UNIVERSIDADE DO EXTREMO SUL CATARINENSE

CURSO DE CIÊNCIA DA COMPUTAÇÃO

DÊNIS PIAZZA MARTINS

O ALGORITMO DE PARTICIONAMENTO *K-MEANS* NA TAREFA DE CLUSTERIZAÇÃO DA *SHELL* ORION *DATA MINING ENGINE*

CRICIÚMA, JULHO DE 2007

DÊNIS PIAZZA MARTINS

O ALGORITMO DE PARTICIONAMENTO *K-MEANS* NA TAREFA DE CLUSTERIZAÇÃO DA *SHELL* ORION *DATA MINING ENGINE*

Trabalho de Conclusão de Curso apresentado para obtenção do Grau de Bacharel em Ciência da Computação da Universidade do Extremo Sul Catarinense.

Orientadora: Profª. M.S.c. Merisandra Côrtes de Mattos

CRICIÚMA, JULHO DE 2007

A minha amada esposa, Lílian, a minha mãe, Gladez e minha irmã Carini, colegas de trabalho e amigos pelo apoio concedido até esse momento.

AGRADECIMENTOS

Agradeço a Deus e ao nosso Senhor Jesus Cristo pelas bênçãos concedidas em minha vida, por permitir que eu chega-se até aqui, estando Ele comigo nos momentos alegres e difíceis de minha vida, por isso, me mantive crente na resposta de Deus para prosseguir e vencer os obstáculos em busca dos meus sonhos.

Agradeço também:

A minha família e esposa que são benção para minha vida, me proporcionando amor, carinho e afeto, além de estarem sempre do meu lado me dando força nos momentos que precisei.

Aos colegas de trabalho e amigos pelo apoio e amizade, e compreensão, pois sempre que precisei tive o apoio deles.

A minha professora e orientadora Merisandra, que foi quem mais acreditou em meu potencial e não permitiu que eu desanima-se em momento algum, contribuindo e muito com sua experiência, conhecimento e amizade**.**

“Melhor é um punhado com tranqüilidade do
que ambas as mãos cheias com
trabalho e vão desejo”
(Eclesiastes 04:06).

RESUMO

A busca de conhecimento em base de dados, de maneira eficaz e inteligente, pode ser realizada por meio do processo de descoberta de conhecimento em bases de dados, que reúne vários passos e tarefas, tendo-se como uma de suas etapas a de *Data Mining*, que é responsável por extrair o conhecimento da base. Mediante isso, o Grupo de Pesquisa em Inteligência Computacional Aplicada do Curso de Ciência da Computação da Unesc, tem como projeto o desenvolvimento de uma *Shell* de *data mining*, denominada Orion *Data Mining Engine*, que está sendo implementada em Java e possibilita a integração via JDBC a qualquer banco de dados. Na realização desta pesquisa desenvolveu-se o módulo correspondente a tarefa de clusterização que é responsável por gerar grupos de dados, chamados de *clusters*, que devem possuir alguma relação entre si. O método aplicado na tarefa foi o algoritmo de particionamento *K-means*, que consiste em encontrar elementos centrais em uma base de dados e associá-los a outros próximos a ele em um mesmo grupo. Nos testes realizados na tarefa de clusterização pelo algoritmo *K-means*, foi utilizada uma base de dados na área da saúde, referente a prevalência de asma e rinite em adolescentes escolares do município de Criciúma, gerando-se satisfatoriamente os grupos referentes aos fatores mais evidentes detectados nesta base.

Palavras chaves: *Data Mining*; Clusterização; Métodos de Particionamento; Algoritmo *K-Means*; *Shell* Orion *Data Mining Engine*.

ABSTRACT

The search of knowledge in database, in efficient and intelligent way, can be carried through the process of discovery of knowledge in databases, which congregates some steps and tasks; one of these stages is the Mining Data, which is responsible for extracting the knowledge of the base. By means of this, the Group of Research in Applied Computational Intelligence of the Course of Computer Science of the Unesc has as project the development of a Shell of data mining, called Orion Data Mining Engine, which is being implemented in Java and makes possible the integration trough JDBC to any data base. In the accomplishment of this research the corresponding module was developed the task of clustering that is responsible for generating databases, called clusters, which must possess some relation between itself. The method applied in the task was the algorithm of partition K-means, which consists of finding central elements in a database and associating them to other at the same group. In the tests carried through the task of clustering for the K-means algorithm, was used a database in the health area, referring the prevalence of asthma and rhinitis in adolescents of the city of Criciúma, generating satisfactorily the referring groups to the detected factors in this base.

Keywords: Data Mining; clustering; Methods of partition; K-means Algorithm; Shell Orion Data Mining Engine.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

DATASUS	Departamento de Informática do Sistema Único de Saúde
DCBD	Descoberta de Conhecimento em Base de Dados
DM	*Data Mining*
DW	*Data Warehouse*
IA	Inteligência Artificial
JDBC	Java *Data Base Connectivity*
KDD	*Knowledge Discovery in Databases*
KDSE	*Knowledge Discovery Support Environment*
RNA	Redes Neurais Artificiais
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	*Structured Query Language*
PAM	*Particioning Around Medoid*s
TCC	Trabalho de Conclusão de Curso
UML	*Unified Modeling Language*
UFPA	Universidade Federal do Pará
UNESC	Universidade do Extremo Sul Catarinense

SUMÁRIO

INTRODUÇÃO

A grande quantidade de informações nos bancos de dados informatizados das organizações pode esconder conhecimentos úteis para a tomada de decisão. O aumento no volume dos dados, associado à crescente demanda por conhecimento novo voltado para decisões estratégicas, tem provocado o interesse na descoberta de relações e novos padrões nas bases de dados.

O processo de Descoberta de Conhecimento em Base de Dados (DCBD) para ser realizado necessita de ferramentas computacionais de *data mining*, pois devido ao volume de dados estes não seriam facilmente descobertos de forma manual. Assim, estas soluções têm sido muito utilizadas em universidades e organizações, porém acarretam um alto custo, devido a sua complexidade e também por existirem poucas ferramentas gratuitas disponíveis, com exceção de alguns projetos que vêm sendo desenvolvidos em algumas universidades.

Mediante isso, o Grupo de Pesquisa em Inteligência Computacional Aplicada do Curso de Ciência da Computação da Unesc, desenvolve um projeto que consiste em uma *shell* de *data mining*, denominada Orion *Data Mining Engine*. Esta ferramenta possibilita a conexão com vários bancos de dados e possui uma interface simples, assim como a compreensão dos resultados.

O desenvolvimento da *Shell* Orion compreende a aplicação de diferentes tarefas e métodos de *data mining*. Atualmente, tem-se as tarefas de associação, classificação e clusterização, sendo que esta pesquisa compreende a implementação do módulo referente a tarefa de clusterização pelo algoritmo de particionamento *K-means*.

A tarefa de clusterização tem por objetivo criar grupos de elementos em uma base de dados, podendo utilizar para isso o algoritmo de particionamento *K-Means*,

responsável por selecionar os elementos centrais para cada grupo a ser gerado e realizar os cálculos de medidas de distância entre todos eles, inserindo os que possuem medidas próximas em um mesmo grupo.

1.1 OBJETIVO GERAL

Implementar na *Shell* Orion *Data Mining Engine* a tarefa de clusterização pelo método de particionamento *K-means***.**

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa são:

a) compreender os conceitos de inteligência artificial e *data mining*;
b) entender a tarefa de clusterização e o método de particionamento pelo algoritmo *K-means*;
c) demonstrar o funcionamento do algoritmo de particionamento *K-means*;
d) aplicar o algoritmo *K-means* para particionamento de dados;
e) testar a tarefa de clusterização na *Shell* Orion *Data Mining Engine* pelo algoritmo *K-Means* por meio de uma base de dados referente a saúde.

1.3 JUSTIFICATIVA

A *Shell* Orion *Data Mining Engine* em desenvolvimento pelo Grupo de Pesquisa em Inteligência Computacional Aplicada da Unesc, pretende contribuir com

diferentes instituições por meio da disponibilização de uma ferramenta de *data mining* gratuita e em português, podendo-se assim, facilitar e incentivar pesquisas em diversas áreas de conhecimento.

Atualmente a *Shell* Orion *Data Mining Engine*, é composta de dois módulos (associação e classificação), sendo que com o desenvolvimento da tarefa de clusterização por meio do algoritmo de particionamento *K-means*, uma nova funcionalidade, no que se refere a tarefa de *data mining*, será inserida.

Harrison (1998) define a tarefa de clusterização como a construção de modelos que encontram registros de dados semelhantes e os reúnem em grupos (*clusters*). O principal benefício gerado por essa tarefa é encontrar *clusters* de registros de dados com características próximas, originando assim conhecimento para o usuário.

Os resultados de uma tarefa de clusterização podem ser obtidos de duas maneiras diferentes (SERRA, 2002; GOLDSCHMIDT; PASSOS, 2005):

a) sumário da base de dados;
b) dados de entrada para outras tarefas, como classificação e associação, pois o *cluster* é um grupo menor e de manuseio mais fácil por parte dos algoritmos.

A tarefa de clusterização foi implementada e aplicada nesta pesquisa por meio do algoritmo de particionamento *K-means*, pois de acordo com Goldschimidt e Passos (2005) esse algoritmo é bastante popular para resolver o problema da clusterização, sua estrutura é simples, seu processamento é rápido e capaz de obter bons resultados.

O algoritmo K-means seleciona os elementos centrais de cada *cluster* a ser gerado, chamado de centróide. Posteriormente, calcula-se a distância de cada registro da

base de dados em relação ao centróide, sendo inserido o registro no *cluster* que possui a menor distância.

1.4 ESTRUTURA DO TRABALHO

Este trabalho é composto por sete capítulos, tendo-se no primeiro uma contextualização do tema proposto, bem como os objetivos e a justificativa para realização desta pesquisa.

Nos Capítulos 2 e 3 são abordados os conceitos fundamentais de descoberta de conhecimento em bases de dados e de *data mining*, indispensáveis para o entendimento da pesquisa.

O Capítulo 4 aborda com maior ênfase a tarefa de clusterização utilizada para resolução do problema, sendo descritas suas características, etapas e alguns dos métodos que utiliza.

O método de particionamento *K-means* aplicado para o desenvolvimento do módulo de clusterização da *Shell* Orion *Data Mining Engine* é descrito no Capítulo 5, enquanto no Capítulo 6 são apresentados alguns exemplos da sua utilização.

O Capítulo 7 apresenta a pesquisa realizada, descrevendo-se os passos de todo o processo de desenvolvimento, bem como a discussão dos resultados obtidos.

E por fim, tem-se a conclusão, onde se encontram algumas sugestões para trabalhos futuros.

2 DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS

A Descoberta de Conhecimento em Base de Dados (DCBD) ou *Knowledge Discovery in Databases* (KDD) foi desenvolvida com a finalidade de buscar conhecimento em grandes bases de dados que não pode ser adequadamente recuperado ou mesmo visualizada, devido a limitações dos sistemas gerenciadores de banco de dados atuais (REZENDE, 2002).

O DCBD pode ser definido como a análise inteligente dos dados, consistindo em produzir conhecimento a partir de uma base de dados, a fim de encontrar informações desejadas. Nos diferentes segmentos da sociedade, as instituições têm buscado na tecnologia recursos que agreguem valor aos seus negócios, seja agilizando operações, suportando ambientes ou viabilizando inovações. Diariamente, pessoas e instituições disponibilizam dados oriundos de tarefas cotidianas a estas plataformas tecnológicas por meio de simples atividades como compras no supermercado do bairro ou operações bancárias (SILVA, 2002).

Os sistemas de computação participam da vida das pessoas de forma cada vez mais próxima e constante. Não obstante, institutos científicos, indústrias, corporações e governos acumulam volumes gigantescos de dados, impulsionados também pela versatilidade e alcance proporcionados pela Internet (SILVA, 2002).

Devido a essas características, todo o processo de DCBD depende de uma geração de ferramentas e técnicas de análise de dados, envolvendo etapas de: definição do problema; pré-processamento de dados; *data mining* e pós-processamento.

2.1 O PROCESSO DE DESCOBERTA DE CONHECIMENTO EM BASES DE DADOS

No processo de DCBD cada fase pode possuir uma interação com as demais, ou seja, os resultados produzidos em uma etapa podem ser utilizados para melhorar sua performance nas fases seguintes. Identifica-se dessa forma que o processo de DCBD é iterativo, buscando sempre aprimorar os resultados a cada iteração. Ao executar o processo de DCBD o usuário pode analisar as informações geradas a cada fase e acrescentar seu conhecimento como analista de dados para obter cada vez mais resultados satisfatórios (LOPES, 1999).

A Figura 1 representa de forma hierárquica uma visão sistemática do processo de DCBD, nesse diagrama são incluídas as principais fases do processo de DCBD: pré-processamento; *Data Mining* (DM) e pós-processamento.

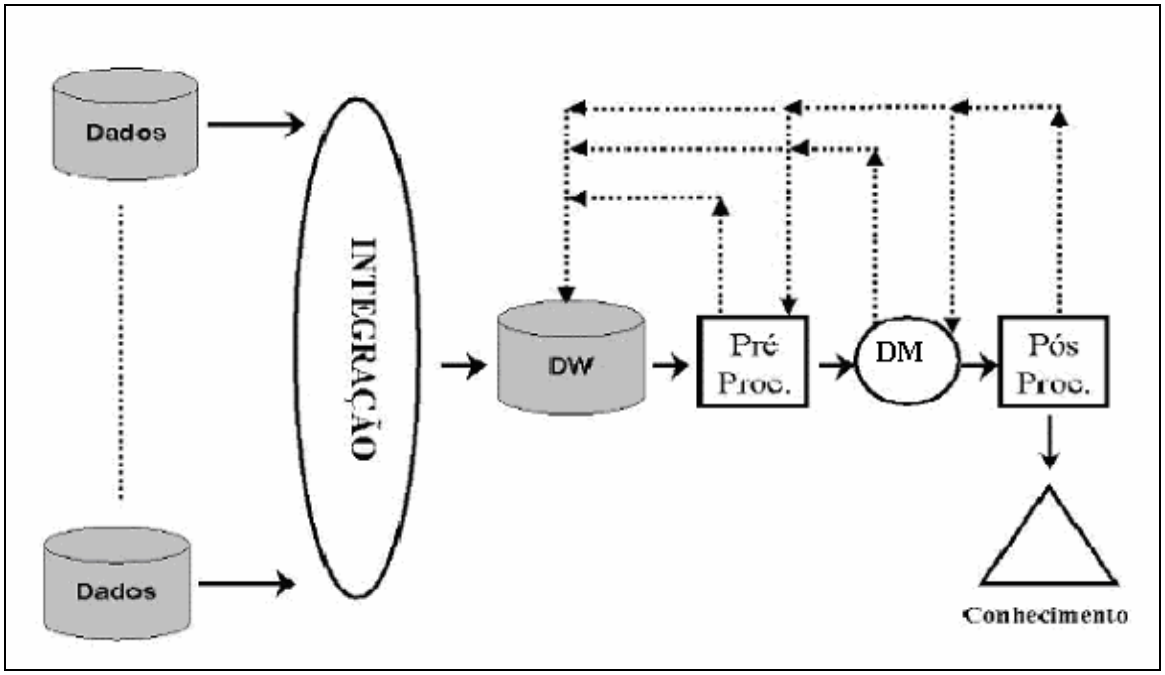

Figura 1. Estrutura de um DCBD
Fonte: LOPES,C. (1999).

O processo de DCBD é constituído por diferentes etapas, tendo-se também vários tipos de bases e bancos de dados que podem ser integrados, constituindo ou não um *Data Warehouse* (DW). Dentre as etapas para sua execução, primeiramente deve-se definir o problema, pré-processar os dados a fim de eliminar ruídos e também convertê-los quando necessário, para então serem aplicadas as tarefas de *data mining.* Finalmente, tem-se o pós-processamento que mostra os resultados obtidos, podendo-se então gerar conhecimento para o usuário de DCBD.

2.1.1 Definição e Fundamentação do Problema

A definição do problema é fundamental para o processo de DCBD, sendo necessário o conhecimento do problema existente e a definição dos objetivos. Para isso, deve-se realizar uma iteração com o solicitador da tarefa de modo que seja exposto tudo o que se relaciona com o problema. Posteriormente, pode-se fixar metas para os objetivos do DCBD (REZENDE, 2002).

2.1.2 Pré-Processamento de Dados

O pré-processamento é uma fase de preparação dos dados, que diz respeito ao entendimento da área de aplicação e a definição do conjunto de dados a serem submetidos à tarefa de DCBD, sendo composto pelas seguintes etapas (LACERDA; SOUZA, 2004):

a) **limpeza dos dados**: verifica a consistência das informações, correção de possíveis erros, eliminação de valores redundantes, remoção de dados duplicados e/ou corrompidos;

b) **seleção de dados**: nesta etapa os dados relevantes para a aplicação do *data mining* são identificados e reunidos formando um subconjunto da base de dados, com isso tem-se uma otimização de tempo, visto que ele apenas trabalhará com um subconjunto de atributos, diminuindo dessa forma o seu espaço de busca;

c) **codificação de dados**: transformação ou consolidação dos dados em uma forma apropriada para o *data mining*, podendo-se converter valores quantitativos em categóricos.

2.1.3 *Data Mining*

Data Mining é uma atividade multidisciplinar que envolve diversas áreas e utiliza-se de ferramentas para a descoberta de conhecimento em grandes massas de dados, sendo um campo de pesquisa que envolve estatística, aprendizado de máquina, banco de dados e inteligência artificial (HAN, KAMBER, 2000).

As tarefas de *data mining* fazem uso de algoritmos que são capazes de extrair eficientemente conhecimento da base de dados. Assim, pode-se dizer que o *data mining* é a fase do DCBD que transforma informações em conhecimento.

2.1.4 Pós-processamento de Dados

Ao término do processo de *data mining*, tem-se a saída de dados, essa fase envolve a interpretação do conhecimento descoberto ou algum processamento deste. Essa fase tem basicamente duas funcionalidades (SILVA, 2002):

a) **avaliação dos padrões**: onde são identificados os padrões realmente interessantes, que representam conhecimento baseado em alguma medida de interesse;

b) **apresentação do conhecimento**: as técnicas de visualização e representação do conhecimento são usadas com a finalidade de apresentar o resultado do *data mining* ao usuário.

Neste capítulo foi apresentado o processo de DCDB e as suas etapas, sendo a de *data mining* a responsável pela efetiva busca do conhecimento. Assim, esta será abordada com mais ênfase a seguir, pois se constitui na base dessa pesquisa.

3 DATA MINING

O *data mining* é uma etapa do DCBD que utiliza dados históricos para aprendizagem objetivando realizar alguma tarefa em particular, que possui como meta responder uma pergunta de interesse do usuário. Portanto, para execução do *data mining* além das tarefas são necessários métodos que as implementem, compostos por diferentes algoritmos (FAYYAD,1996).

Dessa forma, o *data mining* emprega uma série de tarefas pré-determinadas que são utilizadas para descoberta do conhecimento em bases de dados. Porém, sua função não está limitada em analisar e descobrir novos conhecimentos, pois por exemplo, no caso de uma empresa é preciso incorporá-lo e torná-lo uma etapa natural das atividades em uma organização, para que a mesma consiga produzir os resultados esperados e necessários (CARVALHO, 2002).

O objetivo da utilização do *data mining* em uma organização é permitir que a mesma possa visualizar e executar operações de improviso, quando necessárias, nas mais diversas áreas como marketing, vendas e suporte ao cliente, sendo isso somente possível pelo fato da empresa conhecer bem seus clientes (AMARAL, 2001).

De forma geral, pode-se afirmar que o *data mining* tem como objetivo melhorar a qualidade e eficiência no momento da tomada de decisão, pois com esta metodologia é possível complementar ou substituir outras ferramentas de apoio a decisões como, por exemplo: análises estatísticas e relatórios (LACERDA; SOUZA, 2004).

Assim, as organizações necessitam de processos como o de *data mining* que são capazes de proporcionar conhecimentos novos para auxiliar na tomada de decisão. Interessa saber, por exemplo, o que estão pensando seus clientes, como estão os

concorrentes, variações de estoques, mercado financeiro, entre outros. O *data mining* permite percorrer e encontrar padrões em grandes bases de dados, o que pode beneficiar as instituições com a identificação do comportamento de consumidores, a compra de suprimentos ou ainda administrar área comercial e financeira.

A técnica de *data mining* pode ser aplicada em diferentes áreas, como por exemplo:

a) nas finanças muitas empresas estão preocupadas com o gerenciamento de seus investimentos visando encontrar maneiras de aumentar seus lucros. Nesse caso, as ferramentas de *data mining* têm a função de realizar previsão de mercado, onde com base em dados atuais prevê-se a tendência de mercado em curto prazo. Ainda na área financeira as organizações bancárias usam o *data mining* para descobrir os tipos de clientes que possuem, podendo então oferecer, por exemplo, serviços como empréstimos, para determinado perfil de cliente, além de se ter a possibilidade de identificar fraudes (LACERDA; SOUZA, 2004);

b) no conhecimento científico o *data mining* tem sido utilizado pelas instituições na busca de conhecimentos úteis, que se encontram implícitos nos grandes volumes de dados armazenados. A NASA, por exemplo, utiliza esta técnica em pesquisas de mudanças climáticas (ULYSSÉA, 1999);

c) governos de países mais desenvolvidos têm utilizado o *data mining*, na maioria dos casos para aprimorar a cobrança de impostos, como exemplo pode-se citar o caso do Serviço de Imposto de Renda Americano que já tem desenvolvido um sistema para detectar fraudes (ULYSSÉA, 1999);

d) na Medicina a aplicação do *data mining* está relacionada ao auxílio para profissionais, principalmente no que se diz respeito a sistemas que possam: identificar a melhor forma de tratamento dos pacientes, estimar o tempo que um paciente resistirá ao tratamento, entre outros. A exemplo pode-se selecionar apenas os perfis cujas condições médicas sejam de interesse, ou seja, aqueles para os quais existe um procedimento conhecido para melhorar o estado de saúde e reduzir custos (LACERDA; SOUZA, 2004);

e) na área de marketing deve-se compreender e canalizar as necessidades individuais dos clientes, portanto o *data mining* pode identificar preferências e os padrões de compra dos consumidores. Dessa forma, as empresas, podem direcionar melhor os produtos e ofertas para os seus clientes (LACERDA; SOUZA, 2004).

Visando alcançar objetivos como os listados acima, as tarefas de *data mining* precisam de uma metodologia para ser seguida, tema esse abordado no próximo item.

3.1 METODOLOGIA DE DATA MINING

O *data mining* pode ser aplicado em uma base de dados por meio de três diferentes formas, denominadas metodologias (CARVALHO, 2002):

a) **descoberta não supervisionada**: caracteriza-se por buscar nos dados relações novas e escondidas, que sem o auxílio de ferramentas computacionais não seriam localizadas com facilidade;

b) **testagem de hipótese**: é implementada quando se possui alguma hipótese sobre uma relação, podendo então definir uma condição e verificar sua confirmação;

c) **modelagem de dados**: é executada quando se possui um nível de conhecimento maior da área e da relação que se deseja estudar, ou seja, constrói-se um modelo matemático com base no conhecimento que já existe.

A Figura 2 demonstra a aplicação das três metodologias e o nível de conhecimento adquirido com cada uma delas.

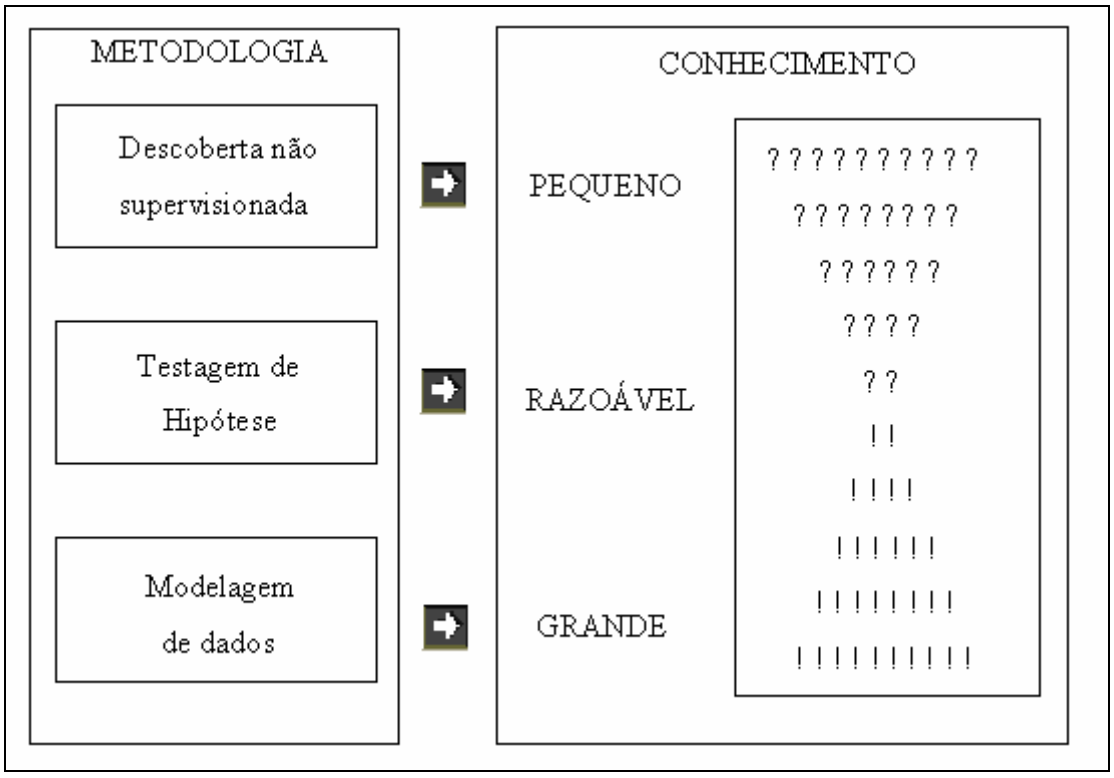

Figura 2. Metodologia em *data mining*
Fonte : CARVALHO, L. (2002).

Observando a Figura 2, verifica-se que a modelagem de dados resulta em mais conhecimento do que a descoberta supervisionada, com isso pode-se concluir que a metodologia é o pré-requisito necessário para a aplicação de uma tarefa de *data mining*.

3.2 TAREFAS DE *DATA MINING*

Segundo Reategui (2002) as tarefas de *data mining* mais utilizadas são associação, classificação, estimativa, previsão de séries temporais e clusterização. Cada tarefa tem suas próprias vantagens e desvantagens, do mesmo modo que nenhuma ferramenta consegue atender todas as necessidades em todas as aplicações.

As tarefas de *data mining* são divididas em dois grupos (REATEGUI, 2002):

a) **tarefas de descoberta direta de conhecimento**: são orientadas por objetivos, constituindo-se na seleção de um campo alvo e solicitação ao sistema para estimá-lo, classificá-lo ou prevê-lo;

b) **tarefas de descoberta indireta de conhecimento**: não há campo alvo nesse caso, simplesmente pergunta-se ao sistema como identificar padrões significativos nos dados, sendo a clusterização uma tarefa de descoberta indireta de conhecimento.

As tarefas de associação, classificação, estimativa e previsão são classificadas como tarefas de descoberta direta de conhecimento e apenas a tarefa de clusterização faz parte das tarefas de descoberta indireta de conhecimento (REATEGUI, 2002). A seguir são descritas as principais tarefas de *data mining*.

3.2.1 Classificação

O ser humano está sempre classificando, criando classes de relação e atribuindo a elas uma forma diferente de tratamento. Como no mundo físico nada é exatamente igual por mais semelhança que exista, para se criar uma classe é preciso ser

flexível as exigências de igualdade e permitir que detalhes sejam desprezados e somente as características principais sejam observadas (CARVALHO, 2002).

A tarefa de classificar normalmente exige a comparação de um dado com outros dados ou objetos que supostamente pertençam a classes anteriormente definida. Na realização desta comparação, utiliza-se uma função de medida para calcular a distância entre os objetos (CARVALHO, 2002).

Segundo Harrison (1998) as tarefas de classificação podem ser utilizadas em situação como:

a) atribuir palavras-chaves a artigos jornalísticos;
b) classificar pedidos de créditos como de baixo, médio e alto risco;
c) esclarecer pedidos de seguro fraudulentos.

3.2.2 Estimativa

Segundo Harrison (1998) essa tarefa é usada para atribuir um determinado valor a uma variável. Baseando-se em dados conhecidos do seu passado ou em dados semelhantes sobre os quais se tem conhecimento, uma fórmula é criada de modo a possibilitar que valores desconhecidos sejam estimados.

A estimativa trabalha com resultados contínuos, tendo-se algum dado de entrada ela é utilizada para estipular um valor a uma variável contínua desconhecida, tal como renda, altura ou limite de cartão de crédito. Alguns exemplos dessa tarefa são: estimar o número de filhos de uma família; estimar a renda total de uma família, entre outros (HARRISON, 1998).

3.2.3 Previsão de Séries Temporais

Segundo Bartolomeu (2002) a tarefa de previsão é o mesmo que classificação ou estimativa, exceto pelo fato de que os registros são classificados de acordo com algum comportamento futuro previsto ou valor estimado, abaixo se tem uma relação de exemplos das tarefas de previsão:

a) previsão da quantia de dinheiro que um cliente utiliza caso seja oferecido a ele um certo limite de crédito;
b) previsão de quais clientes abandonarão os serviços da empresa nos próximos meses;
c) previsão de quais clientes usariam um serviço extra.

3.2.4 Associação

O objetivo da tarefa de associação é encontrar em uma base de dados possíveis tendências que ajudem a compreender padrões. Os algoritmos de associação buscam encontrar relações entre os vários registros existentes na base de dados, verificando os eventos que ocorrem simultaneamente e possibilitando o entendimento de novos modelos, com isso, pode-se atingir melhores resultados (SERRA, 2002).

A associação pode ser utilizada, por exemplo, para analisar a base de dados de terminais de pontos de vendas de um supermercado a fim de descobrir os produtos que são adquiridos juntos. A partir disso, os negociantes podem redefinir a disposição dos mesmos na loja, fazendo com que a venda de um determinado produto induza a comercialização de outro (SERRA, 2002).

3.2.5 Clusterização

Clusterização é a tarefa de *data mining* capaz de realizar agrupamentos de conjuntos físicos ou abstratos de objetos em grupos similares (*cluster*). O *cluster* é uma coleção de objetos de dados que são similares entre si e diferentes dos elementos de outros *clusters*. A forma utilizada para a sua identificação consiste em uma função de similaridade ou distância (HAN, KAMBER, 2000).

A tarefa de clusterização pode servir como única ferramenta de *data mining* ou como um processo de pré-processamento para outros algoritmos agirem nos *clusters* gerados. Esta tarefa pode ser utilizada para aplicações de vendas; reconhecimento de padrões; estudos biológicos e agrupamento de documentos (HAN, KAMBER, 2000). A tarefa de clusterização, tema desta pesquisa, será descrita no próximo capítulo.

4 CLUSTERIZAÇÃO

A tarefa de *clusterização* ou agrupamento tem como princípio básico a reunião de registros que possuam similaridades em uma base dados, particionando-os em subconjuntos, denominados de *clusters.* Dessa forma, os registros pertencentes a um mesmo *cluster* possuem similaridades entre si e, ao mesmo tempo, os objetos pertencentes a *clusters* diferentes apresentem alta dissimilaridade (GOLDSCHMIDT, PASSOS, 2005).

A *clusterização* pode ser definida como uma das tarefas básicas de *data mining* auxiliando o usuário a realizar agrupamentos naturais de registros em um conjunto de dados. A Figura 3 representa o fluxo da tarefa de clusterização, onde os elementos que possuem alguma similaridade são inseridos em um mesmo grupo.

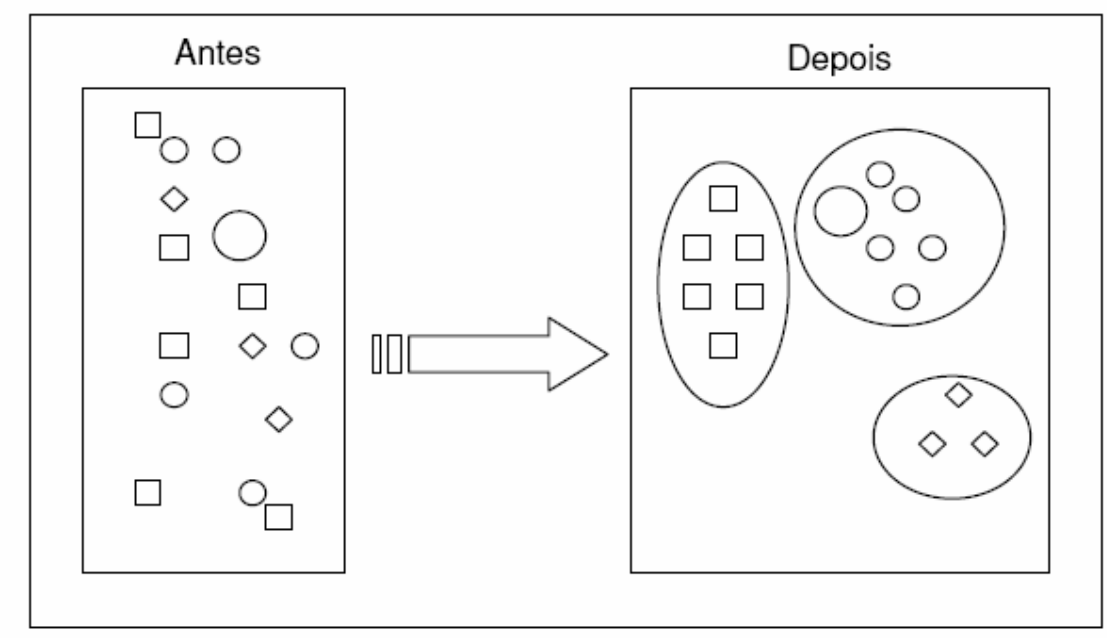

Figura 3. Fluxo da tarefa da clusterização
Fonte: SERRA, L. (2002)

Em *data mining* pode-se definir então que a função básica da clusterização é agrupar um conjunto de objetos em subconjuntos, observando para isso características conforme critérios apropriados. Estes critérios são estabelecidos por ferramentas de *data*

mining e as funções podem produzir descrições implícitas e explícitas (HAN; KAMBER, 2000).

Os métodos de clusterização podem ser caracterizados como qualquer procedimento estatístico, que utilizando um conjunto finito de informações, une os objetos em grupos restritos e homogêneos, permitindo a geração de estruturas agregadas significativas, ou seja, grupos com informações potenciais (REZENDE, 2002).

Dessa forma, pode-se observar então a existência de dois critérios básicos adotados na clusterização para constituição dos *clusters* (SILVA, 2002):

a) **homogeneidade:** refere-se a registros pertencentes a um mesmo *cluster*, que devem ser tão similares quanto possível;
b) **separação*:*** onde os elementos pertencentes a um *cluster* devem diferir significativamente dos demais.

Entre as vantagens oferecidas pelos métodos de clusterização, pode-se citar a geração de padrões ou mesmo grupos inesperados em um conjunto de dados, precisando para isso que os rótulos sejam automaticamente identificados, processo este diferente dos métodos de classificação, onde os rótulos são predefinidos. A clusterização identifica automaticamente os rótulos maximizando a similaridade *intracluster* e minimizando a *intercluster*. Por essa razão, esta tarefa é também denominada indução não supervisionada (GOLDSCHMIDT, PASSOS, 2005).

O aprendizado não supervisionado ou auto-organizável característico da clusterização é condicionado pelo fato de não existir um especialista que lhe indique o que cada padrão representa. Basicamente, a previsão é feita por meio da representação interna montada pelo sistema utilizando um conjunto de dados de entrada (PIMENTEL, FRANÇA, OMAR, 2003).

A análise de *clusters* envolve, portanto, a organização de um conjunto de padrões. Na maioria dos casos os resultados são representados na forma de vetores de atributos ou, conforme mostra a Figura 4, como pontos em um espaço multidimensional que são reunidos de acordo com as suas semelhanças, baseando-se em alguma medida de similaridade (GOLDSCHMIDT; PASSOS, 2005).

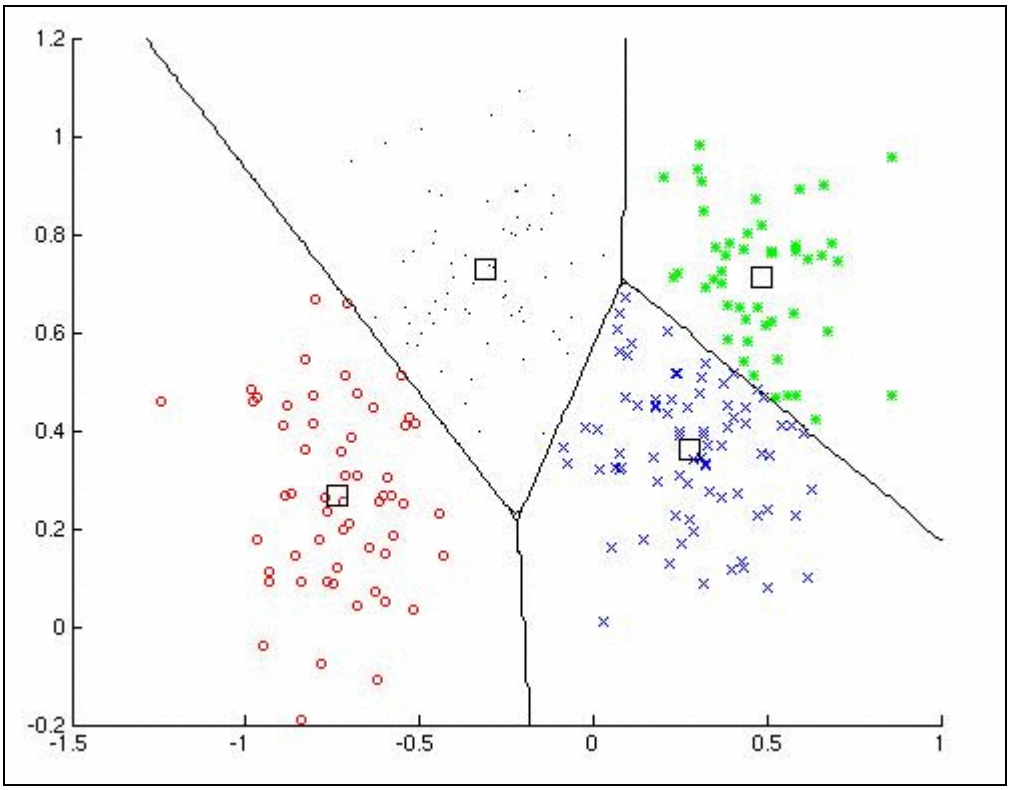

Figura 4. Gráfico da tarefa de clusterização
Fonte: REZENDE, S. (2002)

O sucesso da aplicação da tarefa de clusterização está diretamente relacionado a uma série de parâmetros como (NEVES; FREITAS; CÂMARA, 2001):

a) **escolha de atributos:** definição de quais elementos de uma base de dados serão clusterizados;
b) **medidas de similaridade:** refere-se a forma matemática usada pelo algoritmo de clusterização para calcular a distância entre os elementos;
c) **critérios de agrupamento:** composição da fórmula para definir os elementos centrais de cada grupo;

d) **escolha do algoritmo:** seleção do algoritmo responsável por atribuir aos elementos aos grupos;

e) **Número de *clusters*:** parâmetro informado pelo usuário na ferramenta de *data mining* que define quantos clusters serão gerados.

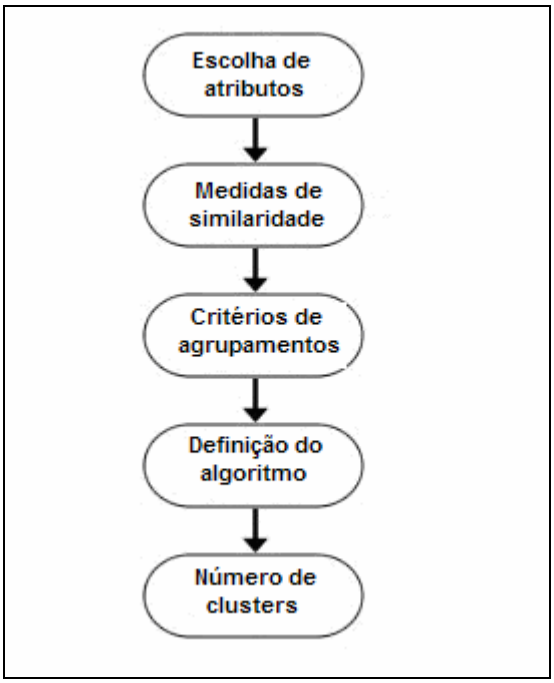

Figura 5. Fluxograma dos algoritmos de clusterização

Baseando-se na definição do número de *clusters* pelo usuário, os registros são separados em grupos de forma que os mais similares pertençam ao mesmo *cluster* e os diferentes sejam colocados em grupos distintos. Assim, é possível fazer uma análise dos dados que compõem cada um deles, identificando as características comuns e criando um rótulo que represente cada grupo (GOLDSCHMIDT, PASSOS, 2005).

A medida de similaridade entre dois elementos consiste na distância entre eles. Neste caso, quanto menor for a distância maior é a similaridade entre eles. As medidas de distância mais utilizadas e conhecidas são *euclidiana* e *city-block*, abordadas no Capítulo 5 (OCHI, DIAS, SOARES, 2004).

Após definir a medida de similaridade, o passo seguinte da tarefa de clusterização é encontrar os elementos centrais (centróides) de cada *cluster* de tal forma

que a distância entre os pontos (registros da base de dados) e o centróide do *cluster* mais próximo seja minimizada. Portanto, para que os algoritmos de clusterização possam efetuar sua tarefa é necessário que sejam utilizadas estruturas de dados capazes de armazenar os objetos a serem processados ou as informações sobre as relações entre estes (GOLDSCHMIDT, PASSOS, 2005).

Segundo Han e Kamber (2000) os algoritmos de clusterização trabalham com dados armazenados na memória principal e normalmente utilizam uma das seguintes estruturas de dados no seu processamento:

a) **matriz de dados:** as linhas representam um dos objetos a serem clusterizados e as colunas, os atributos ou características de cada objeto;

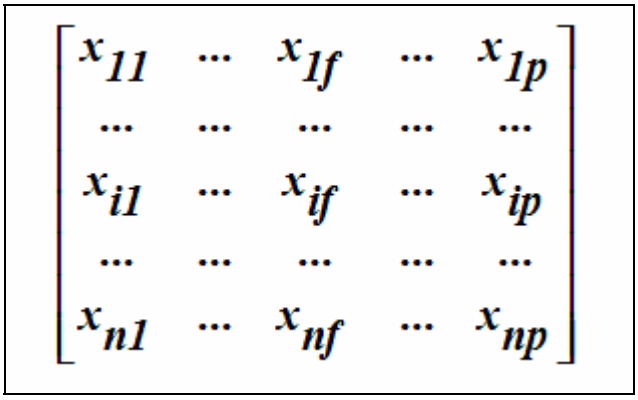

Figura 6. Matriz de dados

b) **matriz de similaridade:** cada elemento representa a distância entre os pares de objetos. Visto que a distância entre o objeto *i* e *j* é igual a do objeto *j* em relação a *i*, não é necessário armazenar todas as distâncias.

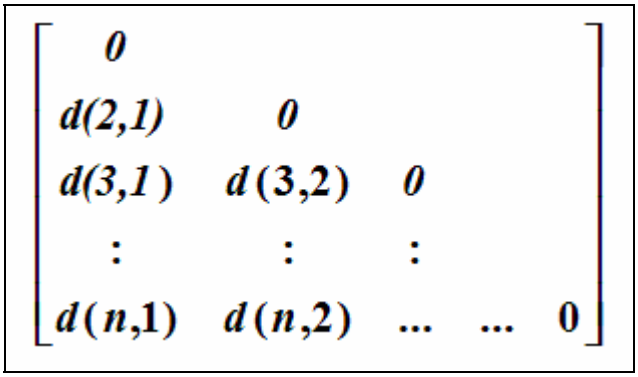

Figura 7. Matriz de similaridade

O ponto *d(i,j)* representa a distância ou similaridade entre o objeto *i* e o *j*. Como as medidas de similaridade expressam o conceito de distância, estas são sempre números positivos. Quanto mais próximo de zero for *d(i,j)*, mais similares serão os objetos.

Na tarefa de clusterização os algoritmos após receberem a matriz de dados geram uma matriz de similaridade antes de iniciar o processo de *data mining*, obedecendo a uma seqüência lógica (HAN; KAMBER, 2000):

a) **matriz de dados:** armazena todos os dados a serem analisados;
b) **formação dos centróides:** encontra os objetos centrais que serão usados como parâmetro para gerar os *clusters*;
c) **medidas de similaridade *d(i,j)*:** calcula a similaridade entre objetos;
d) **matriz de similaridade:** armazena a o valor da distância de um objeto, em relação ao elemento central;
e) **matriz de *clusters*:** responsável por armazenar os índices dos objetos e os respectivos *cluster* a que pertencem.

Pode-se definir então que um problema de clusterização é a geração de *clusters*, onde em alguns casos a medida de distância, não pode ou não é conveniente ser utilizada como medida de similaridade, um exemplo destes casos e quando os valores dos atributos não são escalares, ou seja, os atributos possuem valores muito diferentes um do outro e sem qualquer tipo de relação. Considerando-se como exemplo, um problema de clusterização que envolve atributos sexo e endereço, onde tem-se valores extremante diferentes para cada atributo e ainda a falta de relação entre sexo e endereço, nestes casos seriam necessárias informações originadas de outros atributos que demonstrem o grau de similaridade entre as instâncias da base de dados (OCHI; DIAS; SOARES, 2004).

Segundo Goldschmidt e Passos (2005) os resultados obtidos com a tarefa de clusterização devem atender a alguns requisitos:

a) descobrir *clusters* com forma arbitrária;
b) identificar *clusters* de tamanhos variados;
c) aceitar os diversos tipos de variáveis possíveis;
d) ser sensível a ordem de apresentação dos objetos;
e) trabalhar com objetos que possuam qualquer número de atributos;
f) ser escalável para lidar com qualquer quantidade de objetos;
g) fornecer resultados interpretáveis e utilizáveis;
h) ser robusto na presença de ruídos;
i) exibir o mínimo de conhecimento para determinar parâmetros de entrada;
j) aceitar restrições;
k) encontrar o número adequado de *clusters*.

Nenhum método de clusterização, atualmente, atende a todos esses requisitos adequadamente, embora realize-se um trabalho considerável para atender a cada aspecto separadamente. Os métodos de clusterização mais conhecidos e utilizados são os hierárquicos e de particionamento (GOLDSCHMIDT, PASSOS, 2005).

4.1 MÉTODOS HIERÁRQUICO

Os métodos hierárquicos são caracterizados dessa forma porque permitem implementar vários níveis de agrupamento, aglomerativos e divisivos, gerando uma hierarquia de *clusters*, normalmente representada por meio de uma estrutura em árvore (OCHI; DIAS; SOARES, 2004).

Nos métodos hierárquicos aglomerativos cada objeto é um *cluster* e a cada passo do procedimento, os dois *clusters* mais similares são unidos, até que, ao final, exista somente um grande *cluster* contendo todos os objetos (NEVES; FREITAS; CÂMARA, 2001).

Quanto aos métodos hierárquicos divisivos, o algoritmo é iniciado com todos os objetos pertencendo a um único agrupamento, o qual vai sendo sucessivamente dividido, até que no final, cada *cluster* contenha um único elemento. Esta variação é mais dispendiosa computacionalmente e raramente utilizada (NEVES; FREITAS; CÂMARA, 2001).

Na Figura 8 observa-se que a seta apontada para o método aglomerativo iniciado com todos os elementos (a,b,c,d,e) como sendo um *cluster* e a cada estágio (*Step n*), os elementos vão se agrupando até estarem todos no mesmo *cluster*. Quanto ao método divisivo o oposto é mostrado, onde todos os elementos pertencem ao mesmo *cluster* e a cada estágio vão se dividindo até que cada elemento se transforme em um *cluster*.

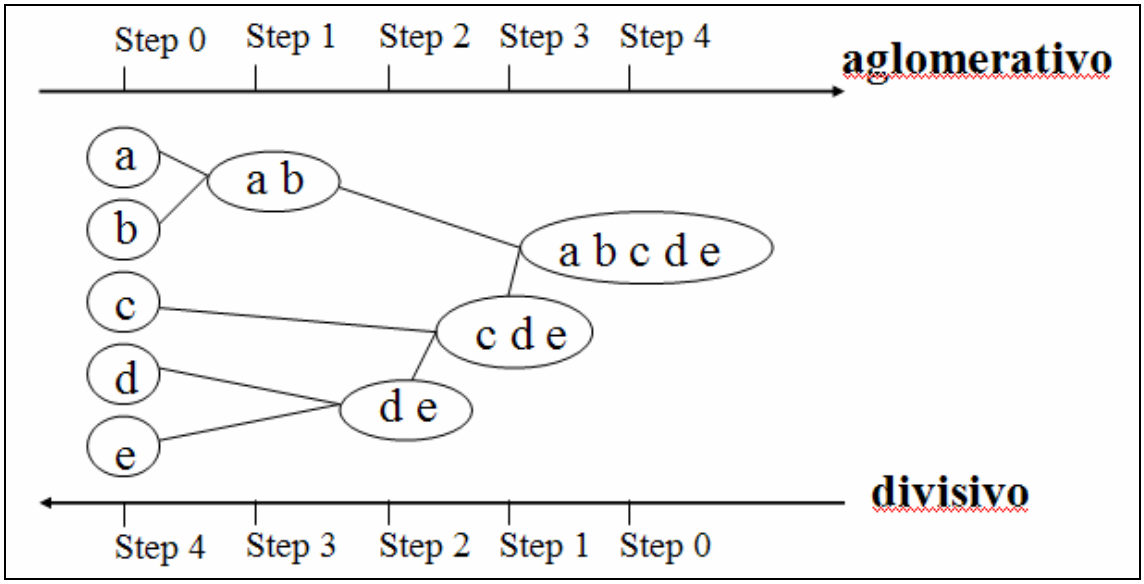

Figura 8. Estágios dos métodos divisivos e aglomerativos
Fonte: NEVES, M.; FREITAS, C.; CÂMARA, G. (2001).

O método hierárquico de clusterização apresenta como uma de suas vantagens a facilidade em lidar com qualquer medida de similaridade e a sua conseqüente aplicabilidade a diferentes atributos (numérico ou categórico). As desvantagens relacionam-se à imprecisão do critério de parada e ao fato de que a maioria dos algoritmos desta classe não reconstrói os *clusters* formados ao longo de suas execuções (OCHI; DIAS; SOARES, 2004).

4.1.1 Algoritmo COBWEB

O método hierárquico COBWEB é um tipo de algoritmo incremental e em forma de árvore. Á medida que vão sendo acrescentadas informações, a árvore vai sendo incrementada, sendo formado um agrupamento de informações nas folhas.

Dada uma árvore e uma nova informação são consideradas e comparadas as hipóteses de: criar um novo grupo (folha da árvore) ou colocar a observação numa das folhas (grupos) já existentes. Além da divisão de um grupo em subgrupos, é também possível reuni-los num só (operação inversa), isto evita dependências da ordem das observações (GAMA, 2002).

Segundo Gama (2002) os passos do algoritmo COBWEB consistem em:

a) dada uma observação é procurado o *cluster* onde melhor se adequa;
b) assumindo-se que não ficaria melhor num *cluster* isolado, antes de posicioná-lo, é considerada a hipótese de reunir os dois melhores grupos;
c) se a agregação resulta num agrupamento melhor então a observação é colocada neste novo *cluster* resultante da união dos dois que lhe são mais próximos;

d) se o passo realizado anteriormente não proporciona uma união, a observação deve ser colocada no *cluster* mais próximo.

4.2 MÉTODO DE PARTICIONAMENTO

Os algoritmos de *clusterização* por particionamento dividem a base de dados em um número de grupos definidos pelo usuário. No início, estes algoritmos escolhem alguns objetos da base como sendo centróides para cada um dos *clusters*. Os objetos então são divididos entre os *clusters* de acordo com a medida de similaridade adotada, de modo que cada objeto seja inserido no grupo que forneça o menor valor de distância entre o objeto e o centro do mesmo. Os algoritmos utilizam então, uma estratégia iterativa, que determina se os objetos devem mudar de *cluster*, fazendo com que cada um contenha somente elementos similares entre si (GOLDSCHIMIDT; PASSOS, 2005).

Os métodos de particionamento buscam encontrar, iterativamente, a melhor partição dos *n* objetos em *k* grupos. Freqüentemente os *k clusters* encontrados pelos métodos de particionamento são de melhor qualidade (grupos internamente mais homogêneos) do que os *k clusters* produzidos pelos métodos hierárquicos. Devido a este melhor desempenho, têm sido mais investigados e utilizados (SILVA, 2002).

Na execução dos algoritmos de particionamento cada iteração é submetida a uma avaliação da função objetivo. Se esta avaliação indicar que a medida não atende ao problema em questão, um nova configuração é obtida por meio da migração de elementos entre os *clusters* e o processo continua de forma iterativa até que algum critério de parada seja alcançado (NEVES; FREITAS; CÂMARA, 2002).

O método de particionamento apresenta como uma de suas vantagens a reconstrução dos *clusters* ao longo de suas execuções, obtendo grupos com medidas

mais aproximadas. As desvantagens relacionam-se à seleção dos centróides, que influencia diretamente no resultado final e na limitação quanto a aplicabilidade de atributo não númericos (OCHI; DIAS; SOARES, 2004).

Os métodos de particionamento possuem como principais algoritmos o *K-medoids* e o *K-means* (SILVA, 2002).

4.2.1 Algoritmo *K-medoid*

O algoritmo *K-medoid* utiliza um objeto representativo chamado *medoid*, localizado no centro do *cluster*. Este algoritmo utiliza-se do método de clusterização *Particioning Around Medoid*s (PAM) que realiza uma busca exaustiva pela troca de um dos *k medoids* previamente selecionados por um dos demais *(n-k)* objetos, que minimize as dissimilaridades entre os *k medoids* e os membros dos *k cluster*s (NEVES; FREITAS; CÂMARA, 2001).

Devido ao fato de procurar continuamente trocas possíveis entre um *medoid* e um objeto não selecionado que minimize as dissimilaridades, o PAM não é eficiente, principalmente quando aplicado a grandes volumes de dados. Portanto, para cada *medoid* são investigadas *(n-k)* possibilidades de troca, considerando-se todos os *medoids* tem-se um total de *k(n-k)* alterações. Como exemplo, em um caso envolvendo 1000 objetos e 10 *clusters*, avaliam-se 9.900 trocas a cada iteração do algoritmo (NEVES; FREITAS; CÂMARA, 2001).

Existem duas vantagens do algoritmo *K-medoids* em relação aos métodos baseados em centrais médios: são robustos à presença de objetos fora do padrão dos demais e independem da ordem que os objetos são examinados (NEVES; FREITAS; CÂMARA, 2001).

4.2.2 Algoritmo *K-Means*

O método de particionamento *K-means* usa o algoritmo de agrupamento de dados por K médias. Este método exige a definição prévia do número de *clusters* e do posicionamento inicial dos centros dos *k clusters* no espaço de atributos. As variações e melhorias propostas para o método ficam por conta da definição inicial dos centros dos *clusters* e de avaliações realizadas ao final ou durante o processo de agrupamento (NEVES; FREITAS; CÂMARA, 2001).

O algoritmo atribui aleatoriamente os *P* pontos, definindo os centróides conforme o número de *clusters* que deve ser gerado e calcula as médias de um ponto em relação a cada elemento, atribuindo-o ao vetor que possui maior similaridade. A seguir, calculam-se novos centróides para cada grupo e o ponto é deslocado para aquele correspondente a menor distância. O processo de re-alocação de pontos a novos grupos cujos centróides são os mais próximos continua até que se atinja uma situação em que todos os pontos pertençam aos grupos dos seus centróides mais próximos (PIMENTEL; FRANÇA; OMAR, 2003).

O comportamento do algoritmo *K-means* apresenta vantagens no que concerne a simplicidade e eficiência. Os cálculos são rápidos e simples, possibilitando o processamento seqüencial dos dados e acarretando no baixo armazenamento de informações a serem processadas.

A desvantagem do algoritmo é a sua dependência dos valores iniciais de k, da ordem em que as amostras são processadas, da escolha dos primeiros centros de agrupamento e da geometria das amostras disponíveis para análise. Em alguns casos sua utilização requer experimentação com vários valores de k e diferentes escolhas dos

parâmetros iniciais (TODESCO; PIMENTEL; BETTIOL, 2004). Na figura 9, tem-se um quadro comparativo entre a vantagens e desvantagens dos algoritmos K-means e K-medoid.

Algoritmo	Vantagem	Desvantagem
K-means	Cálculos rápidos e simples	Dependência das escolha dos primeiros centróides
	Baixo armazenamento de informações a serem processadas.	Sensível à presença de objetos fora do padrão dos demais
K-medoid	São robustos à presença de objetos fora do padrão dos demais	Alto armazenamento de informações a serem processadas.
	Independem da escolha que os objetos centrais são selecionados	Exige um alto número de interações

Figura 9. Quadro de vantagens/Desvantagens dos algoritmos K-means e K-medoid

Considerando-se as vantagens proporcionadas pelo algoritmo de particionamento *K-means*, decidiu-se implementá-lo nesta pesquisa, sendo uma das opções da tarefa de clusterização da *Shell* Orion *Data Mining Engine*.

5 O ALGORITMO DE PARTICIONAMENTO K-MEANS

O *K-means* é um método heurístico clássico da literatura que possui um algoritmo de aprendizagem baseado em resolver problemas por meio da aglomeração de vários conjuntos de dados, que busca nos centros desses a minimização direta do critério de erro calculado em função da distância (GOLDSCHIMIDT; PASSOS, 2005).

A principal função deste algoritmo é organizar *N* objetos da base de dados em *k* partições onde cada uma represente um *cluster*. O *K-means*, apesar de sua eficiência, possui a limitação de trabalhar somente com valores numéricos. O funcionamento dele é descrito resumidamente por particionar os objetos em *k clusters* e a partir da similaridade do valor da média dos atributos numéricos, agrupa os demais objetos da base de dados nestes *clusters* previamente indicados (HAN; KAMBER, 2000).

O *K-means* é chamado de algoritmo não-convexo, pois, a cada iteração diminui o valor da distorção, gerado por elementos pertencentes ao mesmo *cluster*, visto que o resultado final depende do ponto inicial usado pelo algoritmo, enfrentando problemas quando se depara com mínimos locais, ou seja, um espaço de valores muito pequenos entre os centróides (NEVES; FREITAS; CÂMARA, 2001).

Como a maioria dos métodos não-supervisionados, o *K-means* exige a definição prévia do número de *clusters* e do posicionamento inicial dos centros dos *k clusters* no espaço de atributos, onde os dados após o cálculo da distância são distribuídos. As variações e melhorias propostas para o método dependem da definição inicial dos centros dos *clusters* e de avaliações realizadas no final ou durante o processo de agrupamento (NEVES; FREITAS; CÂMARA, 2001).

A tarefa de clusterização por meio do algoritmo *K-means* é iniciada podendo adotar randomicamente, *K* pontos de dados (dados numéricos) como sendo os

centróides (elementos centrais) dos *clusters*. Logo depois, cada ponto (registro da base de dados) é atribuído ao *cluster* cuja distância deste ponto em relação ao centróide é a menor dentre todas as calculadas. Um novo centróide para cada grupo é computado pela média dos pontos do *cluster*, caracterizando a configuração dos mesmos para a iteração seguinte. O processo termina quando os centróides param de se modificar, ou após um número limitado de iterações que tenha sido especificado pelo usuário (GOLDSCHMIDT, PASSOS, 2005).

O usuário define o número de *clusters* que deseja obter, que corresponde na verdade a definição dos centróides de *K* elementos, sendo necessário, um centróide para cada *cluster* que se deseja encontrar. Esta se constitui em uma desvantagem, pois não se consegue gerar o número de centróides igual ao de grupos que devem ser criados, necessitando-se portanto, que sejam realizados novos experimentos variando o número de *clusters* (GOLDSCHMIDT, PASSOS, 2005).

A execução do algoritmo consiste em, primeiro, selecionar aleatoriamente *k* objetos, que representam a média de um *cluster*. Para cada um dos objetos remanescentes, é feita a atribuição ao *cluster* no qual o objeto é mais similar, baseando-se na distância entre o objeto e a média de um *cluster*. Esse processo se repete até que uma condição de parada seja atingida (GOLDSCHMIDT, PASSOS, 2005).

Goldschmidt e Passos (2005) definiram que o algoritmo tenta determinar k partições que minimizem a função do erro quadrado[1]. Apresenta bom desempenho quando os *clusters* são densos, compactos e bem separados uns dos outros, porém o *K-means* não é adequado para descobrir *clusters* com formas não convexas ou de tamanhos muitos diferentes.

[1] Variância somada com o quadrado de cada elemento de um conjunto

Existem objetos, denominados de ruídos (*outliers*), que não seguem o comportamento geral dos dados, pois são diferentes ou inconsistentes em relação ao conjunto de dados formado. O método *K-means* é sensível a ruídos, visto que um pequeno número destes dados pode influenciar, substancialmente, nos valores médios dos *clusters* (GOLDSCHMIDT, PASSOS, 2005).

As características do funcionamento do algoritmo de particionamento *K-means*, constitui-se nas fundamentais abordadas na próxima seção.

5.1 O FUNCIONAMENTO DO ALGORITMO K-MEANS

O objetivo do algoritmo é examinar todos os elementos de uma série de dados e associá-los a um centróide, usando para isso uma função capaz de medir a distância entres os mesmos. Quanto maior a similaridade, menor é a distância entre os pontos (HAN; KAMBER, 2000).

Dentre as fórmulas de cálculo de distância aplicado ao algoritmo *K-means* tem-se:

a) **distância euclidiana:** corresponde a distância mínima da soma das raízes quadradas entre os dois pontos considerando todas as coordenadas do espaço de atributos. Simples de calcular, sua função matemática é definida por (GOLDSCHMIDT, PASSOS, 2005):

$$dist(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2}$$

b) **distância de *hamming* (*city-block*):** soma dos pontos (de origem e em questão) calculados no espaço de atributos. Simples e rápida para calcular, pode ter pouca precisão. A fórmula é compreendida por (GOLDSCHMIDT, PASSOS, 2005):

$$dist(p,q) = | p_1 - q_1 | + | p_2 - q_2 | + ... + | p_n - q_n |$$

Esses cálculos são aplicados após o algoritmo selecionar os centróides, sendo que todos os elementos têm a distância calculada em relação a cada um deles. O algoritmo aloca cada elemento para o *cluster* cuja distância em relação ao centróide for menor. Quando nenhum elemento fica pendente, a primeira etapa está concluída. Nesse momento é necessário encontrar novos centróides, selecionando-os conforme o valor médio da distância encontrado no *cluster*. Então, novos cálculos de distância são realizados, re-alocando cada elemento no grupo com centróide mais próximo. Esse processo só é encerrado quando finalmente os conjuntos de elementos não mais mudam de *cluster* (HAN; KAMBER, 2000).

De acordo com Gama (2002) pode-se definir uma seqüência lógica para o algoritmo *K-Means*:

a) seleciona *K* pontos (elementos) no espaço representado pelos objetos que estão sendo aglomerados. Estes pontos representam centróides iniciais do grupo;
b) atribui cada objeto ao grupo que tem o centróide mais próximo;
c) quando todos os objetos foram atribuídos, recalcula as posições dos centróide de *K*;
d) repete as etapas b e c até que todos os pontos fiquem próximos de um centróide, observando uma limitação no número de iteração do algoritmo.

O funcionamento do algoritmo *k-means* poderá ser melhor compreendido por meio de um exemplo prático, para isso será utilizado um pequeno conjunto de dados mostrando cada funcionalidade do algoritmo:

V = {3, 1, 2, 0, 2, 10, 12, 9, 8, 11}

O algoritmo vai selecionar aleatoriamente dois centróides, por exemplo, 1 e 3 (valores pertencentes ao conjunto mostrado anteriormente) para serem os elementos centrais:

C1 = 1

C2 = 3

O algoritmo *k-means* deve associar todos os elementos a um grupo, onde é calculada a distância deles em relação ao centróide. A seguir, tem-se um exemplo dessa atribuição de elementos aos *clusters*:

G1 = {1, 2, 0, 2}

G2 = {3, 10, 12, 9, 8, 11}

Calcula-se um valor médio para cada *cluster*, abaixo tem-se a soma do valor de cada elemento dividido pela quantidade deles em cada *cluster*:

M1 = 1.25

M2 = 8.8

O algoritmo *k-means* deve encontrar um novo centróide para cada *cluster* usando como base o seu valor médio, onde é selecionado um elemento do conjunto inicial de dados próximo a esse valor:

C1 = 2

C2 = 9

O algoritmo vai calcular a distância do elemento comparada com os novos centróides, gerando novos grupos:

G1 ={3, 1, 2, 0, 2}

G2 = {10, 12, 9, 8, 11}

Novamente calculam-se os valores médios de cada *cluster*:

M1 = 1.6

M2 = 10

Esses procedimentos são repetidos até que o valor médio dos grupos não se modifiquem ou que os elementos pertencentes a eles se alterem. Porém, pode-se permitir ao usuário definir um número máximo de interações no algoritmo, ou o próprio algoritmo limita esse número.

A Figura 10 apresenta o algoritmo de uma função que aloca os elementos em dois grupos, baseando-se no resultado da função de distância, calculando logo depois, o valor médio de cada *cluster*. A função denominada *k-means* recebe por parâmetro o conjunto de dados a ser clusterizados (v) e os elementos definidos como centrais (m1 e m2), conforme o resultado do cálculo da função abs(), que implementa a distância *city-block* o elemento é atribuído para um dos vetores (grupos chamados no fonte de v1[] e v2[]), posteriormente o valor médio de cada um dos grupos são apresentados. Após encerrar as interações para encontrar os *clusters* o algoritmo é finalizado.

```
kmeans <- function(v, m1=1, m2=3) {
 v1 <- NULL
 v2 <- NULL
 j1 <- 0
 j2 <- 0
 for (i in 1 :length(v)) {
  if (abs(v[i] - m1) <= abs(v[i] - m2)) {
        j1 <- j1 + 1
        v1[j1] <- v[i]
  }
  else {
        j2 <- j2 + 1
        v2[j2] <- v[i]
  }
 }
 cat("Grupo 1", v1, "Media ",mean(v1),"\n")
 cat("Grupo 2", v2, "Media ",mean(v2),"\n")
}
```

Figura 10. Algoritmo K-Means
Fonte: GAMA, J. (2002).

Concluindo-se o entendimento acerca do algoritmo de particionamento *K-means* o próximo capítulo apresenta exemplos de alguns trabalhos realizados em diferentes instituições universitárias e de pesquisa com a aplicação da tarefa de clusterização por meio dele.

6 ALGUNS EXEMPLOS DE PESQUISAS REALIZADAS COM O USO DA TAREFA DE CLUSTERIZAÇÃO PELO ALGORITMO *K-MEANS*

O algoritmo de particionamento *K-means* que implementa a tarefa de clusterização tem sido estudado e aplicado em diversas áreas do conhecimento, tais como: psiquiatria com o objetivo de redefinir categorias de diagnósticos existentes; arqueologia para investigar os relacionamentos entre os vários tipos de artefatos; genética, especialmente após a criação do projeto genoma humano; bem como nas áreas de *marketing* e economia com o propósito de obter conhecimento sobre os padrões de consumo (SELINGER, 2003).

6.1 A TÉCNICA DE CLUSTERIZAÇÃO, POR MEIO DO ALGORITMO *K-MEANS*, NO PROCESSO DE *DATA MINING* EM SAÚDE BUCAL

Esta pesquisa foi realizada como Trabalho de Conclusão do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense (UNESC) em 2003. Demonstra a aplicação de *data mining*, por meio da técnica de clusterização, para traçar a incidência da cárie dental na região sul. A base de dados utilizada para a descoberta de conhecimento baseou-se no levantamento epidemiológico de saúde bucal realizado pelo Departamento de Informática do Sistema Único de Saúde (DATASUS), em crianças de 06 a 12 anos de idade no ano de 1996 nas três capitais do sul do Brasil (SELINGER, 2003).

6.2 MINERAÇÃO DE DADOS EM GRANDES BANCOS DE DADOS GEOGRÁFICOS

Desenvolvido pelo Instituto Nacional de Pesquisa Espaciais (INPE) o projeto de *data mining* em grandes bancos de dados geográficos tem como objetivo desenvolver métodos que possam ser aplicados a objetos espaciais com representação poligonal, considerando as relações de vizinhanças entres eles e a dependência espacial entre os atributos. Esta pesquisa volta-se a adaptar, implementar e avaliar um conjunto de algoritmos combinando diferentes abordagens e métodos de clusterização (NEVES; FREITAS; CÂMARA, 2001).

6.3 A IDENTIFICAÇÃO DE GRUPOS DE APRENDIZES NO ENSINO PRESENCIAL UTILIZANDO TÉCNICAS DE CLUSTERIZAÇÃO

Esta pesquisa foi publicada nos Anais do XIV Simpósio Brasileiro de Informática na Educação em 2003 e descreve uma experiência de categorização de alunos utilizando tarefas de clusterização por aprendizado não-supervisionado com dados obtidos por meio de questionários e avaliações. Com isto, foi possível identificar grupos similares de aprendizes, visando a criação e manutenção do modelo do estudante em um sistema tutor inteligente (PIMENTEL; FRANÇA; OMAR, 2003).

6.4 O USO DE FAMÍLIAS DE CIRCUITOS E REDE NEURAL ARTIFICIAL PARA PREVISÃO DA DEMANDA DE ENERGIA ELÉTRICA

Artigo publicado no XXIV Encontro Nacional de Engenharia de Produção em 2004, apresenta os resultados da aplicação de técnicas de inteligência artificial para o estudo das famílias de circuitos (residencial, comercial, industrial, rural, público e outros), utilizando informações sobre o consumo de energia por meio da aplicação de algoritmos de clusterização e, posteriormente, de uma rede neural artificial para a aproximação das curvas de demanda para as famílias identificadas (TODESCO; PIMENTEL; BETTIO, 2004).

Neste capítulo apresentaram-se diferentes aplicações da tarefa de clusterização pelo algoritmo K-means, sendo a seguir demonstrado o seu desenvolvimento na *Shell* Orion *Data Mining Engine*.

7 A TAREFA DE CLUSTERIZAÇÃO PELO ALGORITMO *K-MEANS*

A S*hell* Orion *Data Mining Engine* é um software capaz de implementar tarefas de *data mining* em bases de dados de grande e pequeno porte e está sendo desenvolvida pelo Grupo de Pesquisa em Inteligência Computacional Aplicada do Curso de Ciência da Computação da UNESC.

Na fase inicial da Orion dois módulos foram desenvolvidos, implementando-se as tarefas de associação e classificação. Atualmente, encontra-se em desenvolvimento outros métodos para associação e classificação, bem como a tarefa de clusterização, realizada nesta pesquisa por meio do algoritmo de particionamento *K-means*.

A elaboração desta pesquisa será demonstrada por meio de uma breve introdução referente as tarefas e métodos disponíveis na *Shell* Orion; a metodologia adotada para o desenvolvimento do trabalho e os resultados obtidos.

7.1 SHELL ORION DATA MINING ENGINE

A *Shell* Orion *Data Mining Engine* permite a conexão com os Sistemas Gerenciadores de Banco de Dados (SGBD) PostgreSQL e Firebird, sendo desenvolvida por meio da tecnologia Java (www.sun.org).

Duas pesquisas de Trabalhos de Conclusão de Curso foram realizadas com a implementação de tarefas e métodos de *data mining* para a *Shell* Orion:

a) **A Tarefa de Classificação e o Algoritmo ID3 para Indução de Árvores de Decisão na *Shell* de *Data Mining* Orion:** desenvolvido por Diana Colombo Pelegrin, em 2005, consistiu na geração de regras

(Figura 11), bem como na sua demonstração por meio de uma visualização gráfica no formato de árvore de decisão (Figura 12);

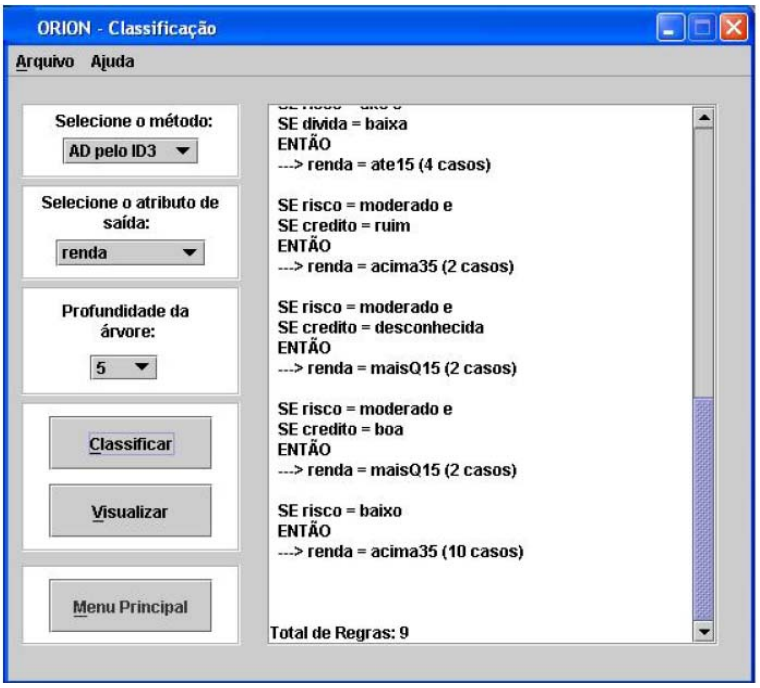

Figura 11. Tarefa de classificação pelo algoritmo ID3 na *Shell* Orion
Fonte: PELEGRIN, D. (2005)

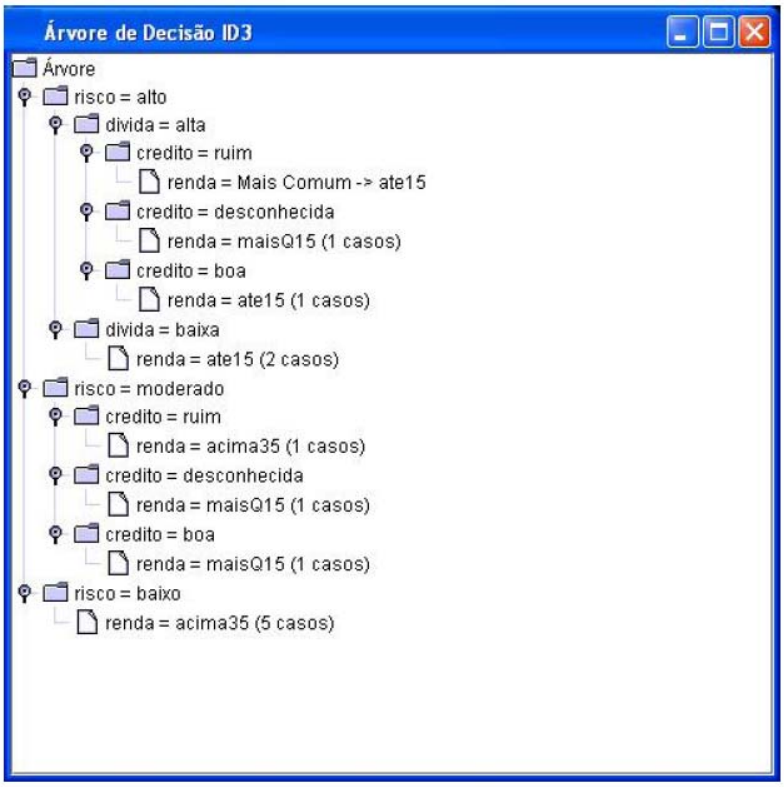

Figura 12. Visualização da árvore de decisão gerada
Fonte: PELEGRIN, D. (2005)

b) **O Módulo da Tarefa de Associação pelo Algoritmo Apriori no Desenvolvimento da *Shell* de *Data Mining* Orion:** desenvolvido por Diego Paz Casagrande, em 2005, esta pesquisa implementou o algoritmo Apriori, considerado o mais utilizado para geração das regras de associação. Neste algoritmo estão presentes os atributos de suporte e confiança que conferem a propriedade antimonotonia da relação e garantem a validade das regras extraídas (Figura 13).

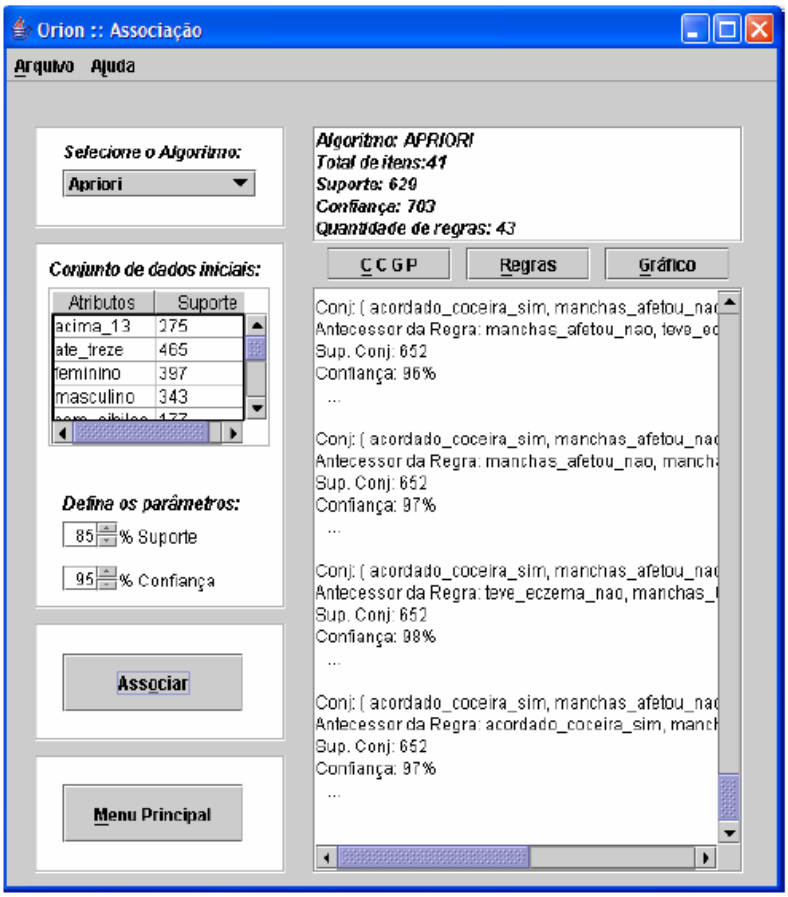

Figura 13. Visualização das regras de associação geradas pela *Shell* Orion
Fonte: CASAGRANDE, D. (2005)

Novas tarefas e métodos encontram-se em desenvolvimento a fim de serem integradas a *Shell* Orion, sendo que o objetivo dessa pesquisa fundamentou-se na implementação da tarefa de clusterização pelo método de partionamento por meio do algoritmo *K-means*.

7.2 METODOLOGIA

O desenvolvimento da tarefa de clusterização da *Shell* Orion *Data Mining Engine* baseou-se metodologicamente pelas seguintes etapas: revisão bibliográfica; modelagem do módulo de clusterização pelo algoritmo de particionamento *K-means;* demonstração matemática do algoritmo *K-means*; implementação e realização de testes.

7.2.1 Revisão Bibliográfica

Consistiu no levantamento bibliográfico dos temas envolvidos nesta pesquisa, com o intuito de compreender e descrever a descoberta de conhecimento em bases de dados e o processo de *data mining.*

Nesta etapa aprofundaram-se os estudos sobre a tarefa de clusterização e o algoritmo de particionamento *K-means*, visto que se constituem na base desta pesquisa.

Durante os estudos realizados, enfatizou-se a compreensão do algoritmo *K-means*, bem como da modelagem matemática para a escolha dos centróides, cálculo de distância e atribuição dos *clusters*.

7.2.2 Modelagem da Tarefa de Clusterização por meio do Algoritmo *K-Means*

A modelagem da tarefa de clusterização por meio do algoritmo *K-means* foi elaborada visando principalmente facilitar o desenvolvimento do módulo, como também a interação do usuário com a ferramenta, proporcionando-lhe uma interface simples e padronizada. Na modelagem da tarefa, utilizou-se a *Unified Modeling Language* (UML), realizando-se os diagramas de casos de uso, atividades e seqüência.

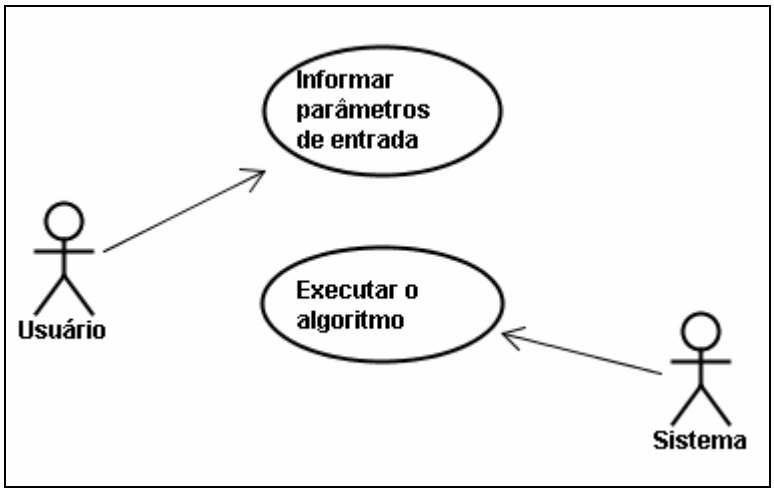

Figura 14. Diagrama de casos de uso

Na Figura 14 pode-se observar o diagrama de casos de uso do módulo da tarefa de clusterização pelo algoritmo *K-means* na *Shell* Orion, que compreende as ações que o atores usuário e sistema deverão executar no sistema:

a) **informar parâmetros de entrada:** o usuário tem a opção de selecionar tabela, atributos, atributo de saída, o cálculo da distância a ser utilizado e o número de *clusters*;

b) **executar algoritmo:** a partir dos parâmetros de entrada informados pelo ator usuário, o sistema realizará a execução do algoritmo K-means a fim de clusterizar a base de dados.

O diagrama de atividades demonstra o aspecto dinâmico do sistema e as tarefas realizadas pelos dois atores (Figura 15):

a) **informa parâmetros de entrada:** deve-se informar a tabela a ser clusterizada, assim como os atributos a serem considerados pelo algoritmo, o atributo de saída (atributo da tabela), o cálculo da distância desejado e o número de clusters;

b) **solicita execução do algoritmo**: o algoritmo não inicializa automaticamente, necessitando que o usuário o faça, antes de ser executado passa ainda por uma validação dos campos informados;

c) **processa o algoritmo *K-means***: após validar os atributos o algoritmo é inicializado;

d) **visualiza os resultados**: é disponibilizado um sumário dos dados onde visualiza-se o número de *clusters* construídos e de registros em cada um, bem como os centróides e a quantidade de ocorrências por *cluster* gerado a partir do atributo de saída selecionado.

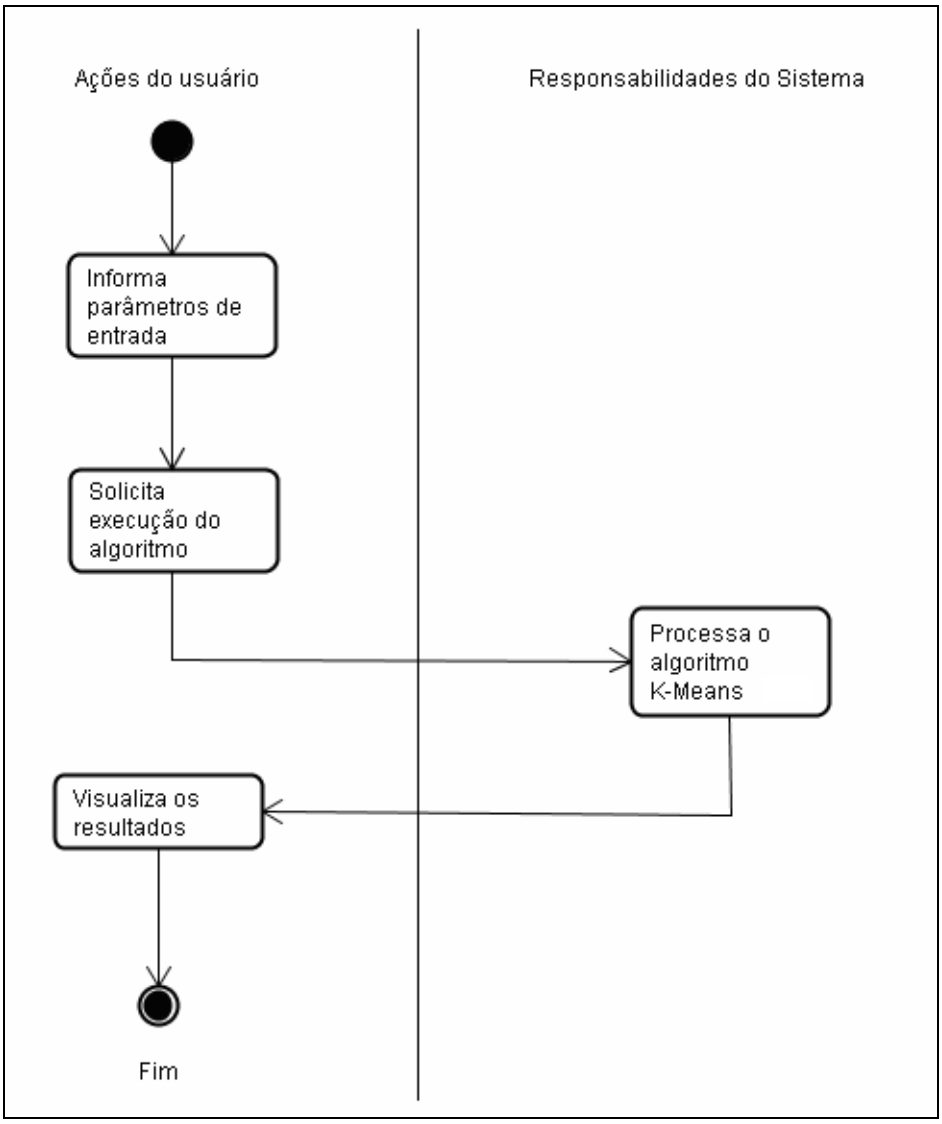

Figura 15. Diagrama de atividades

A seguir pode-se observar o diagrama de seqüência (Figura 16) que fornece uma noção da forma como o sistema se comporta, tendo-se as seguintes funções:

a) **abrirMetodoKmeans():** o processo a partir de uma única classe, onde é enviado por parâmetros os campos informados pelo usuário;

b) **executarAlgoritmo():** inicializa a execução do algoritmo.

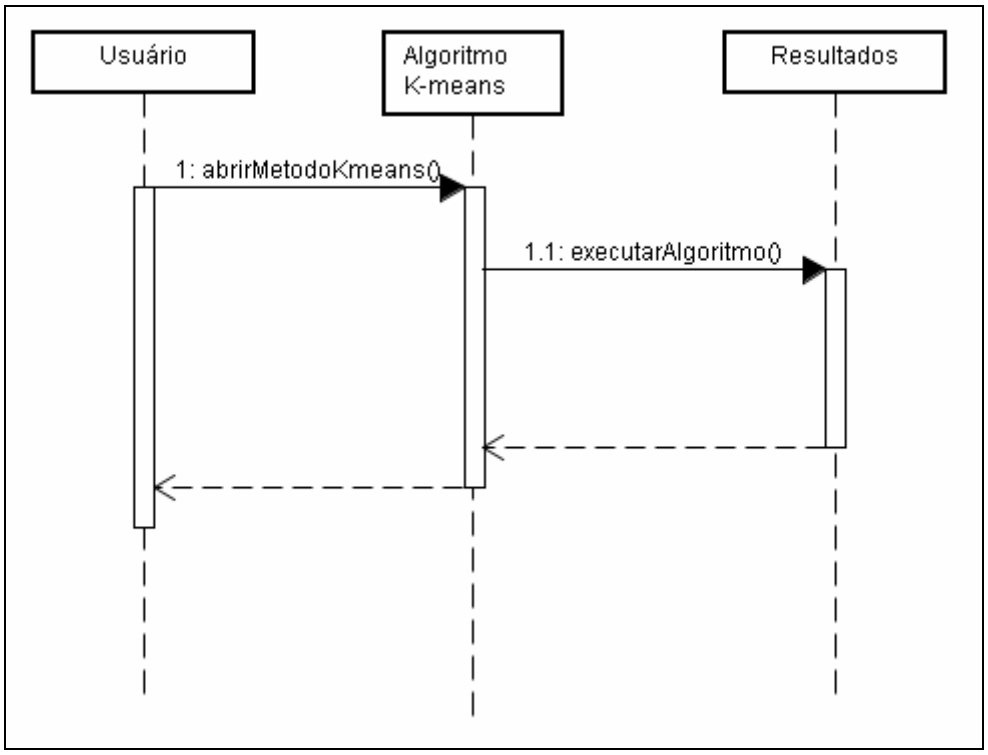

Figura 16. Diagrama de seqüência

7.2.3 Demonstração Matemática do Algoritmo *K-Means*

O algoritmo de particionamento na tarefa de clusterização da *Shell* Orion *Data Mining Engine* possui uma interface, onde o usuário informa os parâmetros necessários para o algoritmo. Após a definição destes parâmetros os passos implementados pelo algoritmo são descritos na Figura 17.

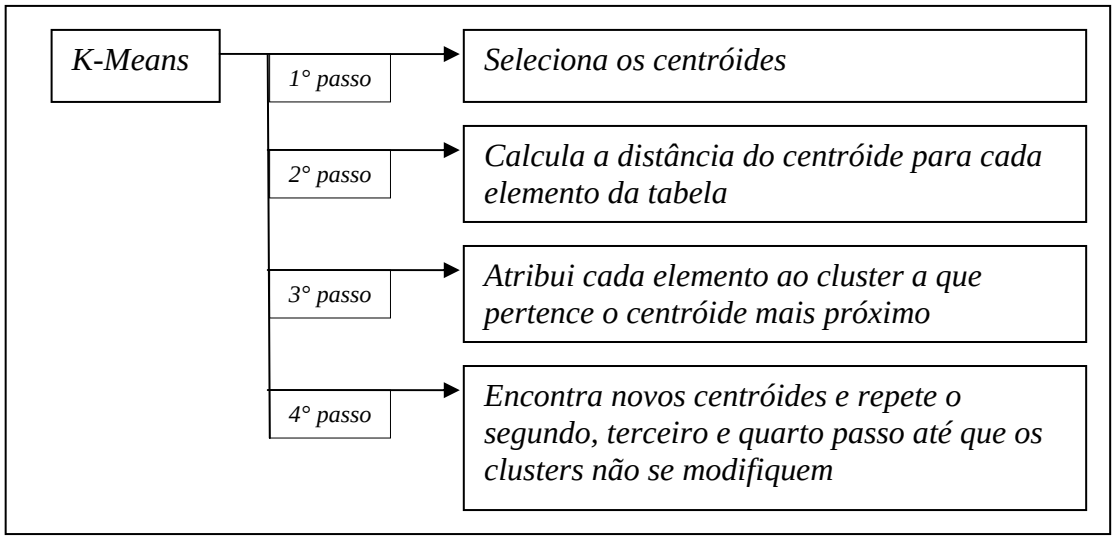

Figura 17. Seqüência de interações do algoritmo *K-means*

Antes do algoritmo *K-means* ser executado, em alguns casos onde a base possui atributos categóricos, é necessária a conversão dos dados selecionados pelo usuário para numérico, pois conforme descrito anteriormente, o *K-means* trabalha somente com este tipo de atributo. Esta conversão contempla a geração de uma tabela numérica representativa dos dados selecionados a fim de possibilitar a realização dos cálculos de distância.

Na Tabela 1 tem-se o exemplo de uma base de dados que se deseja clusterizar, enquanto na Tabela 2 demonstra-se a conversão numérica gerada pelo algoritmo.

Tabela 1. Tabela do banco de dados

Sexo	Idade	Rinite	Asma	Sono perturbado
Masculino	14	Sim	Não	Não
Feminino	12	Não	Não	Não
Feminino	12	Sim	Não	Não
Masculino	13	Não	Sim	Não

Tabela 2. Tabela numérica

Sexo	Idade	Rinite	Asma	Sono perturbado
0	2	1	0	0
1	1	0	0	0
1	1	1	0	0
0	0	0	1	0

Ressalta-se que a Tabela 2 é apenas uma ilustração usada para demonstrar essa etapa da tarefa de clusterização pelo algoritmo *K-means*, podendo existir diferenças em relação aos valores gerados na execução do algoritmo na *Shell* Orion. A partir da conversão dos dados o algoritmo *K-means* pode ser executado, conforme abordado nos próximos itens.

7.2.3.1 Seleciona os centróides

A primeiro passo a ser executado pelo algoritmo é a seleção dos centróides, nesta exemplificação o número de *clusters* usado será igual a dois, porém é importante observar que se trata de uma exemplificação e o usuário da Shell Orion pode definir o número de *cluster*. A partir disso, ele pode selecionar randomicamente os centróides, porém, na implementação do *K-means* na *Shell* Orion, utilizou-se o seguinte critério: os elementos centrais dos *clusters* devem possuir valores de atributos diferentes, pois os centróides influenciam diretamente no resultado da tarefa e se fossem considerados os de valores iguais o cálculo da distância também seria igual o que poderia impedir a geração deste *cluster*.

Baseando-se nessa composição dos centróides, pode-se definir a seguinte condição para os dois centróides dessa demonstração:

Centróide1 = 1° linha

Centróide 2 = 2° linha

O próximo procedimento do algoritmo compreende o cálculo da distância de cada elemento para um cada centróide, conforme é mostrado no item a seguir.

7.2.3.2 Calcula a distância do centróide para cada elemento da tabela

O algoritmo deve calcular a distância de cada elemento da tabela para cada centróide, nesta exemplicação é usada a distância euclidiana, já descrita no Capítulo 5, porém na Shell Orion o usuário pode optar entre distância euclidiana e *city-block*. A seguir, tem-se a aplicação dessa fórmula na *Shell* Orion, onde está sendo calculada para a Tabela 2 a distância da 1° linha em relação ao segundo elemento central (2° linha).

$$\text{distância}(1,2) = \sqrt{(0-1)^2 + (2-1)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2}$$

$$\text{distância}(1,2) = \sqrt{1+1+1+0+0}$$

$$\text{distância}(1,2) = \sqrt{3}$$

distância(1,2) =1,733

Nesta demonstração do cálculo da distância aplicou-se o arredondamento do valor resultante a fim de facilitar a compreensão do mesmo, tendo-se então:

distância(1,2) = 2

O algoritmo continua realizando o cálculo da distância euclidiana para todas as linhas e centróides, tendo-se os resultados dos cálculos armazenados em uma matriz de similaridade (Tabela 3).

Tabela 3. Matriz de similaridade

Linha	Centróide	Distância
1	1	0
2	1	2
3	1	1
4	1	2
1	2	2
2	2	0
3	2	1
4	2	2

O próximo procedimento do algoritmo é atribuir os elementos mais pertos do centróide ao mesmo *cluster*, passo descrito a seguir.

7.2.3.3 Atribui cada elemento ao cluster a que pertence o centróide mais próximo

O algoritmo atribui os elementos mais pertos do centróide ao mesmo *cluster* esse resultado é obtido por meio da regra:

Se Distancia (1,1) <= Distancia (1,2) então

Atribui = (“Linha 1 pertence ao cluster 1”)

Senão

Atribui = (“Linha 1 pertence ao cluster 2”)

Todos os resultados dos cálculos são submetidos a essa regra e com isso associa-se a linha ao grupo, originando-se uma matriz de *clusters* que associa cada linha ao grupo a que pertence (Tabela 4).

Tabela 4. Matriz de Clusters

Linha	*Cluster*
1	1
2	2
3	1
4	2

Depois de concluída a etapa de atribuição aos *clusters*, calcula-se o valor médio de distância deles, demonstrado na próxima sessão.

7.2.3.4 Encontra novos centróides e repete o segundo, terceiro e quarto passo até que os clusters não se modifiquem

Calcula-se o valor médio de distância de cada elemento pertencente ao *cluster*, em seguir, soma-se o valor médio de cada elemento e dividi-se pelo total de elementos que pertencem ao cluster, tem-se então o valor médio de distância do *cluster*, conforme abaixo:

Média = Soma(distância(linha,centróide))/Linhas

A partir disso, selecionam-se novos centróides com o valor médio igual ou próximo aquele calculado para o *cluster*, gerando-se uma nova matriz de similaridade e realizando-se uma nova distribuição dos elementos aos grupos. Caso os *clusters* permaneçam nas mesmas posições o algoritmo é encerrado, ao contrário, repete-se o processo até eles não alterem mais as suas posições, definindo-se para isso um limite de interações.

7.2.4 Implementação da Tarefa de Clusterização por meio do Algoritmo *K-means*

A tarefa de clusterização na *Shell* Orion *Data Mining Engine* foi implementada por meio da tecnologia Java devido a algumas de suas várias

características, como: permite reutilização de código; as aplicações em Java rodam em qualquer sistema operacional (multiplataforma) e suas ferramentas de programação são gratuitas. O ambiente de programação Java utilizado para esta pesquisa foi o NetBeans 5.5 disponível em http://www.netbeans.org.

A *Shell* Orion *Data Mining Engine* possui uma interface de fácil interação, na Figura 18 pode-se visualizar a tela de conexão com o banco de dados, no caso o PostgresSQL.

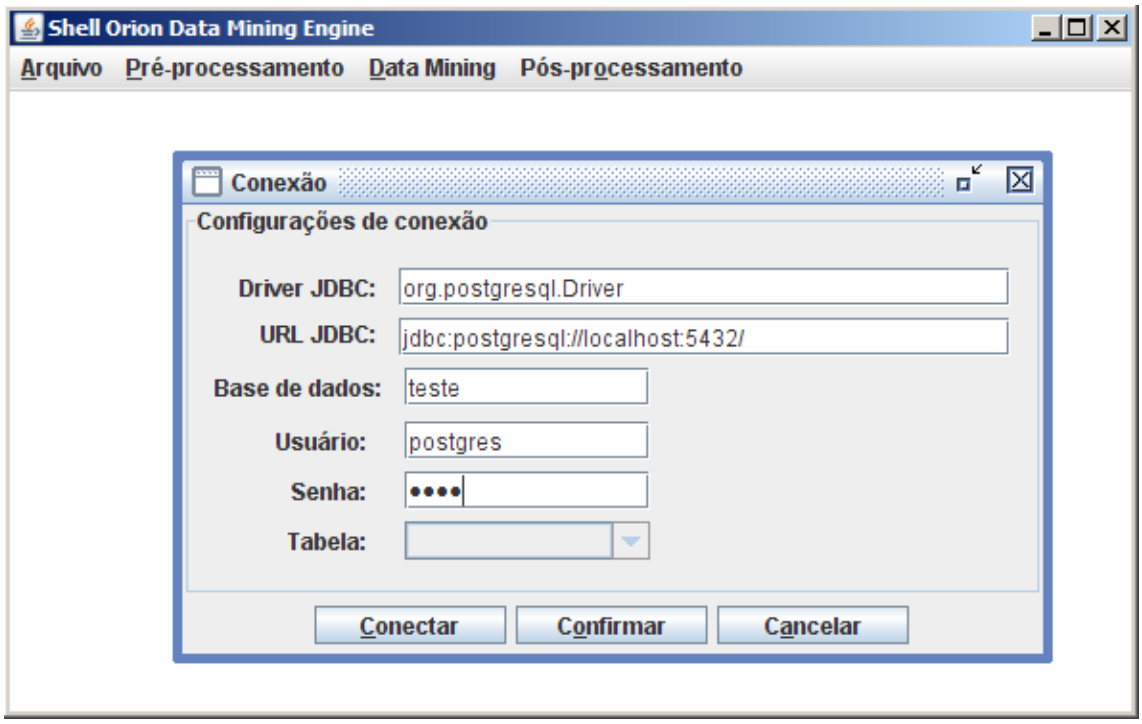

Figura 18. Tela de conexão da *Shell Orion Data Mining Engine*

O usuário da *Shell* Orion tem disponível na tarefa de clusterização a opção *K-means*, onde é processado o algoritmo, nesta tela deve-se selecionar a tabela de origem dos atributos a serem clusterizados, assim como definir o tipo de cálculo da distância a ser utilizada (euclidiana ou *city-block*), o número de *clusters* e o atributo de saída. Os resultados obtidos são disponibilizados por meio de um sumário de dados, podendo-se visualizar também graficamente. Além disso, encontra-se disponível o *help* do módulo, as opções de salvar e imprimir o resultado, bem como gerar o arquivo SQL

a fim de que os clusters construídos possam ser utilizados como dados de entrada para outras tarefas de *data mining* (Figura 19).

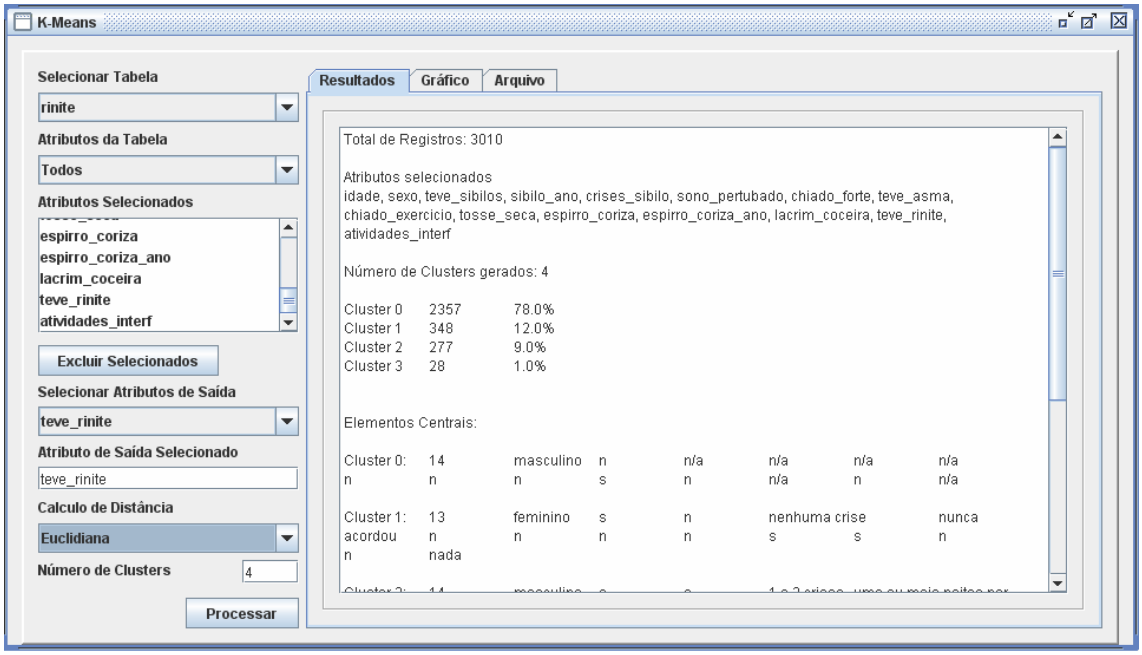

Figura 19. Tela do algoritmo *K-means*

Referente aos parâmetros informados pelo usuário, tem-se a Figura 20 que ilustra todas as tabelas da base de dados, permitindo-se selecionar os atributos que serão utilizados.

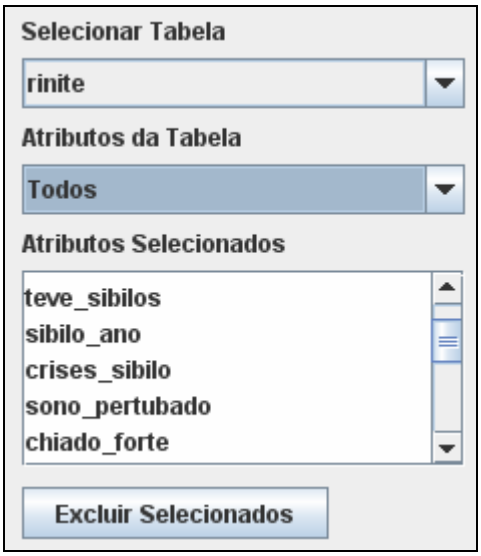

Figura 20. Informações da base de dados

O algoritmo *K-means* possui também na tarefa de clusterização da Shell Orion *Data Mining Engine*, a seleção do atributo de saída, assim como do cálculo da distância e o número de *clusters* a ser gerado (Figura 21).

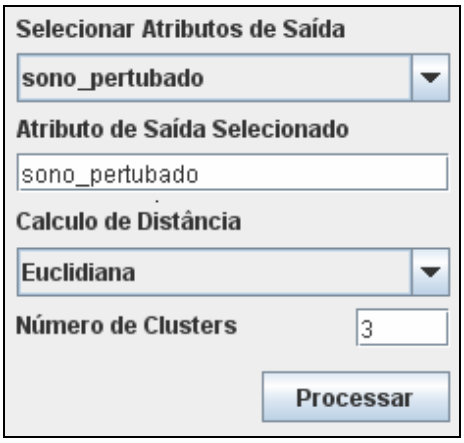

Figura 21. Parâmetros do *K-means*

A Figura 22 demonstra o resultado no formato texto apresentando-se o número total de registros da tabela; a quantidade de registros e o seu percentual por *cluster*; os elementos utilizados como centróide; o atributo de saída onde visualiza-se o número de ocorrências de um determinado dado neste atributo.

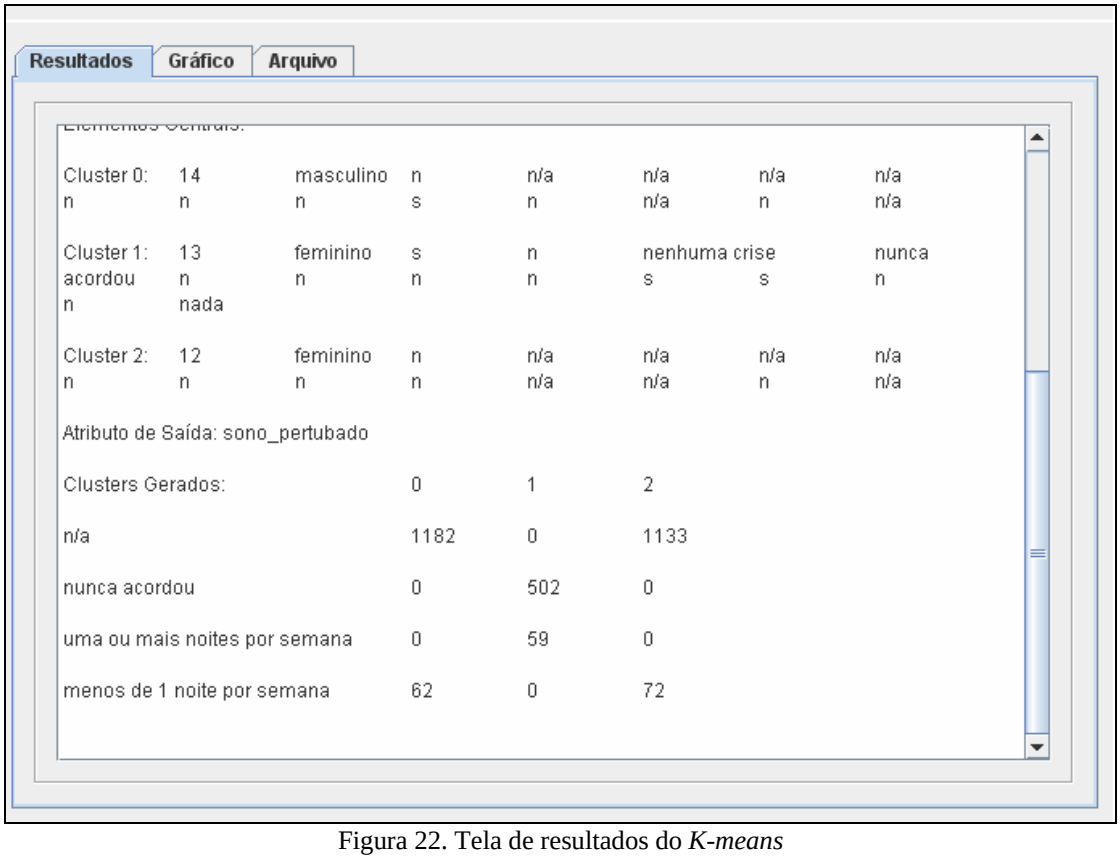

Figura 22. Tela de resultados do *K-means*

O algoritmo de particionamento *K-means* implementado na tarefa de clusterização da *Shell* Orion *Data Mining Engine* segue a demonstração matemática apresentada na Seção 7.2.3.

7.2.5 Realização dos Testes na Tarefa de Clusterização por meio do Algoritmo *K-Means*

Os testes realizados na tarefa de clusterização por meio do algoritmo de particionamento *K-means* teve por objetivo a avaliação do funcionamento da Orion e análise dos resultados obtidos. Estes testes foram executados em um microcomputador com plataforma Windows, 256 MB de memória principal e processador com freqüência de 2,16 GHz.

Nesta etapa utilizou-se a base de dados referente a prevalência da asma e rinite em adolescentes escolares no município de Criciúma que possui 3010 registros e 15 atributos. Esta base foi disponibilizada pela Dra. Jane Bettiol, professora do Programa de Pós-Graduação em Ciências da Saúde da Universidade do Extremo Sul Catarinense, que orientou a realização de um projeto de pesquisa do Programa de Iniciação Científica V, intitulado de Prevalência de Sintomas de Asma e Rinite em Adolescentes Escolares do Município de Criciúma (RAASCH, 2005).

Os resultados obtidos na realização dos testes do módulo de clusterização por meio do algoritmo *K-means* são apresentados na próxima seção.

7.3 RESULTADOS OBTIDOS

Os resultados obtidos compreenderam as análises do módulo de clusterização por meio do algoritmo *k-means* e do conhecimento descoberto na base de dados da asma e rinite em adolescentes escolares do município de Criciúma.

Na realização destes testes analisou-se o tempo de processamento do algoritmo *K-means* para gerar os *clusters*, a partir dos 3010 registros, sendo o número de *clusters* igual a 5, variando-se a quantidade de atributos selecionados e número de interações. Os resultados destas análises foram satisfatórios, se comparado ao tempo de processamento de outras ferramentas de dm disponíveis e estão demonstrados na Tabela 5.

Tabela 5. Análise de tempo de processamento

Quantidade de Atributos	Interações	Tempo
15	10	5 segundos
15	100	17 segundos
12	10	4 segundos
12	100	14 segundos
8	10	3 segundos
8	100	10 segundos
3	10	2 segundos
3	100	7 segundos

Pode-se observar por meio desta análise, que o tempo de processamento do algoritmo *K-means*, está relacionado com o número de interação que o algoritmo precisa executar até gerar o resultado final, sendo o número de interação limitada pelo usuário da *Shell* Orion. Isso se dá pelo fato de que o algoritmo executa o cálculo da distância de todos os atributos em relação a todos os centróides, alternado os elementos centrais e redistribuindo os elementos aos *clusters*, a cada interação. Assim, quanto maior a quantidade de interações e atributos selecionados maior será o tempo de processamento do algoritmo.

Realizando-se a execução do algoritmo *K-means* na *Shell* Orion por meio da tabela rinite foram selecionados todos os seus atributos e informados os seguintes parâmetros: atributos selecionados = todos; cálculo da distância = euclidiana; número de *clusters* = 3. A Figura 23 mostra os resultados obtidos para os três *clusters* gerados com ênfase no atributo de saída desejado (sono_pertubado).

Atributo de Saída: sono_pertubado			
Clusters Gerados:	0	1	2
n/a	1182	0	1133
nunca acordou	0	502	0
uma ou mais noites por semana	0	59	0
menos de 1 noite por semana	62	0	72

Figura 23. Resultado gerado pelo *K-means*

Considerando-se os mesmos parâmetros de entrada utilizados na exemplificação da Figura 23, porém alterando-se o atributo de saída para *teve_asma* o algoritmo originou o seguinte resultado (Figura 24).

Atributo de Saída: teve_asma			
Clusters Gerados:	0	1	2
n	1133	320	1095
s	111	241	110

Figura 24. Resultado gerado na Tela *K-Means*

Relacionando-se o conhecimento descoberto nas Figuras 23 e 24, pode-se observar que os *clusters* 0 e 2 (Figura 24) têm uma incidência menor de asma, sendo justamente estes os *clusters* que não possuem ocorrência de sono perturbado (Figura 23).

Assim, mediante os testes realizados pode-se concluir que o módulo de clusterização por meio do algoritmo *K-means* da *Shell* Orion está processando corretamente os dados, porém futuramente torna-se necessária a sua validação por meio de métodos estatísticos.

CONCLUSÃO

Esta pesquisa demonstrou a importância do *data mining*, bem como da tarefa de clusterização na descoberta de conhecimento em base de dados por meio do estudo referente ao algoritmo *K-means* para a construção de grupos com alguma similaridade.

Além do entendimento acerca do conceito de inteligência artificial, *data mining* e do seu funcionamento pela tarefa de clusterização, um dos propósitos principais deste trabalho foi a demonstração matemática do algoritmo *K-means*.

Durante o desenvolvimento desta pesquisa foram encontradas algumas dificuldades no que se refere a material bibliográfico sobre o algoritmo *K-means*, pois ele é citado, o pseudocódigo é apresentado, porém o funcionamento matemático não é abordado. Portanto, esta pesquisa consiste numa contribuição à comunidade acadêmica e científica, já que explicita a modelagem matemática do *K-means*.

O módulo de clusterização foi submetido a testes realizados a partir da base dados referente a asma e rinite, resultando em grupos adequados, o que comprovou o funcionamento correto do algoritmo *K-means* implementado.

Dessa forma, os resultados obtidos foram satisfatórios, atingindo-se os objetivos propostos que consistiam na compreensão dos conceitos de *data mining*; tarefa de clusterização, algoritmo de particionamento *K-means* e a realização de testes por meio de uma base de dados específica.

Dando continuidade ao desenvolvimento da *Shell* Orion, bem como desta pesquisa sugere-se como trabalhos futuros:

a) realização de um estudo comparativo entre o módulo de clusterização por meio do algoritmo *K-means* da *Shell* Orion com o de outras ferramentas;
b) desenvolvimento de outras medidas de distância, assim como novos gráficos para melhor compreensão dos resultados gerados pelo algoritmo *K-means*;
c) implementação de outros algoritmos de particionamento e hierárquicos na tarefa de clusterização da *Shell* Orion;
d) adicionar métodos de lógica fuzzy para a clusterização como por exemplo, o algoritmo *Fuzzy K-means*.

REFERÊNCIAS

AMARAL, Fernada Cristina Naliota. **Data Mining: Técnicas e Aplixações para o Marketing Direto**. São Paulo: Berkeley, 2001.

AURÉLIO, M.; VELLASCO, M.; LOPES, H. **Descoberta de Conhecimento e Mineração de Dados**. Rio de Janeiro: Departamento de engenharia elétrica da PUC, 1999. Disponível em: <http://www.ica.ele.pucrio.br/cursos/ download/DM-apostila1.pdf> Acesso em: 05 maio 2006

BARTOLOMEU, Tereza Angélica. **Modelo de investigação de acidentes do trabalho baseado na aplicação de tecnologias de extração de conhecimento**. 2002. 302 f. Tese (Doutorado em Engenharia de Produção) - Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina, Florianópolis. Disponível em: < http://teses.eps.ufsc.br/defesa/pdf/3738.pdf> Acesso em: 15 maio 2006.

BARRETO, Jorge Muniz. **Inteligência Artificial No Limiar Do Século Xxxi**. 3. Florianópolis: Duplic, 2001.

CARVALHO, Luís Alfredo Vidal de**. Datamining**: **A Mineração De Dados No Marketing, Medicina, Economia, Engenharia E Administração**. 2.ed. São Paulo: Editora Érica, 2002.

CASAGRANDE, Diego Paz. **O Módulo da Tarefa de Associação pelo Algoritmo Apriori no Desenvolvimento da Shell de *Data Mining* Orion**. 2005.Trabalho de conclusão de curso de Ciência da computação, UNESC.

CASTRO, Armando Antonio Monteiro de; PRADO, Pedro Paulo Leite do. **Algoritmos Para Reconhecimento De Padrões**. Taubaté: Departamento de Engenharia Elétrica Universidade de Taubaté, 2002. Disponivel em: <http://www.unitau.br/prppg/publica/exatas/downloads/algoritmosreconhecimento-99-02.pdf> Acesso em 10 de maio 2006.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. **From Data Mining to Knowledge Discovery in Databases**. 1996. AI Magazine. Menlo Park, 1996. Disponível em: <http://kdnuggets.com/gpspubs/aimag-kdd-overview-1996-Fayyad.pdf > Acesso em 20 fev. 2006.

GAMA, João. **Métodos de Agrupamento**. LIACC, FEP, 2002. Disponível em: <http://www.liacc.up.pt/~jgama/Bdc/aglo.pdf > Acesso em 15 maio 2006.

GOLDSCHIMIT, Ronaldo. PASSOS, Emmanuel. **Data Mining- Conceitos, Técnicas, Ferramentas, Orientações E Aplicações**. Rio de Janeir: Editora Campus, 2005.

HAND, David; HEIKKI, Mannila; SMYTH, Padhraic. **Principles Of *Data Mining***. The MIT Press, 2001.

HAN, Jiawei; KAMBER, Micheline. **Data Mining – Concepts and Techniques**. The Morgan Kaufmann Series in Data Management Systens, Jim Gray, Series Editor; 2000.

HARRISON. Thomas H. Intranet data warehouse. São Paulo: Bekerley Brasil. 1998

LACERDA. Maurício Pio de; SOUZA. Ricardo Cesar Freitas de. **Aplicação da Mineração de Dados em Sistema de Avaliação de Professor e Aluno**. Belém: UFPA, 2004. Disponível em: <http://www.cultura.ufpa.br/informatica/tcc/mauricio_ricardo.pdf> Acesso em 08 maio 2006.

LOPES. Carlos Henrique Pereira. **Classificação de Registros em Banco de Dados por Evolução de Regras de Associação Ultilizando Algoritmo Genéticos**. 1999. Disponível em : <http://www.ica.ele.pucio.br/publicacoes/download/tes_0002.pdf> Acesso em 06 maio 2005.

NEVES, M. C.; FREITAS, C. C.; CÂMARA, G. **Mineração de Dados em Grandes Bancos de Dados Geográficos.** Disponível em: <http://www.dpi.inpe.br/geopro/modelagem/relatorio_data_mining.pdf> Acesso em 10 jun. 2005.

PELEGRIN, Diana Colombo. **A Tarefa de Classificação e o Algoritmo ID3 para Indução de Árvores de Decisão na Shell de *Data Mining* Orion**. 2005.Trabalho de conclusão de curso de Ciência da computação, UNESC.

PIMENTEL, Edson P.; FRANÇA, Vilma F. de; OMAR, Nizam. **A Identificação de Grupos de Aprendizes no Ensino Presencial Utilizando Técnicas de Clusterização.** Disponível em: <http://www.nce.ufrj.br/sbie2003/publicacoes/paper52.pdf> Acesso em 05 jun. 2006.

RAASCH, Caroline Chachamovich et al. **Prevalência de sintomas de asma e rinite em adolescentes escolares do município de Criciúma**. Programa de Iniciação Científica V - Pic V. Criciúma, 2005.

REATEGUI, E. **Data Mining e Personalização Dinâmica. In: X Escola de Informática da Sbc-Sul**, 2002, Criciúma. **Anais...**Porto Alegre, RS: Instituto de Informática UFRGS 2002.

REZENDE, Solange Oliveira. **Sistemas Inteligentes: Fundamentos e Aplicação**. Editora Manole, 2002.

SERRA, L. **A Essência do Bussiness Intelligence**. São Paulo: Berkeley, 2002.

SILVA, Marcelino Pereira dos Santos. SKDQL uma linguagem declarativa de especificação de consultas e processos para descoberta de conhecimento em bancos de dados e sua implementação. 2002. 113f. Dissertação (Mestre em Ciência da Computação) – Centro de Informática da Universidade Federal de Pernambuco, Recife, BR – PE. Disponível em: <http://www.dpi.inpe.br/~mpss/docs/DissertacaoMarcelino.pdf> Acesso em: 15 jun. 2006.

SELINGER, Tarcísio Cardoso. **A Técnica de Clusterização, por Meio do Algoritmo *K-Means*, No Processo de *Data Mining* em Saúde Bucal**. Trabalho de conclusão de curso de Ciência da computação, UNESC.

SOARES, Stênio Sã Rosário Furtado; OCHI, Luiz Satoru. **Um Algoritmo Evolutivo com Reconexão de Caminhos para o Problema de Clusterização automática.** Disponível em: http://www.ic.uff.br/~satoru/conteudo/artigos/CLAIO2004-Stenio.pdf> Acesso em 05 maio 2006.

OCHI, Luiz Satoru; DIAS, Carlos Rodrigo; SOARES, Stênio S. Furtado. **Clusterização em Mineração de Dados.** Disponível em <http://www.ic.uff.br/~satoru/conteudo/artigos/ERI-Minicurso-SATORU.pdf> Acesso em 05 abril 2006.

TODESCO, José Leomar; PIMENTEL, Francisco J. S.; BETTIOL, Arlan L. **O Uso de Famílias de Circuitos e Rede Neural Artificial para Previsão de Demanda de Energia Elétrica.** Disponível em: <http://www.producaoonline.inf.br/v04n04/artigos/PDF/Enegep0902_1284.pdf>. Acesso em 10 abril 2006.

ULYSSÉA, Mário Capanema. **Descoberta de Conhecimento em Base de Dados dos Municípios da Azonasul: Uma Proposta Integradora**. Pelotas, ago. 1999. Disponível em: <http://gpia.ucpel.tche.br/iiioia/pub.htm> Acesso em 10 abril 2005.