos
GRUPO	Grupo no qual o produto foi classificado	4
VALOR	Faixa de valores da compra	6
IDADE	Faixa etária do cliente.	6
SEXO	Sexo do cliente.	2

O atributo GRUPO representa o grupo na qual o produto foi classificado, com 4 valores distintos, a saber: Eletro, Magazine, Móveis, Telecomunicação. Os valores das compras foram decompostos por faixa de valores: abaixo de R$100,00, de R$101,00 à R$300,00, de R$301,00 à R$500,00, de R$501,00 à R$700,00, de R$701,00 à R$900,00 e acima de R$900,00. A idade também foi um atributo organizado em faixa de valores: de 18 a 23 anos, de 24 a 29, de 30 a 37, de 38 a 42, de 43 a 50 e acima de 50 anos. Por fim, o atributo SEXO do cliente.

No cálculo dos desempenhos em termos de alocação de recursos fez-se uso do programa *CProcess (http://www.nirsoft.net/)*, capaz de registrar o valor máximo de alocação de memória RAM. No período dos experimentos estavam executando apenas o Weka e o CProcess, objetivando resultados concretos e contundentes.

Primeiramente definiu-se o Domínio da Base de Dados como sendo um ambiente logístico onde seria levado em consideração o grupo do produto (classificação em categoria), a faixa de preço, a faixa etária e sexo do cliente que realizou a compra. Em seguida, foi definido o foco da avaliação dos algoritmos em: Tempo de Processamento (minutos), Alocação de Memória (MB), Quantidade de Regras Geradas (número de regras) e Qualidade das Regras Geradas.

Com esse escopo definido, a Limpeza de Dados foi feita mediante os resultados obtidos nas consultas iniciais em relação aos cadastros de cliente, grupos de produtos e valores. Visando a remoção de inconsistências, atributos nulos, entre outros. Como também garantir um grau de homogeneidade plausível dos diferentes aspectos e características de cada atributo. Destarte, o almejo de uma maior variação nas regras que serão geradas pelo processo de mineração.

Foram selecionados os atributos indicados na Tabela 1 e transformados para o layout padrão do arquivo utilizado pelo *Weka*, iniciando a Mineração de Dados a fim de encontrar padrões que sirvam de subsídios no descobrimento dos conhecimentos ocultos nos dados.

Os resultados apresentados na Tabela 3 evidenciam que o algoritmo *Tertius* obteve um tempo menor de processamento enquanto a quantidade máxima de alocação de memória foi maior que o algoritmo *Predictive Apriori*. O desempenho foi uniforme para os dois algoritmos, encontrando 10 regras de associação, porém, no que tange a qualidade dessas, avaliadas pelo Gerente de Negócios da empresa, o algoritmo *Tertius* retornou regras obvias (Quadro 9) enquanto as regras extraídas pelo algoritmo *Predictive Apriori* apresentaram conhecimentos relevantes (Quadro 10), pois quatro regras apresentam um conhecimento novo já que não haviam identificado que no segmento Magazine, com produtos até R$100,00, tinham maior venda entre os indivíduos do sexo feminino até 29 anos, além de verificar a preferência masculina por produtos eletro-eletrônicos de custo acima dos R$700,00. O algoritmo Predictive Apriori trouxe duas regras com informações duplicadas e as quatro regras restantes foram consideradas óbvias, porém, corretas, levando em consideração o domínio de negócio e com relação à faixa etária dos compradores, valor médio de compras e sexo dos compradores.

Tabela 3: Resultados obtidos com a aplicação dos dois algoritmos estudados

Algoritmo	Memória (MB)	Tempo (s)	Regras geradas
Predictive Apriori	57	178	10
Tertius	61	0,05	10

```
Tertius
=======

 1. /* 0,032916 0,013106 */ valor = 901_MAX ==> idade = 51_MAX
 2. /* 0,017703 0,015636 */ valor = 901_MAX ==> idade = 43_50
 3. /* 0,011872 0,001037 */ valor = 901_MAX and idade = 24_29 ==> sexo = M
 4. /* 0,008954 0,001094 */ valor = 901_MAX and idade = 38_42 ==> sexo = M
 5. /* 0,007344 0,010041 */ valor = 901_MAX ==> sexo = M
 6. /* 0,004962 0,016639 */ valor = 901_MAX ==> idade = 24_29
 7. /* 0,003498 0,002576 */ valor = 901_MAX and idade = 51_MAX ==> sexo = F
 8. /* 0,000902 0,000661 */ valor = 901_MAX and idade = 18_23 ==> sexo = M
 9. /* 0,000742 0,001311 */ valor = 901_MAX and idade = 30_37 ==> sexo = F
10. /* 0,000000 0,019295 */ valor = 901_MAX ==> FALSE
```

Quadro 9: Resultado obtido com o algoritmo Tertius.

```
Best rules found:

 1. valor=901_MAX idade=18_23 sexo=M 50 ==> grupo=ELETRO 49    acc:(0.96611)
 2. grupo=MAGAZINE idade=18_23 sexo=M 48 ==> valor=0_100 47    acc:(0.9644)
 3. grupo=MAGAZINE valor=301_500 idade=38_42 17 ==> sexo=M 17    acc:(0.96314)
 4. valor=901_MAX idade=18_23 108 ==> grupo=ELETRO 105    acc:(0.96116)
 5. grupo=MAGAZINE valor=301_500 idade=30_37 38 ==> sexo=F 37    acc:(0.95418)
 6. grupo=MAGAZINE idade=24_29 sexo=F 297 ==> valor=0_100 284    acc:(0.95162)
 7. valor=901_MAX idade=51_MAX sexo=M 226 ==> grupo=ELETRO 216    acc:(0.9502)
 8. valor=701_900 idade=30_37 sexo=M 183 ==> grupo=ELETRO 174    acc:(0.94439)
 9. valor=701_900 idade=51_MAX sexo=M 180 ==> grupo=ELETRO 171    acc:(0.94349)
10. grupo=MAGAZINE idade=18_23 278 ==> valor=0_100 263    acc:(0.94284)
```

Quadro 10: Resultado obtido com o algoritmo Predictive Apriori

Após a análise das regras e a verificação de um nicho de mercado a ser investido, os gestores optaram por medidas estratégicas específicas, com campanhas de marketing e promoções específicas definidas pelo entendimento que tiveram do conhecimento obtido.

6. Conclusões e Trabalhos Futuros

Este trabalho demonstrou o potencial para obter regras que trazem conhecimento a partir de algoritmos que utilizam a estatística como base, mesmo tendo sido verificado a necessidade de um aperfeiçoamento do algoritmo Predictive Apriori no que tange o seu desempenho, além de demonstrar na prática como as diversas tecnologias ligadas ao processo de descoberta de conhecimento em bases de dados podem apoiar as tomadas de decisões, de forma a manter as organizações competitivas com relação à concorrência e, principalmente, manterem-se no mercado.

Como trabalho futuro, pode-se realizar uma comparação entre outros algoritmos de regras de associação, realizando experimentos que integrem ferramentas de mineração de dados com ferramentas específicas para gestão do conhecimento, apoiando ainda mais o nível gerencial corporativo para aumentar e propiciar melhores condições na tomada de decisões de forma eficaz e eficiente.

Referências

[1] Agrawal, R.; Imielinski T.; Srikant R. (1993) "Mining Association Rules between Sets of Items in Large Databases", Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, Washington, Estados Unidos, p. 207–216.

[2] Agrawal, R. *et al.* (1994) "Fast algorithms for mining association rules in large databases", in: International Conference on Very Large Data Bases, VLDB, 20, Santiago, Proceedings Hove: Morgan Kaufmann, p. 478-499.

[3] Fayyad, U. M., Piatetsky Shapiro, G., Smyth, P. & Uthurusamy, R. (1996) "Advances in Knowledge Discovery and Data Mining", AAAIPress, The Mit Press.

[4] Flach, Peter A.; Lachiche, Nicolas (2001) "Confirmation-Guided Discovery of First-Order Rules with Tertius" Machine Learning, 42, 61–95, Kluwer Academic Publishers

[5] Gonçalves, Eduardo Corrêa (2005) "Regras de Associação e suas Medidas de Interesse Objetivas e Subjetivas", UFF - Universidade Federal Fluminense, in INFOCOMP – Journal of Computer Science, Rio de Janeiro, RJ.

[6] Győrödi, Cornelia *et al.* (2004) "A Comparative Study of Association Rules Mining Algorithms", in: 1st Romanian, Hungarian Joint Symposium on Applied Computational Intelligence, Oradea, Romania.

[7] Peixoto, João José Lopes (2006) "Algoritmos de Aprendizagem de Contexto em Computação Ubíqua: Avaliação para o caso de estudo em PDAs". 172f Dissertação de mestrado em Informática e Sistemas. Faculdade de Ciências e Tecnologia - Universidade de Coimbra.

[8] Pinto, F.; Santos M. F. (2005) "Descoberta de Conhecimento em Bases de Dados em Actividades de CRM"; Datagadgets 2005; 1º Congresso Espanhol de Informática CEDI 2005; Granada.

[10]Scheffer, Tobias (2001) "Finding association rules that trade support optimally against confidence", Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-01).

[10]Vasconcelos, L. M. R. de; Carvalho, C. L. de (2004) "Aplicação de Regras de Associação para Mineração de Dados" na Web. UFG. Relatório Técnico. Goiás, GO.