Aplicação de *Data Mining* em *Data Warehouse* Desenvolvimento da Ferramenta *ToolMiner*

Nilo Braccini Pessano, Carine Halmenschlager

Curso de Ciência da Computação – Universidade de Santa Cruz do Sul (UNISC)
Av. Independência, 2293 – 96.815-900 – Santa Cruz do Sul – RS – Brasil
nilo@pessano.com.br, carine.halmenschlager@direkt.com.br

Resumo. *Este trabalho utilizou um data warehouse como fonte de informação para a aplicação de data mining. Para isso, analisou-se uma base de dados real de uma empresa de varejo, correspondente a um módulo do software Sadig, na qual se identificaram algumas tarefas de data mining que poderiam ser aplicadas. Para resolver essas tarefas, escolheu-se dois algoritmos: o K-means e o C4.5, implementados na ferramenta denominada ToolMiner. Os algoritmos foram validados em bases de repositórios públicos e aplicados no data warehouse para resolver duas das tarefas identificadas. A partir dos resultados das tarefas, são analisados os conhecimentos gerados.*

Abstract. *This work used one data warehouse as a source of information for data mining application. To do this, it was analyzed a real database of a retail company, corresponding to a module of Sadig software, in which it had been identified some tasks of data mining that could be applied. To solve these tasks, it was chosen two algorithms: the K-means and the C4.5, implemented in the ToolMiner software. The algorithms had been validated in databases of public repositories and applied to the data warehouse to solve two of the identified tasks. From the results of the tasks, the generated knowledge is identified.*

1. Introdução

A *Descoberta de Conhecimento em Base de Dados* (DCBD) visa descobrir padrões ocultos em base de dados que possam representar conhecimento válido [Fayyad 1996]. Possui três fases principais: o pré-processamento, quando os dados são selecionados, limpos e transformados; o *data mining* (DM), que aplica algoritmos específicos nos dados gerados pelo pré-processamento; o pós-processamento, que analisa os resultados da aplicação desses algoritmos, procurando identificar os conhecimentos válidos.

O *data warehouse* (DW) fornece uma fonte de informação histórica a respeito dos negócios da organização, onde os dados são selecionados e extraídos das bases operacionais e organizados por assunto, no qual os sistemas de apoio à decisão farão as suas consultas, gerando informações aos níveis gerencial e estratégico da organização [Inmon 1997]. Apresenta-se como uma importante fonte de informação para a DCBD, pois pode reduzir o tempo de pré-processamento, visto que os seus dados já estão selecionados, organizados por assunto e armazenados em um único repositório. Além disso, como são dados utilizados nos níveis gerenciais da organização, possuem uma boa qualidade de informação.

Neste contexto, este trabalho utiliza um DW como fonte de informação para a aplicação de DM. Para isso, é utilizado um DW de uma empresa que possui uma rede de lojas de venda de eletrodomésticos, móveis, aparelhos eletrônicos e ferramentas e que foi implementado pelo software SADIG. A base de dados utilizada é de um módulo do SADIG denominado de *Faturamento Realizado*, composta por uma única tabela que contém os registros de itens vendidos pelas lojas da empresa, em que cada registro armazena todas as informações da venda. É utilizada uma amostra desta base de dados de um período de duas semanas de venda, num total de 29.373 registros.

Inicialmente é efetuada uma análise do DW, em conjunto com os departamentos de marketing e TI da empresa, com o objetivo de identificar quais as tarefas de *data mining* poderiam ser resolvidas neste domínio. São identificadas seis tarefas, três do tipo classificação, duas de agrupamento e uma de associação. Descartou-se a tarefa de associação após ser descoberto que, na base de dados, 95% das vendas eram de apenas um item, restringindo a apenas 5% os itens vendidos em conjunto, o que representa um percentual muito baixo de registros a ser utilizado por esta tarefa, sobrando para o estudo as tarefas de classificação e agrupamento. Para a resolução dessas tarefas, escolheu-se os algoritmos *C4.5* e *K-means*, que estão entre os mais utilizados pelas ferramentas de *data mining* para a resolução desses tipos de tarefas.

2. Tarefas e algoritmos implementados

A tarefa de classificação utiliza o aprendizado supervisionado[1] para classificar um conjunto de treinamento em classes pré-definidas. Para tanto são escolhidos os atributos descritivos e o atributo preditivo, também denominado atributo meta. Uma técnica para resolver essa tarefa é a indução de uma árvore de decisão, que permite a geração de um conjunto de regras de classificação [Halmenschlager 2002].

A tarefa de agrupamento, também denominada de *clusterização*, utiliza o aprendizado não supervisionado[2] para classificar um conjunto de dados em grupos (*clusters*). Os grupos são formados por registros que possuem características similares, determinadas pela aproximação dos valores dos atributos analisados [Halmenschlager 2002].

Para resolver as tarefas de classificação, implementou-se o algoritmo *C4.5*. Criado por Ross Quinlan (1993), este algoritmo constrói uma árvore de decisão com um número variável de partições, escolhendo como nodos da árvore os atributos mais informativos, que são os que possuem o maior *ganho de informação*, utilizando, para isto, o *critério da entropia* [Feldens 1996]. O algoritmo termina a construção de um ramo da árvore quando o *critério de parada* é satisfeito, ponto em que é adicionado um nodo folha. O *critério de parada* comum é quando todos os registros do subconjunto pertencem a uma mesma classe, porém outros *critérios de parada* podem ser definidos. Na implementação deste trabalho, definiu-se um valor percentual para o *critério de parada*, que indica a quantidade de registros da mesma classe em relação ao subconjunto analisado, que são suficientes para determinar uma folha, podendo ser configurado dinamicamente.

[1] Aprendizado supervisionado é quando são fornecidas as saídas esperadas do algoritmo.
[2] Aprendizado não supervisionado é quando que não são fornecidas as saídas esperadas.

Para resolver as tarefas de agrupamento, implementou-se o algoritmo *K-means*. Proposto por MacQueen em 1967 [Engel 2002], o *K-means* classifica os dados em grupos (*clusters*), em que cada *cluster* possui um valor central denominado *centróide*. O algoritmo trabalha calculando as distâncias dos atributos analisados em relação aos *centróides* de cada cluster. Após os cálculos das distâncias de todos os registros, são calculadas as médias dos atributos e ajustados os valores dos *centróides*, até que estes não mais se alterarem. Na implementação deste algoritmo, para o cálculo das distâncias, escolheu-se a fórmula da *distância euclidiana*. A fim de determinar o número de *clusters* a serem criados é definida a variável *k*, que pode ser configurada dinamicamente. O *K-means* trabalha com atributos numéricos, que podem estar em escalas distintas e dificultar a análise de similaridade do algoritmo; por esse motivo se implementou uma função que realiza o *escalonamento* de valores, transformando-os para uma mesma escala.

3. Aplicação do *data mining*, ferramenta *ToolMiner*

Para a aplicação do DM, desenvolveu-se uma ferramenta denominada *ToolMiner*, escrita na linguagem *Delphi 7* e que utiliza o gerenciador de banco de dados *InterBase* 6.5, podendo ser executada no ambiente Windows. As funções principais da ferramenta são: *Importação*, que importa dados externos no formato nativo do SADIG e no formato texto; *Pré-Processamento*, que permite a seleção dos atributos a serem trabalhados e possui uma função de discretização[3] manual de atributos; *Mineração*, que aplica os algoritmos *K-means* e *C4.5*; *Visualização*, que permite analisar os resultados dos algoritmos em vários formatos distintos; *Relatórios*, que fornece um *log* dos passos da execução dos algoritmos, exibindo os valores parciais e finais. Para a validação dos algoritmos, utilizaram-se bases de dados de repositórios públicos de DM, disponíveis na internet [Monash 2005][UCI 2005][Weka 2005].

3.1 Importação de dados

Na importação de dados, a ferramenta utiliza um arquivo de definição (Figura 1), que serve para reconhecer o formato do arquivo de dados e criar a tabela interna.

```
# Arquivo de definicao da base de dados de faturamento
@file-name MOD0001.DBF;
@table-name Faturamento;
@attribute DtVenda,D,8,0;
@attribute Filial,C,30,0;
@attribute LinhaProd,C,30,0;
@attribute VlVenda,N,12,2;
```

Figura 1. Exemplo do arquivo de definição da *ToolMiner*

Após a importação do DW de *Faturamento*, foram necessárias algumas operações de limpeza na base de dados, efetuadas através de comandos SQL. Visando facilitar a execução destas operações, implementou-se na própria ferramenta uma opção que permite executar comandos SQL na base interna.

[3] Função de discretização: transforma os valores dos atributos de contínuos para nominais.

Para a aplicação da ferramenta no domínio, escolheu-se duas tarefas de DM, entre as tarefas identificadas anteriormente, avaliadas como as mais importantes pela empresa: *tarefa de agrupamento de vendas por lucratividade* e *tarefa de classificação de clientes por linha de produto*.

3.2 Tarefa de agrupamento de vendas por lucratividade

Esta tarefa teve como objetivo criar *clusters* com base nos atributos: valor da venda (VLVENDA) e valor do lucro (VLLUCRO), o que possibilitou a descoberta das relações existentes entre esses atributos e a identificação das características de cada *cluster*.

Inicialmente, no módulo de pré-processamento da ferramenta, efetuou-se a seleção dos atributos e foi criada a tabela de trabalho para ser utilizada na fase de mineração. Após, no módulo de mineração (Figura 2), definiram-se os parâmetros para a execução do algoritmo *K-means*: criação de três grupos e execução da função de escalonamento.

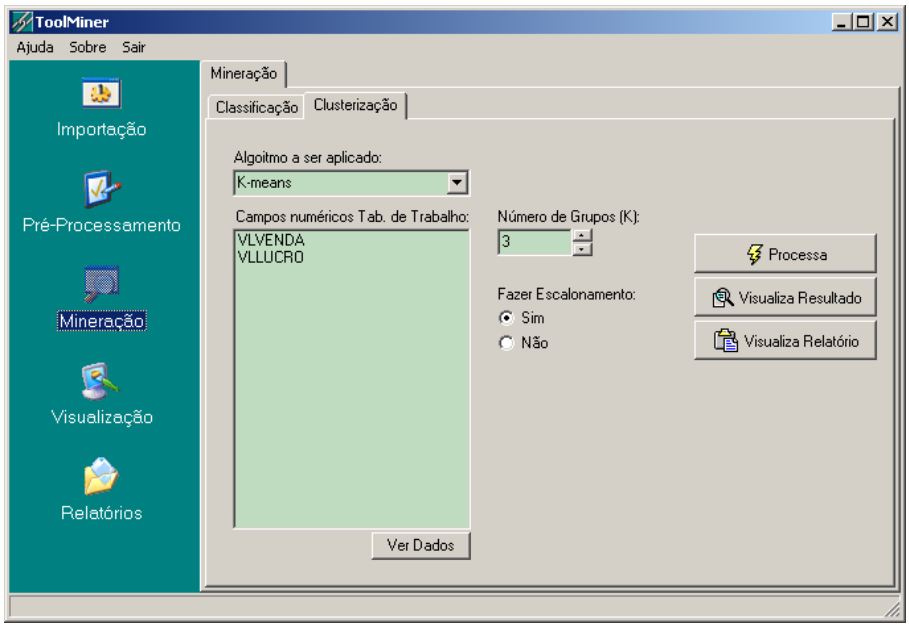

Figura 2. Tela de definição dos parâmetros para a execução do *K-means*

Concluído o processamento do algoritmo, visualizou-se os resultados em vários formatos distintos, definidos no módulo de visualização da ferramenta, o que possibilitou estabelecer as características e as relações existentes em cada *cluster*.

A Figura 3 exibe uma das visualizações da ferramenta, denominada *Visualização dos Resultados*, que permite verificar os resultados finais do algoritmo e que, neste caso, exibe os resultados da tarefa executada. No *grid Resultado do K-means* são exibidos os registros utilizados, com os valores originais dos atributos trabalhados, os valores escalonados destes atributos, os valores das distâncias euclidianas em relação a cada *cluster* e o *cluster* no qual cada registro foi classificado. No *grid Centróides* são exibidos os valores finais dos centróides de cada cluster, e no *grid Estatística dos Clusters*, o total e o percentual de registros em cada cluster.

Figura 3. Tela de *Visualização de Resultados* do algoritmo *K-means*

Através de outra opção de visualização, implementada na ferramenta, que mostra os intervalos de valores e as médias dos atributos utilizados pelo algoritmo (VLVENDA e VLLUCRO), dos registros já classificados em cada *cluster*, pode-se verificar como o algoritmo agrupou esses os valores. No *cluster K1*, os registros com os valores médios de venda e valores de lucro maior, no *cluster K2*, os valores baixos de venda e lucro menor e *no cluster K3*, os valores altos de venda e lucro negativo. Estas relações também podem ser visualizadas na opção *Visualização Gráfica* (Figura 4).

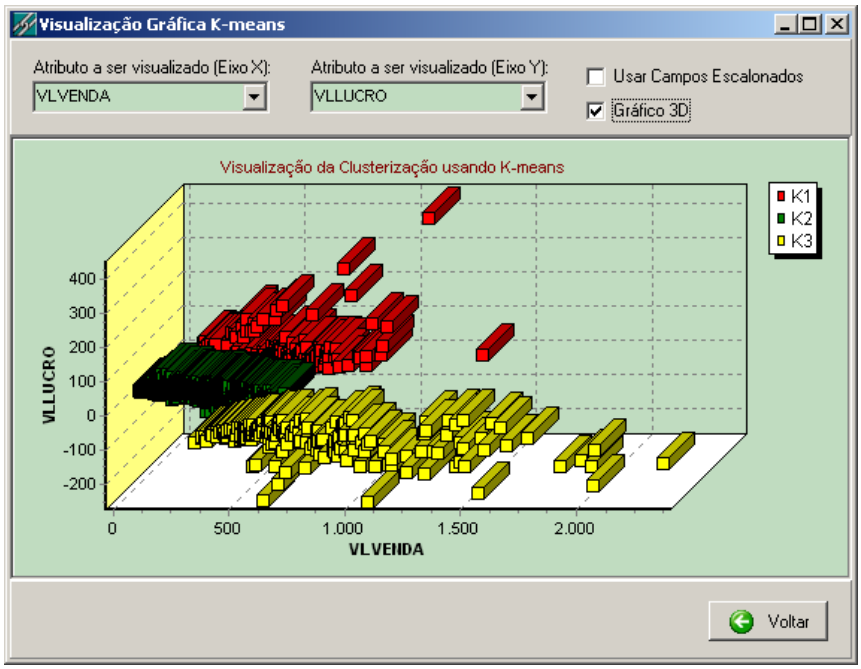

Figura 4. Tela *Visualização Gráfica do K-means*

Para definir ainda melhor as características de cada *cluster*, analisaram-se os valores de outros atributos presentes nos registros de cada *cluster*, como: linha de produto, sexo do cliente e idade do cliente, cujo resumo é mostrado na Tabela 1.

Tabela 1. Resumo das características dos *clusters*

K	% de registros	Valor médio de venda (R$)	Valor médio do lucro (R$)	Sexo do cliente	Idade do cliente	Linha de produto
K1	6,01%	539,94	51,97	Ambos	18-30	Móveis e Decoração - 4,9 %
K2	86,74%	130,82	0,57	Feminino	18-30	Telefonia Móvel – 19,74% Móveis e Decoração – 16,27% Bazar – 10,76%
K3	7,25%	890,93	-71,74	Ambos	18-30	Som e Imagem – 3,78% Eletrodomésticos – 2,89%

Validou-se com a empresa os conhecimentos gerados: a maioria das vendas (*cluster K2*) é de produtos com valores de venda e lucro baixos; a predominância de clientes na faixa etária de 18 a 30 anos não era de conhecimento da empresa; o *cluster K1* que apresenta os melhores lucros, em que predomina a linha de produtos *Móveis e Decoração*, é válido, pois essa é a linha de produtos que possui a maior margem de lucro; o *cluster K3* que apresenta uma média negativa no lucro, em que predominam os produtos das linhas *Som e Imagem* e *Eletrodomésticos*, é válido, pois são esses os produtos que possuem as menores margens de lucro. Assim, a maioria do conhecimento foi validado, comprovando a eficiência da tarefa e do algoritmo aplicados, porém não representaram novas descobertas, com a exceção da predominância, em todos os *clusters*, de clientes com idade entre 18 e 30 anos.

3.3 Tarefa de classificação de clientes por linha de produto

Aplicou-se esta tarefa para descobrir os perfis de clientes de cada linha de produto. Para isto, utilizaram-se os atributos: linha de produto (LINHAPROD), como atributo preditivo (meta); sexo do cliente (SEXO), estado civil do cliente (ESTCIVIL), salário do cliente (SALARIO_N_D) e idade do cliente (IDADE_N_D), como atributos descritivos. Os atributos SALARIO_N_D e IDADE_N_D são atributos resultantes das operações de discretização dos atributos SALARIO_N e IDADE_N. A Figura 5 exibe a tela inicial de execução do algoritmo *C4.5*, que define os atributos utilizados e o *critério de parada*.

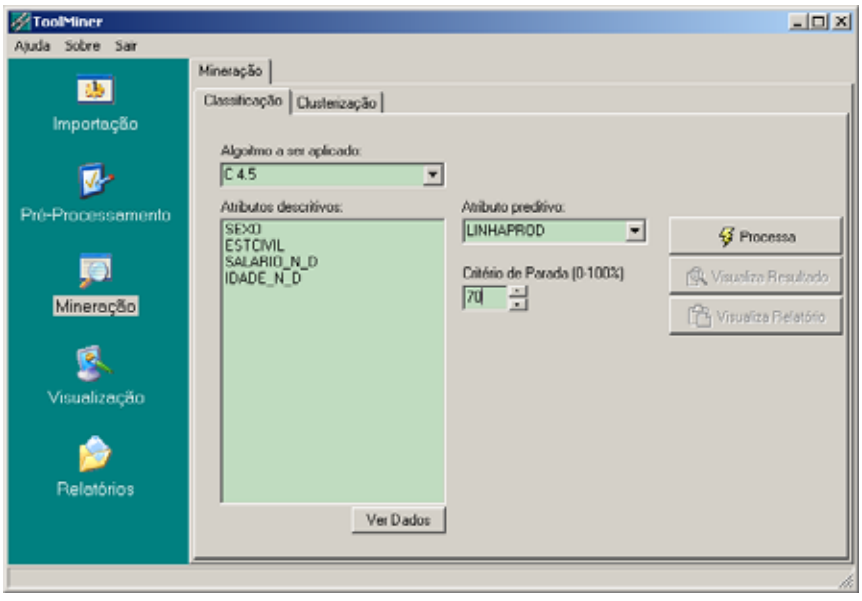

Figura 5. Tela de definição dos parâmetros para a execução do *C4.5*

O percentual de 70%, escolhido como *critério de parada* (Figura 5), foi o valor que apresentou os melhores resultados e não gerou uma árvore muito grande. Observou-se que um valor alto de *critério de parada* gera uma árvore grande e com regras pouco representativas; um valor bastante baixo acarreta na generalização da árvore para as regras mais representativas. Por esses motivos, para cada tipo de tarefa aplicada, deve-se testar valores distintos de *critério de parada*, até se encontrar um valor que gere uma árvore não muito grande e com regras bem representativas.

Após a execução do *C4.5*, analisaram-se os resultados do algoritmo, exibidos nas visualizações da árvore de decisão e das regras de classificação. Os conhecimentos gerados foram extraídos das regras mais representativas de cada linha de produto (Tabela 2).

Tabela 2. Regras mais representativas de cada linha de produto

SE SEXO = MASCULINO E SALARIO_N_D = MEDIO (501-1200) ENTÃO LINHAPROD = MOVEIS E DECORACAO (Q.1237 / R.6,35% / P.30,05%)
SE SEXO = FEMININO E ESTCIVIL = CASADO E IDADE_N_D = PLENA (31-50) E SALARIO_N_D = BAIXO (150-500) ENTÃO LINHAPROD = ELETRODOMESTICOS (Q.655 / R.3,37% / P.33,73%)
SE SEXO = MASCULINO E SALARIO_N_D = BAIXO (150-500) E IDADE_N_D = JOVEM (19-30) E ESTCIVIL = SOLTEIRO ENTÃO LINHAPROD = TELEFONIA MOVEL (Q.407 / R.2,09% / P.34,67%)

As validações feitas pela empresa dos conhecimentos extraídos das regras mais representativas são:

a) *clientes do sexo masculino e com salário médio compram móveis e decoração*: não era conhecido pela empresa;

b) *clientes do sexo feminino compram eletrodomésticos*: era conhecido; *mulheres casadas, com idade entre 31 e 50 anos e que possuem salário baixo compram eletrodomésticos*: não era conhecido;

c) *homens com salário baixo, jovens e solteiros compram celulares*: não era conhecido.

Analisando-se estas validações, verifica-se que a maioria dos conhecimentos gerados são novos para a empresa, o que significa que esta tarefa gerou novas descobertas, que poderão ser comprovadas através de outras amostras da base de dados, que possuam uma população maior de registros.

4. Conclusão

A utilização deste *data warehouse* para a aplicação de *data mining* apresentou como pontos positivos: a centralização dos dados em um único repositório, o que evitou a procura e a seleção nas tabelas dos sistemas transacionais, reduzindo o tempo de pré-processamento; as informações presentes no DW permitiram identificar tarefas aplicáveis, as quais geraram conhecimentos válidos e novas descobertas. Entretanto, foram

identificados pontos negativos como: a limitação da escolha das tarefas em função dos atributos existentes no DW, que neste caso se referiam a apenas um assunto de negócio; a necessidade da execução de algumas operações de limpeza na base de dados. Assim, comprovou-se que a grande vantagem de se utilizar um DW é a redução do tempo de pré-processamento do processo de DCBD.

Constatou-se que para extrair bons resultados na aplicação das tarefas, durante a execução do processo de DCBD, são necessárias várias execuções dos algoritmos de DM, combinando atributos distintos da base de dados e alterando os parâmetros destes algoritmos, como: o percentual do *critério de parada*, no *C4.5*, e o valor da variável *k*, no *K-means*. Após cada execução do algoritmo devem ser analisados os resultados obtidos, em conjunto com os analistas de negócio da empresa, para verificar o modelo que gera os melhores resultados.

A ferramenta desenvolvida e os algoritmos implementados comprovaram a sua eficiência através da resolução das tarefas propostas. A *ToolMiner* pode ser utilizada para a resolução de outras tarefas do mesmo tipo, em qualquer base de dados que esteja no formato nativo do SADIG ou em formato texto. Para a evolução da ferramenta, sugere-se: a integração direta com outros gerenciadores de banco de dados; a incorporação de mais algoritmos para resolver outros tipos de tarefas; a implementação de técnicas de discretização de dados e o desenvolvimento de novas formas de visualização dos resultados. Além disso, poderiam ser desenvolvidos novos algoritmos de *data mining*, otimizando os já existentes.

Referências

Engel, P. M. (2002) “Sistemas de Informação Inteligentes”, Material de Aula, PPGC–UFRGS.

Fayyad, U. *et al* (1996) “Advances in Knowledge Discovery and Data Mining”, Califórnia: American Association for Artificial Intelligence.

Feldens, M. A. (1996) “Descoberta de Conhecimento Aplicada à Detecção de Anomalias em Base de Dados”, Trabalho Individual, PPGC–UFRGS.

Halmenschlager, C. (2002) “Um Algoritmo de Indução de Árvores e Regras de Decisão”, Dissertação de Mestrado, PPGC-UFRGS.

Inmon, W. H. (1997) “Como Construir o Data Warehouse”, Rio de Janeiro: Campus.

Monash (2005) “K-means Clustering”, Faculty of Information Technology – Monash University: Australia”, http://www.csse.monash.edu.au/courseware/cse5230/assets/tutorials/clustering.pdf, March.

Quinlan, J. R. (1993) “C4.5: Programs for Machine Learning”, San Mateo: Morgan Kaufmann.

UCI (2005) “UCI Machine Learning Repository Content Summary”, Information and Computer Science - University of California: Irvine, http://www.ics.uci.edu/~mlearn/MLSummary.html, March.

Weka (2005) “Weka 3: Data Mining Software in Java - Collections of datasets”, The University of Waikato: New Zealand, http://www.cs.waikato.ac.nz/ml/weka, March.