UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

JOSÉ MÁRCIO CASSETTARI JUNIOR

O MÉTODO DE LÓGICA *FUZZY* PELO ALGORITMO GUSTAFSON-KESSEL NA TAREFA DE CLUSTERIZAÇÃO DA *SHELL* ORION *DATA MINING ENGINE*

CRICIÚMA, JULHO DE 2008

JOSÉ MÁRCIO CASSETTARI JUNIOR

O MÉTODO DE LÓGICA *FUZZY* PELO ALGORITMO GUSTAFSON-KESSEL 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.

Orientador: Profa. MSc. Merisandra Côrtes de Mattos

CRICIÚMA, JULHO DE 2008

JOSÉ MÁRCIO CASSETTARI JUNIOR

O Método de Lógica *Fuzzy* pelo Algoritmo Gustafson-Kessel na Tarefa de Clusterização da *Shell* Orion *Data Mining Engine*

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

_____
Profa. MSc. Ana Claudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação

Banca Examinadora:

_____
Profa. MSc. Merisandra Côrtes de Mattos (UNESC)
Orientador

_____
Prof. MSc. Maurício Braga de Paula (Universidade Federal de Pelotas – UFPEL)

_____
Profa. Esp. Alair Salete Budni Morais (UNESC)

Aos meus pais, José Márcio e Genésia, meu irmão Luiz Henrique e meus amigos pelo apoio concedido.

AGRADECIMENTOS

Agradeço a Deus por estar presente em minha vida, por me confortar nas horas difíceis e pelo auxilio em minhas escolhas.

Agradeço também:

A minha família, por todo carinho, amor, compreensão e apoio concedido, por sempre estarem ao meu lado e por contribuírem em minha formação pessoal.

A minha namorada Marcela, por estar ao meu lado nas horas difíceis, transmitindo todo seu amor, carinho e compreensão, além de sempre estar a disposição para me auxiliar, independente da hora ou dia.

Aos meus amigos Felipe, Fernando e Rubens, pelo apoio concedido durante a execução deste trabalho, por sua amizade e por todos os momentos alegres compartilhados ao longo da graduação.

A todos os meus colegas de graduação, por compartilharem suas experiências de vida e momentos de diversão e descontração durante todo o curso.

A professora Alair, pelo auxilio nas duvidas referentes aos formalismos matemáticos envolvidos neste trabalho.

Ao professor Dr. Leandro Coelho, por gentilmente conceder uma cópia do artigo original do algoritmo desenvolvido nesta pesquisa.

E finalmente, agradeço a grande incentivadora desta pesquisa, minha professora e orientadora Merisandra, que não mediu esforços para a conclusão deste trabalho, contribuindo com seu conhecimento e sua amizade.

Enfim, a todos que de alguma forma contribuíram para a conclusão desta pesquisa.

"A mente que se abre a uma
nova idéia jamais voltará
ao seu tamanho original".

(Albert Einstein)

RESUMO

O avanço dos modelos de armazenamento possibilitou a criação de grandes bases de dados, tornando-se necessário a utilização de técnicas que auxiliem a exploração destas informações. Dentre as disponíveis, destaca-se a aplicação dos conceitos de *data mining*, a principal etapa do processo de descoberta de conhecimento em bases de dados, sendo utilizadas para este objetivo ferramentas computacionais, que em sua maioria são comerciais. Considerando isto, encontra-se em desenvolvimento a *Shell* Orion *Data Mining Engine*, um projeto do Grupo de Inteligência Computacional Aplicada da UNESC que consiste na criação de uma ferramenta gratuita que implemente os diferentes métodos de *data mining*. Dentre os existentes, esta pesquisa fundamentou-se na demonstração matemática e implementação do algoritmo de lógica *fuzzy* Gustafson-Kessel para a tarefa de clusterização, que tem como objetivo particionar os diferentes dados em grupos com características semelhantes, considerando que a lógica *fuzzy* auxilia neste processo pois possibilita aos elementos pertencerem a grupos distintos simultaneamente. Durante a pesquisa, foram realizados alguns testes no módulo implementado, a fim de verificar os resultados obtidos pelo algoritmo.

Palavras-chave: Inteligência Artificial, *Data Mining*, Tarefa de Clusterização, Lógica *Fuzzy*, Algoritmo Gustafson-Kessel.

ABSTRACT

The improvement of storage models made possible the creation of great databases, making necessary the use of techniques that assist the exploration of these information. Amongst the available ones, it is distinguished application of data mining concepts, the main stage of the process of knowledge discovery in databases, being used for this objective computational tools, that in its majority are commercial. Considering this, it is found in development the Shell Orion Data Mining Engine, a project of the Group of Applied Computational Intelligence of the UNESC that consists of the creation of a gratuitous tool that implements different methods of data mining. Amongst the existing ones, this research was based on the mathematical demonstration and implementation of the fuzzy logic algorithm Gustafson-Kessel for the clustering task, that has as objective to partition the different data in groups with similar characteristics, considering that the fuzzy logic assists in this process therefore makes possible to the elements to belong the distinct groups simultaneously. During the research, some tests in the implemented module had been carried through, in order to verify the results gotten from the algorithm.

Keywords: Artificial Intelligence, Data Mining, Clustering Task, Fuzzy Logic, Gustafson-Kessel Algorithm.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

BIRCH	*Balanced Iterative Reducing and Clustering using Hierarchies*
CART	*Classification and Regression Trees*
DBSCAN	*Density-Based Spatial Clustering of Applications with Noise*
DM	*Data Mining*
FCE	*Fuzzy C-Elliptotypes*
FCM	*Fuzzy C-Means*
FCS	*Fuzzy C-Shells*
FCV	*Fuzzy C-Varieties*
GK	Gustafson-Kessel
HSQLDB	*Hypersonic Structured Query Language Database*
IA	Inteligência Artificial
ID3	*Iterative Dichotomiser* 3
KDD	*Knowledge Discovery in Databases*
PCA	*Principal Component Analysis*
PUCPR	Pontifícia Universidade Católica do Paraná
SQL	*Structured Query Language*
STING	*Statistical Information Grid*
TCC	Trabalho de Conclusão de Curso
UML	*Unified Modeling Language*
UNESC	Universidade do Extremo Sul Catarinense
UTI	Unidade de Terapia Intensiva

SUMÁRIO

1 INTRODUÇÃO

O grande volume de dados gerado pelas organizações necessita ser transformado em conhecimento a fim de beneficia-las. Neste caso, é comum estas instituições utilizarem métodos estatísticos, porém apenas estes cálculos matemáticos não são suficientes para descobrir informações e gerar conhecimento.

Considerando isto, surgiu o conceito de *data mining*, que implementa técnicas de Inteligência Artificial[1], Estatística[2], Banco de Dados[3] e Aprendizado de Máquina[4], para descoberta de conhecimento relevante em diferentes bases de dados. O processo de *data mining* envolve tarefas e métodos que possuem características específicas e são aplicados conforme os objetivos da descoberta de conhecimento, em uma determinada base de dados.

Estas tarefas e métodos de *data mining* encontram-se implementados em ferramentas, denominadas de *Shells*, as quais são utilizadas a fim de auxiliar na descoberta de padrões e relações relevantes. No entanto, grande parte destas aplicações são comerciais, tornando-se inviável para determinadas organizações.

Dentre estas ferramentas, existe em desenvolvimento a *Shell* Orion *Data Mining Engine*, um projeto acadêmico implementado pelo Grupo de Pesquisa em Inteligência Computacional Aplicada, do Curso de Ciência da Computação, na Universidade do Extremo Sul Catarinense.

---

[1] Ciência a qual pesquisa métodos de desenvolvimento de sistemas inteligentes, o quais possuem capacidades similares ao cérebro humano, como raciocínio e tomada de decisão (RUSSELL; NORVIG, 2004).

[2] Área matemática que realiza a análise, a descrição e a compreensão de informações numéricas (BERRY; LINOFF, 2004).

[3] Repositório de armazenamento de dados, possibilitando a organização e a utilização destas informações posteriormente (WITTEN; FRANK, 2005).

[4] Área da inteligência artificial dedicada a implementação de algoritmos que permitam o aprendizado de um sistema, com o objetivo de melhorar seu desempenho (LUGER, 2004).

A *Shell* Orion possui implementado os módulos referentes as tarefas de associação pelo algoritmo *APriori*, de classificação pelos algoritmos ID3 e CART e de clusterização pelos algoritmos *K-Means* e *Kohonen*.

A tarefa de clusterização, também conhecida como agrupamento, objetiva particionar a base de dados em grupos, denominados de *clusters*, onde cada elemento integrante seja similar aos outros que se encontram no mesmo grupo, diferenciando-os dos demais *clusters* (GOLDSCHMIDT; PASSOS, 2005).

No entanto, quando a tarefa de clusterização é aplicada em grandes bases de dados, parte da informação pode estar localizada entre dois *clusters*. Nesta situação, existe a possibilidade destes dados serem forçados a pertencer a um grupo o qual não possua suas características, o que prejudica a execução da tarefa e por conseqüência, gera ambigüidade dos resultados obtidos (BEZDEK et al, 2005).

A fim de solucionar este problema existe, dentre os métodos de clusterização, o de lógica *fuzzy*, onde os elementos de um *cluster* podem pertencer a outros grupos ao mesmo tempo, dependendo dos graus de pertinência envolvidos. Desta forma, esta pesquisa consiste na modelagem matemática e implementação deste método na *Shell* Orion, por meio do algoritmo Gustafson-Kessel. Conforme Höppner et al (1999) este algoritmo se caracteriza por permitir que cada *cluster* modifique-se de tamanho, independente dos outros grupos, para se adaptar aos diferentes tipos de dados.

1.1 OBJETIVO GERAL

Desenvolver na tarefa de clusterização da *Shell* Orion *Data Mining Engine* o método de lógica *fuzzy* por meio do algoritmo Gustafson-Kessel.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa consistem em:

a) entender o conceito de *data mining* e a tarefa de clusterização;
b) compreender o método de lógica *fuzzy* e o funcionamento do algoritmo Gustafson-Kessel;
c) aplicar o algoritmo Gustafson-Kessel na tarefa de clusterização em *data mining*;
d) demonstrar a modelagem matemática do algoritmo Gustafson-Kessel;
e) utilizar uma base de dados a fim de testar o funcionamento do algoritmo Gustafson-Kessel no módulo de clusterização da *Shell* Orion *Data Mining Engine*.

1.3 JUSTIFICATIVA

O conceito de *data mining* possibilita a descoberta de conhecimento relevante em bases de dados, utilizando para isso diferentes tarefas e métodos, com o objetivo de verificar estes dados, auxiliar na exploração dos registros e criar novos conhecimentos.

A fim de analisar este volume de informações com maior facilidade e eficiência, utilizam-se ferramentas denominadas de *Shells*, que são indispensáveis para as organizações poderem analisar os dados que se encontram armazenados e descobrir relações que auxiliem na tomada de decisão e nos seus planos de negócios. De acordo com Berry e Linoff (2004) muitas destas instituições objetivam, por exemplo,

identificar a relação com seus clientes com o objetivo de melhorá-la, beneficiando todos os envolvidos.

No processo de *data mining* aplicam-se métodos que foram descritos em artigos ou teses. Apenas nos últimos anos, alguns destes algoritmos começaram a ser incorporados em ferramentas comercias, a fim de atender a necessidade de descobrir padrões e conhecimento nas bases de dados das instituições (BERRY; LINOFF, 2004). Porém, a maioria das ferramentas desenvolvidas são comerciais, o que inviabiliza a sua aquisição por algumas organizações.

Existem algumas *Shells* de *data mining* gratuitas, como por exemplo a Weka, desenvolvida pela universidade de Waikato na Nova Zelândia. No entanto, estas ferramentas não implementam todos os métodos disponíveis, o que não resolve o problema de algumas organizações, as quais precisam adquirir uma solução comercial.

Considerando isto, projetos nesta área se justificam por implementar diferentes tarefas e métodos de *data mining* em uma ferramenta gratuita, como por exemplo, o projeto acadêmico da *Shell* Orion *Data Mining Engine*. Desta forma, esta pesquisa tem como finalidade continuar este projeto, visando ampliar as suas funcionalidades, com o desenvolvimento de um método de lógica *fuzzy* no módulo de clusterização.

A tarefa de clusterização é primária no processo de descoberta de conhecimento em *data mining*, pois outras tarefas como classificação e sumarização podem ser executadas em cada *cluster* obtido, a fim de aprimorar a extração de informações relevantes em cada grupo de dados (GOLDSCHMIDT; PASSOS, 2005). Até o momento, a *Shell* Orion possui apenas dois métodos desenvolvidos neste módulo, os algoritmos *K-Means* e *Kohonen*.

Nesta pesquisa a escolha do método de lógica *fuzzy* na tarefa de clusterização, deve-se ao fato deste ser capaz de identificar com maior precisão registros que se encontram entre dois *clusters*, ou seja, dados com características semelhantes a estes grupos. O método efetua cálculos de pertinência nestas informações e dependendo das operações matemáticas, estes registros podem pertencer a vários *clusters* simultaneamente, a fim de evitar que dados sejam distribuídos em grupos os quais não possuam suas principais características, melhorando a abstração do conhecimento gerado (BEZDEK et al, 2005). Dentre os métodos de lógica *fuzzy*, será implementado o algoritmo Gustafson-Kessel.

O método proposto por Gustafson e Kessel tem como objetivo produzir *clusters* de diferentes tamanhos e formas, aumentando sua flexibilidade na divisão dos grupos. No entanto, este algoritmo exige maior capacidade de processamento, o que não inviabiliza a sua aplicação, visto que possui a característica de cada *cluster* mudar de forma ou tamanho, a fim de se adaptar as diferentes informações existentes na base de dados (HÖPPNER et al, 1999).

1.4 ESTRUTURA DO TRABALHO

Esta pesquisa é composta por oito capítulos, considerando que o Capítulo 1 é composto pela contextualização ao tema proposto, objetivos e justificativa para realização deste trabalho.

No Capítulo 2 são abordados os conceitos de *data mining*, indispensáveis para o entendimento desta pesquisa.

A *Shell* Orion *Data Mining Engine* é comentada no Capítulo 3, apresentando suas características, funcionalidades, como também os métodos e algoritmos implementados até o momento.

No Capítulo 4 é descrita a tarefa de clusterização no processo de *data mining*, bem como seus diferentes métodos. O método de lógica *fuzzy* para clusterização e alguns exemplos de algoritmos que o implementam são abordados no Capítulo 5.

O algoritmo Gustafson-Kessel, desenvolvido nesta pesquisa no módulo de clusterização da *Shell* Orion, é abordado detalhadamente no Capítulo 6. Alguns exemplos de aplicações deste algoritmos são comentados no Capítulo 7.

No Capítulo 8 é descrito o trabalho desenvolvido, a modelagem matemática do algoritmo Gustafson-Kessel e os resultados obtidos.

Por fim, tem-se a conclusão desta pesquisa, onde também são comentadas algumas sugestões para trabalhos futuros.

2 *DATA MINING*

O avanço tecnológico dos modelos de armazenamento possibilitou o aumento da capacidade de reter informações. No entanto, arquivar estes dados sem inteligência não produz o conhecimento necessário para a organização se manter no mercado competidor existente (BERRY; LINOFF, 2004).

A fim de analisar estes dados, métodos estatísticos são utilizados pelas instituições com o objetivo de descobrir conhecimento nestas bases. Porém, este processo pode apresentar algumas limitações se levar em consideração características como (KANTARDZIC, 2003):

a) necessidade de pessoal especializado para execução dos métodos;
b) diferentes tipos de informação torna sua análise trabalhosa;
c) grande volume de dados aumenta o tempo de execução dos cálculos e inviabiliza a utilização do conteúdo total da base de dados.

Considerando estes aspectos, surgiu o conceito de *Data Mining* (DM) com o objetivo de compreender, analisar e gerar conhecimento que auxilie na tomada de decisões das diferentes organizações.

A expressão *data mining* também é conhecida como descoberta de conhecimento em bases de dados, denominada de *Knowlegde Discovery in Databases* (KDD). No entanto, o termo DM é confundido como se fosse todo o processo KDD, por ser a sua principal etapa, sendo responsável pela busca de informações relevantes e criação efetiva do conhecimento (HAN; KAMBER, 2001).

Kantardzic (2003) conceitua *data mining* como um processo interativo o qual é definido pela exploração de novos conhecimentos. Independente dos métodos

utilizados, seu objetivo é encontrar novas e valiosas informações ocultas em grandes bases de dados.

O DM envolve busca de conhecimento de forma prática e não teórica, a fim de encontrar padrões nas bases de dados e reconhecer novas informações que possam auxiliar na descoberta de conhecimento (WITTEN; FRANK, 2005).

A fim de atender a necessidade de descoberta de conhecimento nas bases de dados das instituições, alguns destes métodos de *data mining* começaram a ser incorporados em ferramentas comercias. Além disso, alguns fatores auxiliaram as organizações a implantarem este tipo de tecnologia (BERRY; LINOFF, 2004):

a) **alta produção de dados**: organizações como bancos, supermercados, empresas de cartões de créditos e telefonia produzem por dia milhões de dados. Estes dados precisam ser analisados a fim de extrair informações que possam auxiliar as tomadas de decisões das instituições;
b) **centralização das informações**: os dados produzidos pelas organizações estão sendo armazenados em bancos de dados agrupados, denominados de *data warehouses*, facilitando a execução dos métodos de DM;
c) **relacionamento com clientes**: as empresas estão preocupadas com os interesses de seus consumidores. Considerando isto, as informações sobre seus clientes se tornaram importantes para melhorar e aproximar esta relação.

No entanto, o processo de descoberta de conhecimento em bases de dados envolve diversos tipos de aplicações, dependendo dos objetivos de cada instituição em relação a análise de suas informações.

2.1 DESCOBERTA DE CONHECIMENTO EM BASES DE DADOS

O KDD origina-se de diversas áreas como inteligência artificial, estatística, aprendizado de máquina, conhecimento de padrões e banco de dados. Considerando isto, são muitas as atividades relacionadas ao processo de KDD, e podem ser organizadas em três grandes grupos (GOLDSCHMIDT; PASSOS, 2005):

a) **desenvolvimento de tecnologias**: compreende as ferramentas, tecnologias, aprimoramento e desenvolvimento de algoritmos que possam auxiliar na descoberta de novos conhecimentos;
b) **execução das tarefas de KDD**: consiste nas atividades relacionadas a busca efetiva do conhecimento;
c) **aplicação dos resultados**: refere-se a aplicação dos resultados obtidos na área específica em que foi executado o processo de KDD.

A descoberta de novos conhecimentos em bases de dados é um processo interativo por exigir como responsável pela operação a atuação do homem, e iterativo por possibilitar repetições parciais ou integrais do processo de KDD, a fim de atingir resultados satisfatórios por meio de aprimoramentos sucessivos (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

O processo de KDD envolve algumas etapas as quais precisam ser executadas corretamente a fim de maximizar a descoberta de conhecimento em bases de dados.

2.1.1 Etapas da Descoberta de Conhecimento em Bases de Dados

Conforme Fayyad, Piatetsky-Shapiro e Smyth (1996) o processo de KDD envolve tarefas essenciais na descoberta de conhecimento (Figura 1):

a) **pré-processamento**: a fim de melhorar o processo de KDD, os dados precisam estar formatados e organizados. Esta etapa compreende três passos:

- **seleção**: compreende a identificação de quais informações serão utilizadas efetivamente no processo de KDD. Estes dados são combinados de acordo com o objetivo desejado e organizados em tabelas, onde o usuário pode interagir eliminando linhas ou colunas e executando junções de tabelas. As junções podem ocorrer de duas formas: junção direta, onde todos os dados são incluídos em uma nova tabela, sem ser feita uma análise crítica das variáveis e casos que possam auxiliar no processo de KDD, ou junção orientada, onde os especialistas no domínio da aplicação e no KDD, selecionam as variáveis e atributos relevantes que possam contribuir no processo de descoberta de conhecimento (GOLDSCHMIDT; PASSOS, 2005),
- **pré-processamento**: evita que dados imprecisos ou incorretos interfiram no processo de *data mining* (HAN; KAMBER, 2001). Esta etapa realiza operações básicas como: remoção de ruídos[5]; coleta de informações necessárias para estimular ou modelar um ruído; escolha de estratégias para manipular campos de dados ausentes; agrupar e

[5] Dados os quais possuem erros ou valores diferentes do esperado (HAN; KAMBER, 2001).

organizar dados relevantes (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996),

- **transformação**: consiste em organizar as informações em um formato padrão. Esta etapa modifica os dados numéricos por meio de operadores matemáticos ou lógicos, e exclui atributos redundantes e inconsistentes, a fim de facilitar a execução das tarefas de *data mining* (BERRY; LINOFF, 2004);

b) ***data mining***: é a principal etapa do KDD, pois é neste momento onde é realizada a busca efetiva pelo conhecimento em bases de dados (GOLDSCHMIDT; PASSOS, 2005). Nesta etapa ocorre o processo de descoberta de padrões, construção de modelos e exploração visual dos dados. Após isto, é possível executar a interpretação e avaliação das informações obtidas (KANTARDZIC, 2003).

c) **pós-processamento**: também denominado de interpretação, envolve a visualização, análise e interpretação do conhecimento gerado pela etapa de *data mining* (GOLDSCHMIDT; PASSOS, 2005). As informações adquiridas podem auxiliar na tomada de decisão das organizações ou buscar novas alternativas de descoberta de conhecimento, por meio da repetição de etapas já executas anteriormente (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

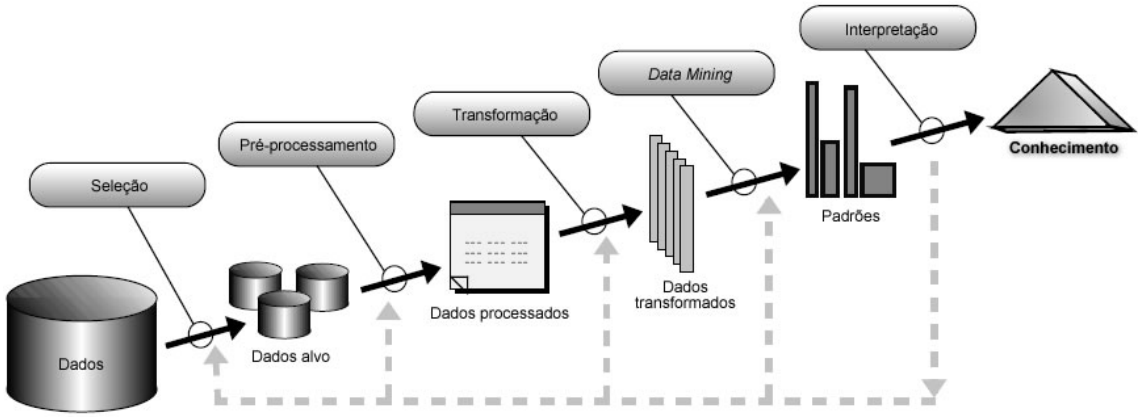

Figura 1. Etapas do processo de descoberta de conhecimento
Fonte: Adaptado de FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. (1996)

A descoberta de conhecimento em bases de dados é um processo complexo, o qual não depende apenas do computador utilizado, mas também da preparação dos dados pelo usuário e interpretação dos resultados obtidos, conforme a área de aplicação do KDD.

2.1.2 Aplicações da Descoberta de Conhecimento em Bases de Dados

O processo de KDD está sendo utilizado em diversas áreas, como: Administração, Educação, Medicina, Telecomunicações, entre outros. Berry e Linoff (2004) descrevem alguns exemplos interessantes de aplicações de *data mining*:

a) **supermercado como centro de informações**: supermercados os quais possuem cartões de desconto, registram as compras de seus clientes e salvam estas informações em bancos de dados. Estes dados podem ser utilizados em uma ferramenta de KDD e por meio do conhecimento gerado, oferecer a determinados clientes melhores ofertas ou auxiliar fornecedores com técnicas de incentivo ao uso de seus produtos;

b) **negócios baseados em recomendações**: empresas que possuem comércio eletrônico, podem fazer pesquisas sobre seus produtos e

realizar o balanço de vendas a determinados compradores. Após isto, pode-se aplicar a estes dados tarefas de *data mining*, a fim de verificar as preferências de seus clientes e oferecer em suas páginas principais produtos de acordo com a característica de compra de seus consumidores;

c) **reter bons consumidores**: o processo de *data mining* pode ser utilizado em qualquer negócio onde clientes estão livres para mudar de fornecedor por este possuir preços mais baixos, e onde a concorrência é alta. Considerando isto, efetua-se a análise dos consumidores que estão vulneráveis a trocar de fornecedor e cria-se um plano de retenção por meio do conhecimento gerado, atingindo estes clientes;

d) **afastar maus consumidores**: em organizações prestadoras de serviços de empréstimos, é difícil definir com precisão o perfil de seus clientes. Ao aplicar uma ferramenta de KDD a estes dados, a instituição pode analisar o histórico de seus clientes e classificá-los em grupos. O conhecimento gerado pode auxiliar a instituição a oferecer empréstimos de acordo com o perfil de cada consumidor.

Considerando as diversas aplicações do processo de *data mining*, deve-se compreender suas várias tarefas, as quais podem ser utilizadas em conjunto ou separadamente independente da área de aplicação, a fim de maximizar o conhecimento gerado.

2.2 TAREFAS DE *DATA MINING*

O *data mining* é composto por várias tarefas. Cada uma delas possui uma característica única, assim como os métodos e algoritmos específicos para determinados problemas. Estas tarefas podem ser classificadas em duas categorias (KANTARDZIC, 2003):

a) **preditiva**: consiste em prever valores de um determinado atributo com base em outros dados. A classificação, regressão e previsão de séries temporais são exemplos de tarefas preditivas;
b) **descritiva**: compreende a descoberta de padrões para auxiliar a compreensão humana dos dados. A associação e clusterização são exemplos de tarefas descritivas.

Muitos problemas podem ser solucionados com tarefas específicas de *data mining*. No entanto, a utilização conjunta delas pode auxiliar e melhorar o desempenho da descoberta de conhecimento em bases de dados (BERRY; LINOFF, 2004).

A seguir estão descritos os objetivos das principais tarefas de *data mining*, bem como exemplos de suas aplicações:

a) **associação**: busca relacionamentos e semelhanças entre um determinado grupo de dados. Esta tarefa possui a característica de encontrar regras de associação freqüentes, dependendo das configurações utilizadas pelo especialista no domínio da aplicação (WITTEN; FRANK, 2005). Um exemplo de utilização desta tarefa seria uma empresa associar quais produtos são vendidos em conjunto aos seus clientes;
b) **classificação**: encontra características de um conjunto de dados a fim de diferenciá-los uns dos outros. De acordo com estas informações, a tarefa

cria regras de classificação que podem ser utilizadas em outro grupo de dados (HAN; KAMBER, 2001). Esta tarefa pode ser aplicada, por exemplo, em uma avaliação de crédito a um cliente, considerando sua idade e seus pagamentos anteriores;

c) **clusterização**: também conhecida como agrupamento, objetiva particionar a base de dados em grupos, denominados de *clusters*, onde cada elemento integrante seja similar aos outros que se encontram no mesmo grupo, diferenciando-os dos demais *clusters* (KANTARDZIC, 2003). Um exemplo de utilização desta tarefa seria uma organização que divide seus clientes de acordo com suas preferências;

d) **previsão de séries temporais**: baseado no comportamento passado dos dados, verifica e estima valores futuros (BERRY; LINOFF, 2004). Uma empresa que prevê quais clientes podem deixar de comprar seus produtos em alguns meses pode ser citado como exemplo desta tarefa;

e) **regressão**: similar a tarefa de classificação, é diferenciada por permitir apenas atributos numéricos. A regressão busca funções matemáticas que mapeiem em valores reais um conjunto de dados (GOLDSCHMIDT; PASSOS, 2005). Um exemplo da utilização desta tarefa seria a visualização de riscos de determinados investimentos.

A eficiência e a qualidade do resultado das tarefas de *data mining* dependem também do conhecimento do especialista sobre sua área de aplicação. Isto é essencial para a escolha correta destas tarefas, de acordo com os objetivos do usuário em relação a descoberta de conhecimento (KANTARDZIC, 2003).

Cada tarefa de *data mining* possui diferentes métodos de implementação, dependendo do conhecimento que se deseja extrair da base de dados (WITTEN; FRANK, 2005). A seguir estão descritos alguns métodos bastante utilizados:

a) **algoritmos genéticos**: inspirados na teoria da evolução[6] e reprodução genética[7], estes algoritmos desenvolvem soluções satisfatórias para problemas complexos, por meio de funções de reprodução entre conjuntos de dados e mutações (RUSSELL; NORVIG, 2004);

b) **árvores de decisão**: são estruturas utilizadas para dividir grande quantidade de dados em conjuntos menores, por meio de regras de decisão. As sucessivas divisões criam estruturas com membros cada vez mais específicos (BERRY; LINOFF, 2004);

c) **lógica *fuzzy***: diferente da lógica tradicional, a lógica *fuzzy* pode representar valores entre zero e um, o que possibilita raciocínios aproximados. Esta abordagem pode descrever conceitos imprecisos ou incertos, como por exemplo, verificar se a temperatura do ambiente está alta ou se uma pessoa possui estatura baixa (COX, 2005);

d) **redes neurais**: as redes neurais artificiais são modelos matemáticos baseados na estrutura do cérebro e no funcionamento dos neurônios humanos. Estas redes possuem algumas características similares ao homem, como aprendizado por meio dos dados e propagação de informações (GOLDSCHMIDT; PASSOS, 2005).

Estas tarefas e métodos de *data mining* encontram-se implementados em ferramentas, denominadas de *Shells*, as quais são utilizadas a fim de auxiliar na

---

[6] Descreve que as características herdadas de outras gerações podem se modificar ao longo do tempo. As modificações podem ser vantajosas ou problemáticas (BÄCK; FOGEL; MICHALEWICZ, 1997).

[7] União de genes que resulta em um descendente contendo características de seus progenitores (CECIL; GOLDMAN; AUSIELLO, 2005).

descoberta de padrões e relações relevantes. A Tabela 1 apresenta uma visão geral das ferramentas disponíveis.

Tabela 1. Exemplos de ferramentas de *data mining*

Nome	**Tipo**	**Tarefas Disponíveis**	**Fabricante**
SPSS/Clementine	Comercial	Classificação, Regras de Associação, Clusterização, Seqüências, Detecção de Desvios	SPSS Inc. www.spss.com
PolyAnalist	Comercial	Classificação, Regressão, Regras de Associação, Clusterização, Sumarização, Detecção de Desvios	Megaputer Intelligence www.megaputer.com
Darwin	Comercial	Classificação	Thinking Machines http://en.wikipedia.org/wiki/Thinking_Machines
Intelligent Miner	Comercial	Classificação, Regras de Associação, Seqüências, Clusterização, Sumarização	IBM Corp. www.ibm.com
WizRule	Comercial	Sumarização, Classificação, Detecção de Desvios	WizSoft Inc. www.wizsoft.com
Bramining	Comercial	Classificação, Regras de Associação, Regressão, Sumarização	Graal Corp. www.graal-corp.com.br
SAS Enterprise Miner	Comercial	Classificação, Regras de Associação, Regressão, Sumarização	SAS Inc. www.sas.com
Oracle *Data Mining*	Comercial	Classificação, Regressão, Associação, Clusterização, Mineração de Textos	Oracle www.oracle.com
Weka	Gratuita	Classificação, Regressão, Regras de Associação	University of Waikato www.cs.waikato.ac.nz
Orion *Data Mining Engine*	Gratuita	Classificação, Associação, Clusterização	Grupo de Pesquisa em Inteligência Computacional Aplicada – UNESC www.unesc.net/ica

Fonte: Adaptado de GOLDSCHMIDT, R.; PASSOS, E. (2005)

As *Shells* de *data mining* em sua maioria são comerciais, tornando-se inviável para determinadas organizações. Um exemplo de ferramenta gratuita é a *Shell* Orion *Data Mining Engine*, projeto iniciado em 2005 que vem sendo desenvolvida por acadêmicos, abordando os conceitos relacionados ao processo de *data mining* apresentados nos tópicos anteriores.

A *Shell* Orion *Data Mining Engine* possui algumas tarefas e métodos implementados e suas características, bem como funcionalidades disponíveis, serão apresentadas no Capítulo 3.

3 *SHELL* ORION *DATA MINING ENGINE*

A *Shell* Orion, que aplica diferentes tarefas e métodos de *data mining*, é um projeto implementado por acadêmicos e mantido pelo Grupo de Pesquisa em Inteligência Computacional Aplicada, do Curso de Ciência da Computação, na Universidade do Extremo Sul Catarinense.

Até o momento, a *Shell* Orion possui implementado os módulos de associação, classificação e clusterização. Cada método foi desenvolvido por um acadêmico como Trabalho de Conclusão de Curso (TCC). A Tabela 2 apresenta todos os algoritmos implementados até o momento.

Tabela 2. Algoritmos implementados na *Shell* Orion *Data Mining Engine*

Ano	Tarefa	Algoritmo	Método	Atributos	Referência
2005	Associação	*Apriori*	Regras de associação	Numéricos	(CASAGRANDE, 2005)
2005	Classificação	ID3	Árvores de decisão	Nominais	(PELEGRIN, 2005)
2007	Classificação	CART	Árvores de decisão	Nominais e numéricos	(RAIMUNDO, 2007)
2007	Clusterização	*K-Means*	Particionamento	Numéricos	(MARTINS, 2007)
2007	Clusterização	*Kohonen*	Redes neurais	Numéricos	(BORTOLOTTO, 2007)

A ferramenta está sendo desenvolvida por meio da linguagem de programação Java, devido algumas de suas características como: ser multiplataforma, ter a possibilidade de reutilização do código e pelas ferramentas de programação serem gratuitas (PELEGRIN, 2005).

A fim de possibilitar a execução dos algoritmos desenvolvidos, como também outras funcionalidades da *Shell* Orion, foi desenvolvida uma interface principal com o objetivo de centralizar todas as funções da ferramenta.

3.1 INTERFACE DA *SHELL* ORION

A interface principal da ferramenta foi desenvolvida a fim de facilitar a interação com o usuário e simplificar a inserção de novos métodos. A Figura 2 apresenta a interface principal da *Shell* Orion.

Figura 2. Interface principal da *Shell* Orion *Data Mining Engine*
Fonte: BORTOLOTTO, L. (2007)

A *Shell* Orion objetiva proporcionar ao usuário diferentes opções de descoberta de conhecimento e vários tipos de conexões com banco de dados. A nova interface possibilita a utilização de qualquer *driver* de conexão, precisando apenas do *driver* do banco de dados instalado na ferramenta.

Após a instalação correta do *driver* do banco de dados, suas respectivas configurações de conexão devem ser fornecidas na interface de *Cadastro de Conexões*, acessada por meio do menu *Arquivo*, *Conexões*. O usuário deve informar o nome, *driver* e endereço do banco correspondente ou selecionar um existente no menu conexão. A Figura 3 mostra um exemplo de conexão com o banco de dados *freeware* PostgreSQL[8], que pode ser obtido por meio do *site* do fabricante.

[8] PostgreSQL disponível em (http://www.postgresql.org).

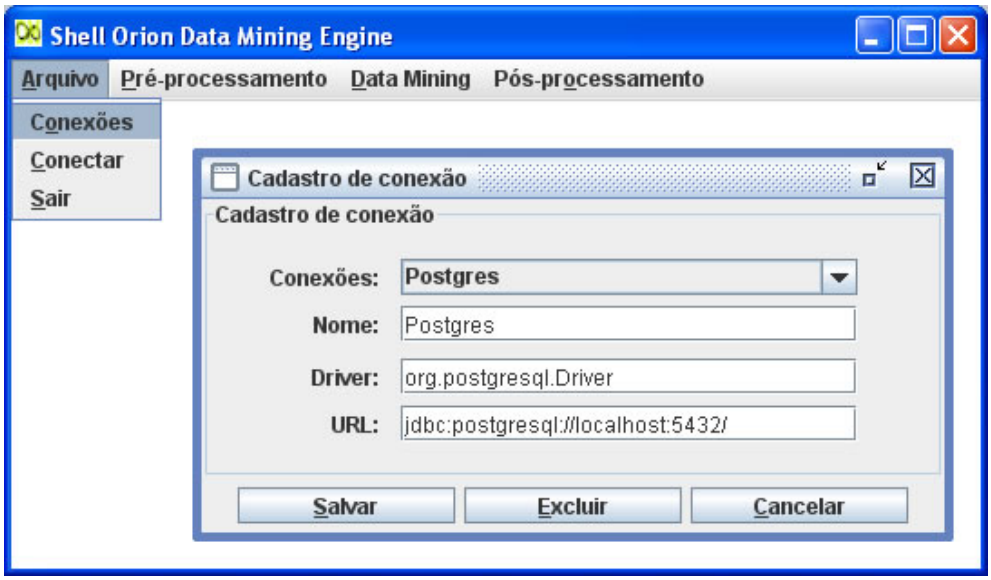

Figura 3. Cadastro de conexão com o banco PostgreSQL
Fonte: Adaptado de BORTOLOTTO, L. (2007)

Ao configurar o *driver* de forma correta, é possível executar a conexão com o banco de dados escolhido, por meio da interface de *Conexão* (Figura 4), acessada no menu *Arquivo*, *Conexões*. O usuário deve informar o nome da base de dados, seu nome de usuário e sua senha. Após a conexão, pode-se selecionar o nome da tabela a fim de executar as tarefas de *data mining*.

Figura 4. Conexão como banco PostgreSQL
Fonte: Adaptado de BORTOLOTTO, L. (2007)

Efetuada a conexão com a base de dados, o usuário pode escolher no menu entre as etapas de pré-processamento, *data mining* e pós-processamento, considerando que o pré e pós-processamento ainda não estão implementados. No entanto, os métodos de *data mining* já desenvolvidos podem ser executados por meio da *Shell* Orion.

3.2 TAREFAS DE *DATA MINING* DA *SHELL* ORION

A *Shell* Orion possui até o momento as tarefas de associação, classificação e clusterização implementadas. Cada uma delas está disponível em módulos separados, onde cada um possui suas próprias configurações de descoberta de conhecimento, conforme as especificidades de cada algoritmo.

No modulo de associação (Figura 5) encontra-se implementado o algoritmo *Apriori*, método clássico para encontrar os elementos mais freqüentes e criar regras de associação (HAN; KAMBER, 2001).

Este algoritmo processa por meio de várias iterações os *itemsets* (conjunto de itens) mais comuns da base de dados. Cada iteração possui duas etapas: a primeira consiste na geração dos candidatos, ou seja, itens que possivelmente são freqüentes; a segunda se refere à contagem e seleção destes candidatos. As iterações terminam quando não existem mais candidatos (KANTARDZIC, 2003).

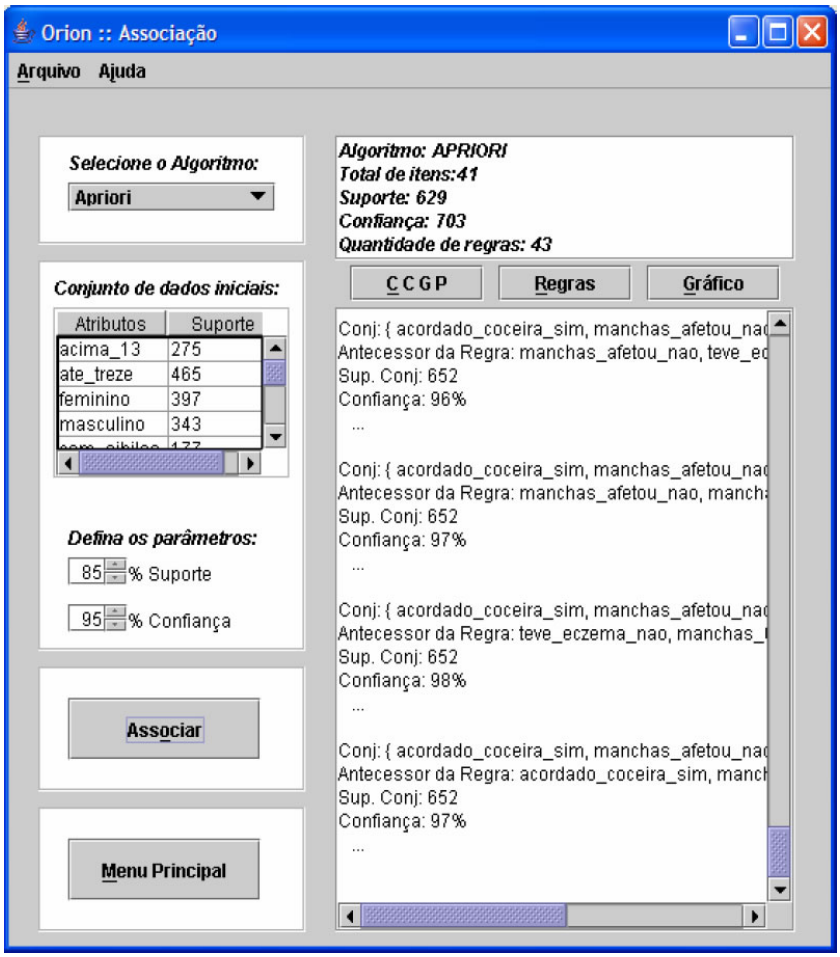

Figura 5. Módulo de associação da *Shell* Orion *Data Mining Engine*
Fonte: CASAGRANDE, D. (2005)

Já o módulo de classificação (Figura 6) possui dois métodos implementados, entre eles o algoritmo ID3, caracterizado por gerar árvores de decisão. Este algoritmo utiliza o critério de ganho de informação para classificar cada atributo, onde os melhores classificados se tornam os nós da árvore (HAN; KAMBER, 2001).

As árvores de decisão são uma das possíveis maneiras de representar o conhecimento gerado pela tarefa de classificação. Segundo Witten e Frank (2005) este tipo de representação pode facilitar a visualização e o entendimento das informações pelo usuário, conforme seus objetivos em relação à descoberta de conhecimento na base de dados.

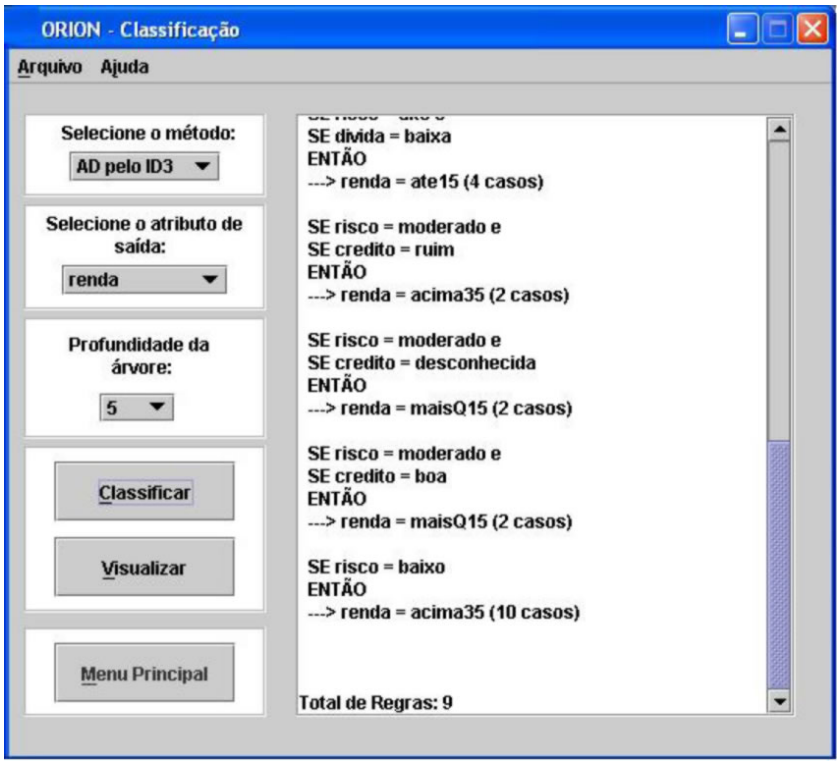

Figura 6. Módulo de classificação pelo algoritmo ID3
Fonte: PELEGRIN, D. (2005)

O módulo de classificação da *Shell* Orion possui implementada esta função a qual permite a visualização da árvore criada (Figura 7), a fim de melhorar a representação do atributo que mais influenciou na criação do modelo de conhecimento (PELEGRIN, 2005).

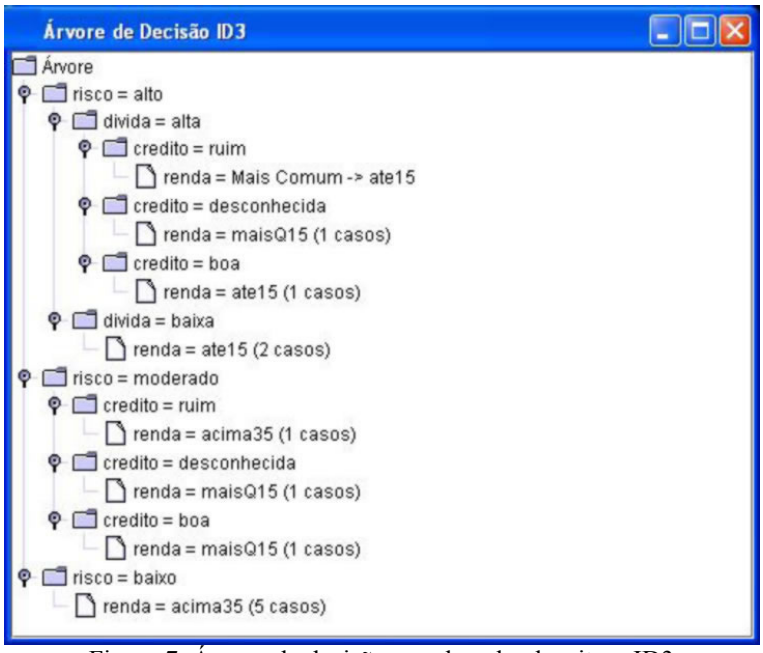

Figura 7. Árvore de decisão gerada pelo algoritmo ID3
Fonte: PELEGRIN, D. (2005)

O modelo de árvores de decisão também está presente no outro método disponível no módulo de classificação. O algoritmo CART divide os nós da árvore de acordo com a qualidade dos dados, por meio de critérios pré-selecionados. Este processo previne e controla o crescimento exagerado da árvore, simplificando o conhecimento gerado pelo algoritmo (HAND; MANNILA; SMYTH, 2001).

Dentre os critérios disponíveis, optou-se pelo desenvolvimento do critério de Gini, pelo fato deste selecionar com maior precisão os divisores puros da árvore e por conseqüência, apresentar melhores resultados (RAIMUNDO, 2007).

Após a utilização do algoritmo CART por meio da *Shell* Orion, é possível visualizar as regras de classificação geradas (Figura 8), como também a árvore de decisão criada pelo método (Figura 9).

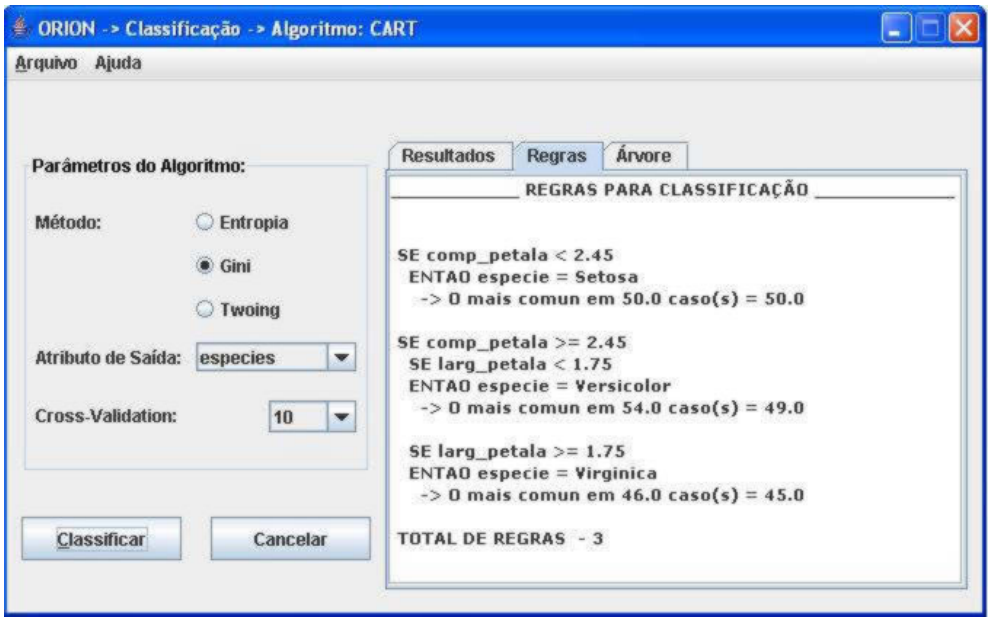

Figura 8. Regras geradas pelo algoritmo CART
Fonte: RAIMUNDO, L. (2007)

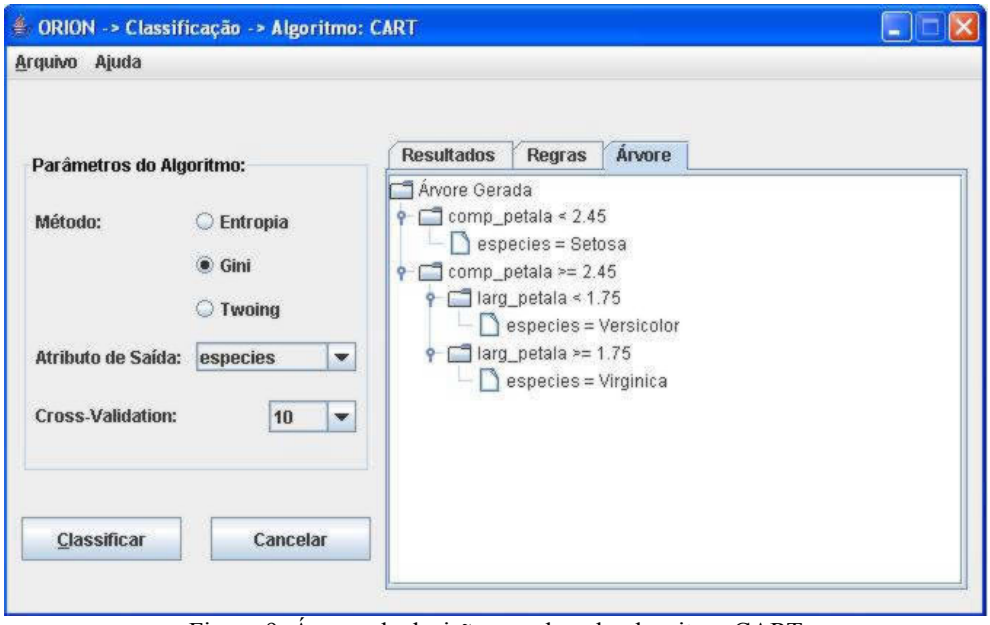

Figura 9. Árvore de decisão gerada pelo algoritmo CART
Fonte: RAIMUNDO, L. (2007)

Além da associação e classificação, *a Shell* Orion também possui o módulo de clusterização, onde tem-se disponível um popular método desta tarefa. O algoritmo *K-Means* seleciona de forma aleatória *k* pontos de informações na base de dados, que serão os elementos centrais dos *clusters*[9]. Após isto, o método calcula as distâncias dos

[9] Em relação a tarefa de clusterização, um *cluster* é considerado um grupo de dados ou informações que possuem características em comum (COX, 2005).

outros dados em relação aos pontos centrais e agrupa estes dados em grupos que possuem propriedades similares (HASTIE; TIBSHIRANI; FRIEDMAN, 2001).

O *K-Means* é o algoritmo mais comum entre os métodos de clusterização, por possuir funções simples e de fácil compreensão. O "*K*" em seu nome se refere ao número de elementos centrais que o algoritmo seleciona durante sua execução (BERRY; LINOFF, 2004).

Ao utilizá-lo por meio da *Shell* Orion, o usuário deve informar o número de *clusters* que deseja criar, variáveis de entrada, tipo do cálculo de distância (euclidiana[10] ou *city-block*[11]) e variável de saída.

Os resultados são disponibilizados na mesma tela (Figura 10), como também a função de gerar um arquivo SQL para utilização em outras tarefas de *data mining* da *Shell* Orion, como também em outras ferramentas.

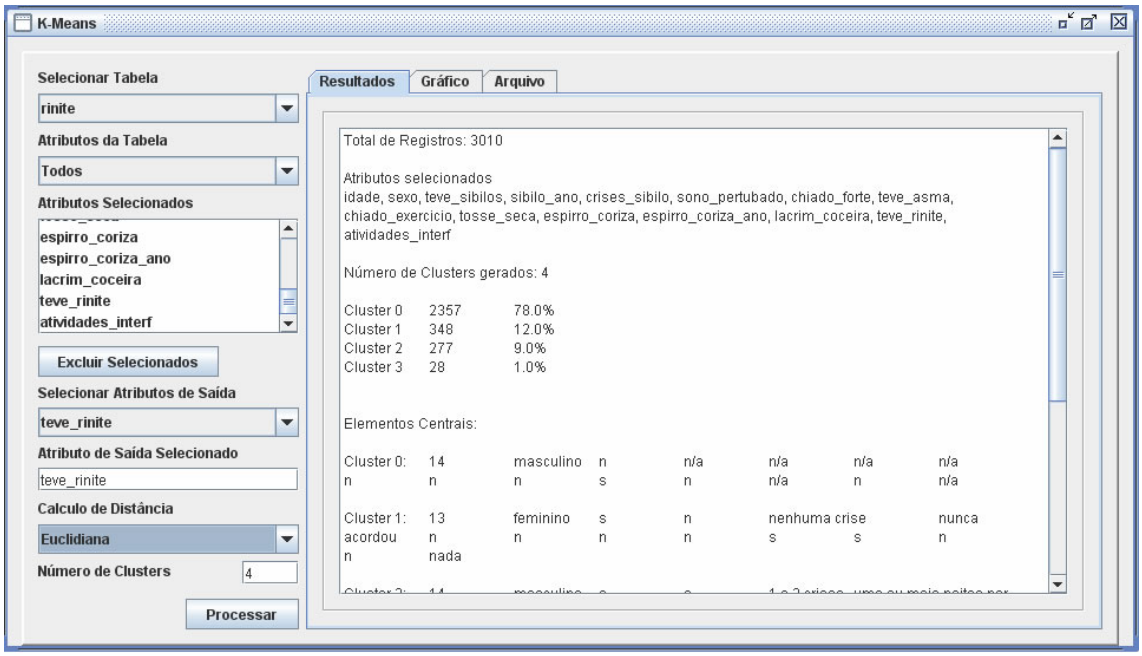

Figura 10. Resultados obtidos por meio do algoritmo *K-Means*
Fonte: MARTINS, D. (2007)

[10] Compreende o somatório das raízes quadradas das diferenças de distância entre dois pontos (BEZDEK et al, 2005).

[11] Similar a distância euclidiana, no entanto realiza apenas o somatório das diferenças de distância entre dois pontos. Em situações complexas, pode ter pouca precisão (GOLDSCHMIDT; PASSOS, 2005).

O método de redes neurais também está presente no módulo de clusterização da *Shell* Orion, por meio do algoritmo de *Kohonen*, que se caracteriza por pertencer a classe das redes neurais auto-organizáveis, onde o treinamento não é supervisionado[12] (GOLDSCHMIDT; PASSOS, 2005).

Este método utiliza os dados como variáveis de entrada e conforme o aprendizado da rede, o algoritmo organiza estas informações em grupos de acordo com sua similaridade (KANTARDZIC, 2003).

Após a utilização do algoritmo por meio da *Shell* Orion, os resultados podem ser obtidos em forma gráfica (Figura 11), pela visualização dos grupos (Figura 12) e informações estatísticas (Figura 13). A função de exportar um arquivo SQL para utilização em outras tarefas de *data mining* também está presente neste algoritmo.

Figura 11. Gráfico gerado pelo algoritmo *Kohonen*
Fonte: BORTOLOTTO, L. (2007)

[12] Os neurônios da rede não sofrem restrições de aprendizagem durante o treinamento (LUGER, 2004).

Figura 12. Grupos gerados pelo algoritmo *Kohonen*
Fonte: BORTOLOTTO, L. (2007)

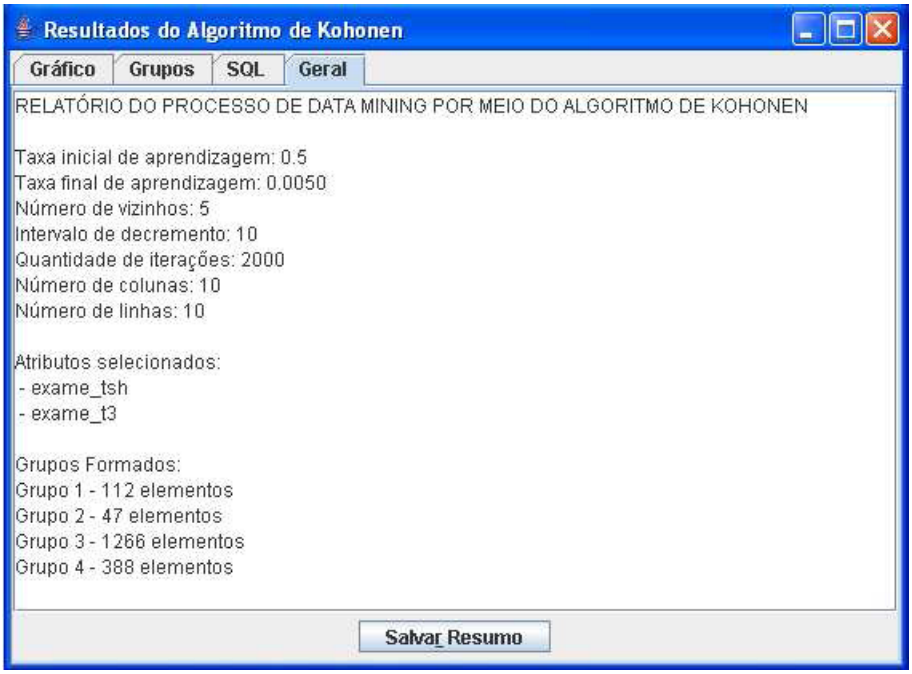

Figura 13. Resumo da clusterização pelo algoritmo *Kohonen*
Fonte: BORTOLOTTO, L. (2007)

Além destes métodos, outros estão em fase de desenvolvimento, como por exemplo o algoritmo de classificação C4.5 e a tarefa de pré-processamento do processo de KDD, com o objetivo de aumentar as funcionalidades disponíveis na *Shell* Orion *Data Mining Engine*.

Considerando isto, esta pesquisa consiste na implementação do método de lógica *fuzzy* pelo algoritmo Gustafson-Kessel para a tarefa de clusterização da *Shell* Orion, com o objetivo de expandir as diferentes maneiras de descoberta de conhecimento da ferramenta.

A tarefa de clusterização tem como principal objetivo agrupar os diferentes tipos de dados disponíveis em grupos, chamados de *clusters*, onde cada grupo se diferencia dos outros em função de algumas características específicas.

4 A TAREFA DE CLUSTERIZAÇÃO NO PROCESSO DE *DATA MINING*

O processo de agrupar dados que possuem características similares em diferentes grupos é chamado de clusterização. Esta tarefa não necessita de exemplos ou classes pré-definidas para ser executada, pois a criação dos grupos acontece conforme são detectadas as informações que os dados possuem em comum (BERRY; LINOFF, 2004).

Os grupos criados pela tarefa são identificados como *clusters* e pelo fato da clusterização unir os diferentes tipos de dados em grupos com características semelhantes, esta tarefa também é conhecida como agrupamento (GOLDSCHMIDT; PASSOS, 2005).

O processo de clusterização é uma importante atividade humana. Desde criança, o homem aprende a distinguir objetos ou animais por tipo, como por exemplo diferenciar um gato de um cachorro. Além disto, o processo de agrupamento é amplamente utilizado em áreas como: processamento de imagens, análise de dados, reconhecimento de padrões, entre outros (HAN; KAMBER, 2001).

A clusterização é explorada com o objetivo de facilitar a identificação de padrões complexos, por meio de diferentes técnicas que auxiliem o reconhecimento de características ocultas dentro das bases de dados, considerando suas diversas áreas de aplicação (BEZDEK et al, 2005).

Além disso, a importância do agrupamento dos dados no processo de *data mining* se deve ao fato de que após sua execução, outras tarefas como classificação e sumarização[13] podem ser aplicadas ao seu resultado, com o objetivo de aumentar a eficiência da descoberta de conhecimento na base de dados (SERRA, 2002).

---

[13] Considerando uma base de dados, a sumarização define uma descrição resumida deste conteúdo a fim de facilitar sua compreensão (GOLDSCHMIDT; PASSOS, 2005).

O objetivo principal da tarefa de clusterização é particionar um conjunto de dados em *clusters*. No entanto, a divisão destes dados deve seguir as seguintes propriedades (HÖPPNER et al, 1999):

a) **homogeneidade nos *clusters***: os dados que pertencem ao mesmo grupo devem ser o mais similar possível;

b) **heterogeneidade entre *clusters***: os dados que pertencem a grupos distintos devem ser o mais diferente possível.

A fim de atingir este objetivo, a clusterização analisa cada dado da base por meio de critérios de similaridade ou dissimilaridade. Porém, a performance da tarefa depende diretamente do tamanho do conjunto de informações. Se este for muito extenso, pode ocorrer o aumento do tempo de execução da tarefa (HASTIE; TIBSHIRANI; FRIEDMAN, 2001).

Goldschmidt e Passos (2005) também afirmam que o tamanho da base interfere na detecção dos *clusters*. Quanto mais atributos existirem, mais difícil será de encontrar os diferentes padrões entre os dados, pois alguns grupos podem estar dentro de outros *clusters*.

Mas o fato destes dados estarem dispostos em um espaço de grande dimensão pode não ser a principal dificuldade da clusterização. Algumas bases podem conter muitos dados e poucos padrões. Já outras podem ter tantas características que nem os mais avançados algoritmos conseguem identificar com precisão seus padrões (BERRY; LINOFF, 2004).

Considerando isto, a clusterização é considerada uma área que exige um estudo detalhado para cada caso, onde as aplicações desenvolvidas possuem objetivos específicos. Mas alguns requerimentos básicos são necessários para a clusterização ter sucesso na identificação dos grupos (HAN; KAMBER, 2001):

a) **escalabilidade**: grande parte dos algoritmos consegue trabalhar com poucos dados. No entanto, em grandes bases de dados existe a necessidade de análise prévia dos dados e considerar apenas os atributos relevantes, a fim de obter resultados satisfatórios;

b) **habilidade de manipular diferentes tipos de dados**: alguns algoritmos conseguem apenas manipular dados numéricos. Considerando isto, é necessário executar o pré-processamento dos dados a fim de possibilitar a clusterização de diferentes tipos de dados, como binários e nominais;

c) **descobrir *clusters* com forma arbitrária**: muitos dos *clusters* encontrados pelos algoritmos são baseados na distância euclidiana, formando grupos de forma esférica e de tamanhos parecidos. Porém, os *clusters* podem ser de qualquer forma sendo importante que os algoritmos encontrem grupos de diferentes modelos;

d) **minimizar os requerimentos para determinação dos parâmetros de entrada**: parte dos algoritmos necessita de dados de entrada antes de entrar em execução. Estes parâmetros são difíceis de determinar e podem mudar completamente o resultado da tarefa. É preciso algoritmos que minimizem estes procedimentos a fim de melhorar o controle da qualidade do resultado;

e) **habilidade de manipular dados com ruídos**: em sua grande maioria, as bases de dados contêm dados imprecisos, desconhecidos ou errôneos. Alguns algoritmos, como por exemplo o *K-Means*, são sensíveis a estes dados e isto pode resultar em *clusters* de baixa qualidade;

f) **alta dimensão**: algoritmos conseguem manipular facilmente de duas a três camadas de dados e também é fácil para uma pessoa verificar a sua

qualidade. Mas o desafio é criar algoritmos que possam criar grupos com qualidade em bases de dados de grande dimensão;

g) **clusterização baseada em constantes**: algumas bases possuem dados que são constantes, nunca mudam. Os algoritmos desenvolvidos a fim de atender estas bases, precisam encontrar grupos que são baseados nestas constantes;

h) **ser interpretável e ter usabilidade**: não adianta um algoritmo possuir qualidade na detecção de *clusters* e não mostrar resultados compreensíveis. Os usuários esperam que os dados criados pelo algoritmo sejam claros e objetivos. Considerando isto, é importante verificar quanto os objetivos da aplicação podem interferir na escolha de um algoritmo.

Os algoritmos que conseguem atender a todas estas especificações são raros. Porém, podem ser utilizados algoritmos em conjunto a fim de melhorar os resultados proporcionados pela tarefa (WITTEN; FRANK, 2005).

A tarefa de clusterização possui alguns métodos específicos que podem ser aplicados com maior precisão em algumas bases de dados, de acordo com os objetivos da aplicação.

4.1 MÉTODOS DE CLUSTERIZAÇÃO

Existem diversos métodos de clusterização para utilização no processo de *data mining*. Cada método possui seus próprios algoritmos e podem ser implementados de acordo com os dados disponíveis e propósitos da aplicação (HAN; KAMBER, 2001).

O domínio da aplicação deve estudar os propósitos em relação a aplicação desta tarefa. Estas informações podem refletir diretamente no desempenho da clusterização como também na qualidade dos grupos gerados (BERRY; LINOFF, 2004).

A seguir são apresentados os principais métodos de clusterização que são classificados em diferentes categorias a fim de facilitar a compreensão dos algoritmos e sua implementação (HAN; KAMBER, 2001):

a) métodos hierárquicos;

b) métodos baseados em densidade;

c) métodos baseados em grade;

d) métodos baseados em modelos;

e) métodos de particionamento.

A fim de possibilitar a descrição, como também alguns exemplos de cada método, estes serão descritos separadamente.

4.1.1 Métodos Hierárquicos

Este tipo de abordagem divide a base de dados de forma hierárquica. De acordo com o tipo de técnica, os objetos vão sendo decompostos em camadas mais específicas, desenvolvendo uma estrutura de árvore como ilustra a Figura 14 (WITTEN; FRANK, 2005).

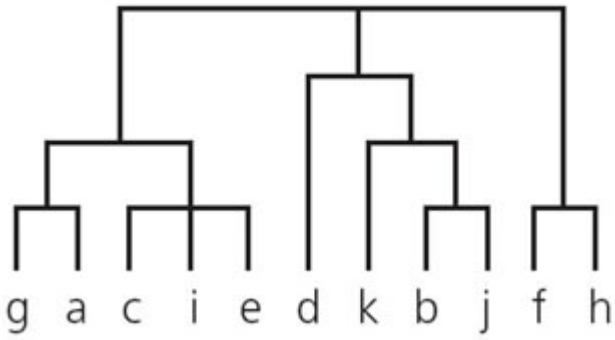

Figura 14. Exemplo de estrutura hierárquica
Fonte: WITTEN, I. H.; FRANK, E. (2005)

Os métodos hierárquicos podem ser divididos em dois grandes grupos (KANTARDZIC, 2003):

a) **abordagem aglomerativa**: também chamada de *bottom-up*, é caracterizada por iniciar das folhas para a raiz, onde cada *cluster* possui apenas um objeto. A cada iteração, os dados vão sendo agrupados de acordo com suas características em comum, até chegar ao topo da hierarquia ou quando o número de grupos definidos pelo usuário for atingido. Exemplo: o algoritmo *BIRCH*[14];

b) **abordagem divisiva**: conhecida também por *top-down*, é o oposto do *bottom-up*, ou seja, inicia-se da raiz para as folhas. Todos os elementos começam no mesmo *cluster* e em cada iteração, os objetos vão sendo separados até chegar a *clusters* que contenham somente um elemento ou uma condição de parada a fim de interromper a execução do algoritmo. Exemplo: o algoritmo *Chameleon*[15].

[14] Caracteriza-se por dividir os dados em uma estrutura de árvore e após isto melhorar a qualidade dos *clusters* por meio de outros algoritmos de clusterização, como por exemplo o algoritmo *K-Means* (HAN; KAMBER, 2001).

[15] Caracterizado por se adaptar a diferentes tipos de bases de dados, este algoritmo calcula a similaridade entre dois *clusters* a fim de encontrar novos padrões e aumentar a qualidade dos grupos (KANTARDZIC, 2003).

4.1.2 Métodos Baseados em Densidade

A maioria dos métodos de clusterização é baseada na distância entre os objetos. Porém, estes algoritmos são capazes de gerar *clusters* de um só tipo e têm dificuldade em encontrar grupos de diferentes formas (BEZDEK et al, 2005).

A noção de densidade, ou seja o espaço que um elemento ocupa, permite que sejam criados *clusters* de tamanhos variados. Conforme a densidade dos dados, os grupos crescem e mudam de forma. A Figura 15 ilustra o encadeamento de dados de acordo com o espaço que estes ocupam. O algoritmo *DBSCAN*[16] é um exemplo comum desta técnica (HAN; KAMBER, 2001).

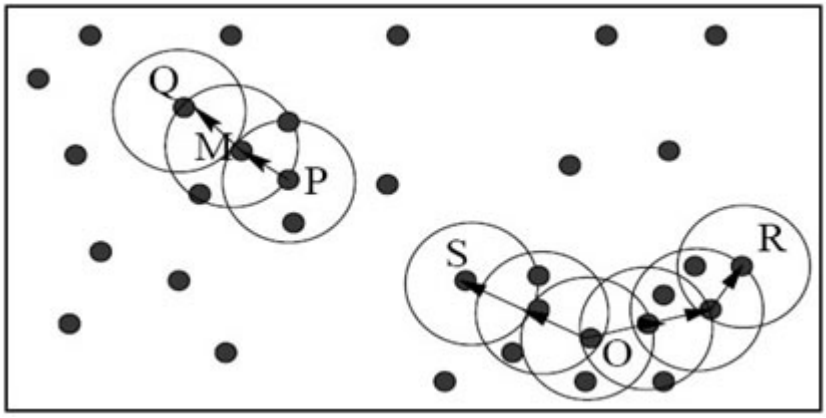

Figura 15. Exemplo de clusterização por densidade
Fonte: HAN, J.; KAMBER, M. (2001)

4.1.3 Métodos Baseados em Grade

Os métodos baseados em grade quantificam o espaço de um objeto em células que formam uma estrutura de grade. Todo processamento deste método é baseado na estrutura de grade criada (WITTEN; FRANK, 2005).

---

[16] Caracteriza-se por calcular a densidade que um elemento ocupa e agrupar os itens que se encontram dentro deste espaço. A cada elemento adicionado, sua densidade é calculada a fim de melhorar a qualidade dos grupos e gerar *clusters* de diferentes formas e tipos (PAL; MITRA, 2004).

Uma das vantagens desta técnica é a velocidade de execução dos algoritmos, que dependem apenas do número de estruturas criadas. Deste modo, o tamanho da base não interfere no seu tempo de processamento. O algoritmo STING[17] é um exemplo típico deste método (PAL; MITRA, 2004). A Figura 16 demonstra a construção das camadas, onde a primeira camada contém apenas um grupo e conforme o algoritmo executa os cálculos, as camadas internas começam a agrupar os dados com características em comum.

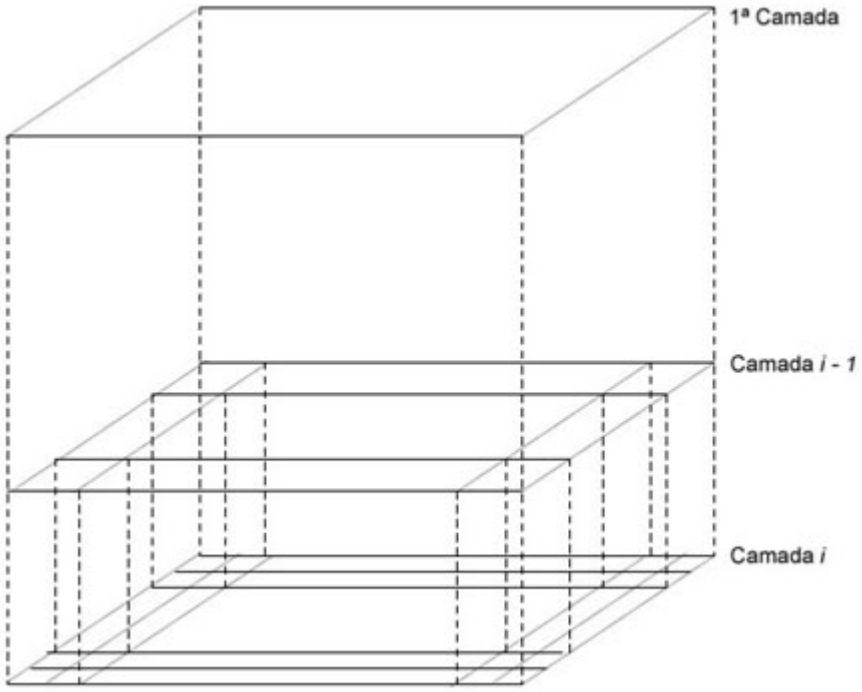

Figura 16. Método baseado em grade por meio do algoritmo STING
Fonte: Adaptado de HAN, J.; KAMBER, M. (2001)

4.1.4 Métodos Baseados em Modelos

Este tipo de método assimila um modelo a cada grupo que é criado a fim de detectar com maior precisão os tipos de *clusters*. Algumas técnicas baseadas em

[17] Divide os dados em células retangulares (*clusters*) e forma uma estrutura hierárquica, onde as primeiras camadas contêm características genéricas e as últimas células contém grupos com dados que possuem características em comum (HAN; KAMBER, 2001).

modelos conseguem determinar a melhor quantidade de grupos para execução da tarefa de clusterização (BERRY; LINOFF, 2004).

Os métodos baseados em modelos podem ser definidos em dois grupos (HAN; KAMBER, 2001):

a) **aproximação por estatística**: caracterizado pela aprendizagem sem supervisão, este tipo faz parte da clusterização conceitual, que utiliza métodos de estatística tradicionais, procurando primeiro pelas características dos grupos. Após isto, os dados são assimilados aos *clusters* que possuem informações em comum;

b) **aproximação por redes neurais**: mais eficiente na detecção de grupos, esta abordagem utiliza aprendizado não supervisionado para identificar exemplos de elementos de um determinado *clusters* e posteriormente, agrupar outros dados a este *cluster* baseados no modelo adquirido. Exemplo: algoritmo de *Kohonen*.

4.1.5 Métodos de Particionamento

Este método particiona os objetos de uma determinada base de dados. As partições representam os *clusters*, considerando que o número de *clusters* não deve ser maior que o número de objetos. A quantidade de grupos criada pelo método pode ser definida pelo usuário (KANTARDZIC, 2003).

Após definir o número de *clusters*, é criado um centro para cada grupo. De acordo com a distância entre um objeto e o centro de um grupo, o método assimila este elemento ao *cluster* mais próximo. Além disso, o particionamento utiliza um processo interativo de reordenação, onde objetos podem mudar de grupo, baseado nas

informações entre elementos de diferentes *clusters*. Isto aumenta a identificação de grupos que possuam somente dados com características semelhantes (GOLDSCHMIDT; PASSOS, 2005).

Este método é muito utilizado por ter simples implementação. Os algoritmos *K-Means* e o *k-medoids*[18] são exemplos deste método (HASTIE; TIBSHIRANI; FRIEDMAN, 2001). A Figura 17 exemplifica o uso do *K-Means*, onde na primeira imagem tem-se a primeira definição dos grupos e conforme o algoritmo executa suas iterações, os *clusters* começam a agrupar os elementos que possuem características em comum.

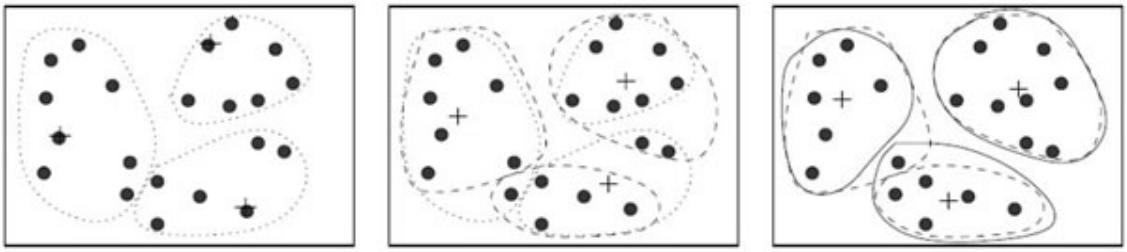

Figura 17. Método de particionamento por meio do algoritmo K-Means
Fonte: HAN, J.; KAMBER, M. (2001)

Han e Kamber (2001) afirmam que para o método de particionamento dividir uma base de dados em grupos, este deve seguir os seguintes requerimentos: cada *cluster* criado deve conter pelo menos um objeto; cada objeto deve pertencer a somente um *cluster*.

No entanto, existe a possibilidade de um elemento ser identificado a um grupo o qual não compartilhe suas características e isto prejudica a qualidade dos *clusters* detectados pelo método (BEZDEK et al, 2005).

[18] Similar ao *K-Means*, este algoritmo se diferencia por escolher um elemento central para cada grupo criado. Este registro deve possuir as características principais de seu *cluster*. Durante suas iterações, o algoritmo troca os elementos centrais de cada grupo, a fim de melhorar a qualidade dos mesmos (GOLDSCHMIDT; PASSOS, 2005).

Considerando os problemas descritos, foi desenvolvido o método de lógica *fuzzy* para a tarefa de clusterização, onde é abordado a possibilidade dos objetos pertencerem a vários *clusters* simultaneamente, afim de auxiliar na identificação correta dos grupos.

5 O MÉTODO DE LÓGICA *FUZZY* PARA A TAREFA DE CLUSTERIZAÇÃO

O agrupamento tem como principal objetivo a divisão de dados em grupos contendo características específicas. O sucesso da tarefa de clusterização depende da capacidade que o algoritmo utilizado possui para separar corretamente as informações e desenvolver *clusters* de alta qualidade (BERRY; LINOFF, 2004).

Os algoritmos tradicionais utilizam a lógica clássica (*crisp*) ao particionar os dados em grupos. Esta abordagem tem como principal característica a lógica booleana[19], onde as informações divididas em grupos podem pertencer somente a um determinado *cluster* (BEZDEK et al, 2005).

Em pequenas bases de dados, as informações podem ser particionadas de maneira simples e os *clusters* podem ser visualizados com clareza. A Figura 18 demonstra um pequeno grupo de dados em um espaço bidimensional, onde podem ser identificados com facilidade dois *clusters*.

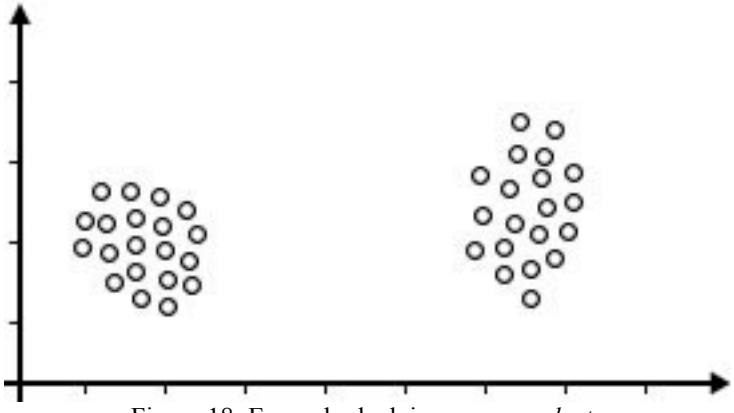

Figura 18: Exemplo de dois pequenos *clusters*
Fonte: Adaptado de COX, E. (2005)

No entanto, não é necessário analisar exemplos de grandes bases de dados a fim de encontrar problemas de clusterização. Estes problemas já aparecem em grupos pequenos, quando dados específicos se diferenciam dos *clusters* criados (HÖPPNER et

[19] Formalizada por George Boole em 1854, tem como principal característica a execução de cálculos exatos por meio de conjuntos *crisp*. Estes conjuntos são restritos aos valores 0 (falso) e 1 (verdadeiro) (COX, 2005).

al, 1999). A Figura 19 apresenta um grupo de dados e dois pontos (P1 e P2) que possuem características exclusivas e que diferem dos padrões adotados pelos *clusters*.

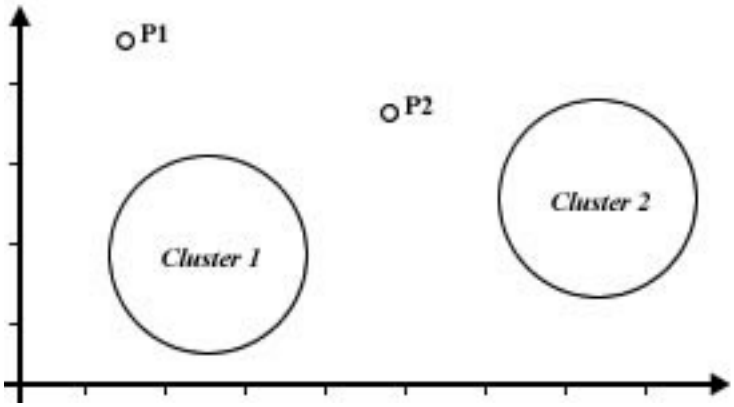

Figura 19: C*lusters* com pontos distintos
Fonte: Adaptado de HÖPPNER, F. et al (1999)

No entanto, encontrar dados os quais apresentam *clusters* que possam ser particionados com facilidade são raros. Grande parte destas informações é disposta em uma área onde os grupos compartilham dados em comum (COX, 2005). A Figura 20 demonstra esse tipo de problema, onde os pontos de dados estão agrupados. Além disso, dois elementos estão distantes deste grupo, o que dificulta ainda mais a identificação correta dos *clusters*.

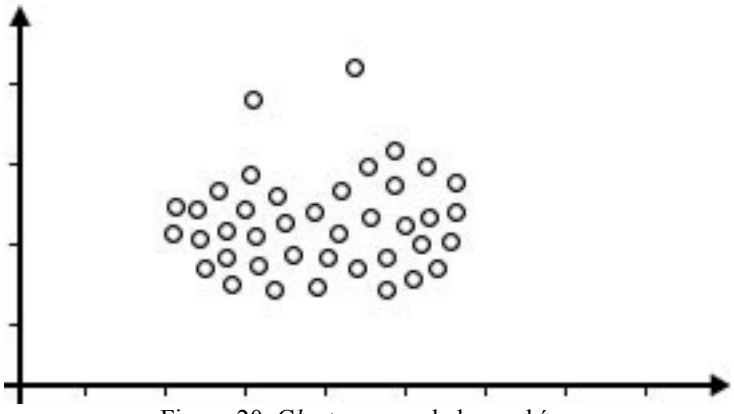

Figura 20: C*lusters* com dados ambíguos
Fonte: Adaptado de COX, E. (2005)

Em algoritmos clássicos que utilizam a lógica booleana, como por exemplo o *K-Means*, o agrupamento correto destes tipos de *cluster* depende dos métodos matemáticos utilizados no pré-processamento. No entanto, nestes algoritmos tradicionais os dados são obrigados a pertencerem a somente um grupo e isto pode

dificultar a identificação de um dado ambíguo. Este pode acabar pertencendo a um *cluster* o qual não possua suas reais características (CHEN, 2001).

Considerando isto, surgiu o método de lógica *fuzzy*, que permite que estes dados pertençam a diversos *clusters* simultaneamente. Esta abordagem assimila graus de pertinência para cada dado em relação aos grupos encontrados, a fim de permitir maior flexibilidade na divisão dos *clusters* (HÖPPNER et al, 1999).

A lógica *fuzzy* possui uma teoria genérica em relação a lógica booleana. Os conjuntos *fuzzy* aceitam valores intermediários entre 0 e 1, permitindo uma abordagem diferenciada em aplicações que possuem dados incertos.

5.1 LÓGICA *FUZZY*

A lógica *fuzzy* é uma teoria matemática que proporciona a modelagem de sistemas para tratamento da incerteza[20] por imprecisão. Este modo aproximado de raciocínio assemelha-se ao conhecimento humano na necessidade de tomada de decisão em situações de ausência de informações (KLIR; YUAN, 1995).

Também conhecida como lógica nebulosa ou difusa, consiste em interpretar variáveis lingüísticas tais como alto, médio e baixo. Cada pessoa pode entender estas expressões de maneira diferente e isto desenvolve ambigüidade nos resultados. A lógica *fuzzy* permite uma abordagem que interprete estas variáveis e retorne saídas de acordo com os objetivos do sistema (CALDEIRAS et al, 2007).

Diferente da lógica booleana (que permite apenas os valores 0 e 1), a lógica difusa utiliza os conjuntos *fuzzy* para executar seus cálculos, os quais podem assumir valores intermediários, ou seja, valores entre 0 e 1. A teoria utiliza a função de grau de

[20] Falta de informações ou de conhecimentos adequados em relação a uma situação (LUGER, 2004).

pertinência para identificar o quanto um elemento pertence a um conjunto (ZADEH, 1965).

Os graus de pertinência determinam a porcentagem de quanto um elemento pertence a um determinado conjunto. O número de conjuntos será a quantidade de graus de pertinência que um elemento possuirá (COX, 2005). Por exemplo, se forem definidos cinco conjuntos, cada dado deverá ter cinco pertinências diferentes.

A fim de calcular o quanto um atributo pertence a um conjunto, são utilizadas as funções de pertinência. A seguir, são demonstrados alguns exemplos, considerando que a variável *x* é o valor a ser verificado (KLIR; YUAN, 1995):

a) **função trapezoidal**: tem como característica a forma de um trapézio. Sua função geral é denominada por:

$$\mu(x)=\begin{cases} 0, \, se \; x < a \\ \frac{x-a}{m-a}, \, se \; x \in [a, m] \\ 1, \, se \; x \in [m, n] \\ \frac{b-x}{b-n}, \, se \; x \in [n, b] \\ 0, \, se \; x > b \end{cases}$$

b) **função triangular**: apresenta a forma de um triangulo. A função geral é definida por:

$$\mu(x)=\begin{cases} 0, \, se \; x \leq a \\ \frac{x-a}{m-a}, \, se \; x \in [a, m] \\ \frac{b-x}{b-m}, \, se \; x \in [m, b] \\ 0, \, se \; x \geq b \end{cases}$$

c) **função S**: sua forma apresenta uma curvatura em forma de S. A função geral é descrita por:

$$\mu(x)=\begin{cases} 0,\ se\ x \le a \\ 2\cdot\left(\dfrac{x-a}{b-a}\right)^2,\ se\ x \in [a,m] \\ 1-2\cdot\left(\dfrac{x-b}{b-a}\right)^2,\ se\ x \in [m,b] \\ 1,\ se\ x > b \end{cases}$$

A Figura 21 representa as formas destas funções em gráficos, considerando que cada uma delas possui suas respectivas variáveis em pontos diferentes de sua forma geométrica.

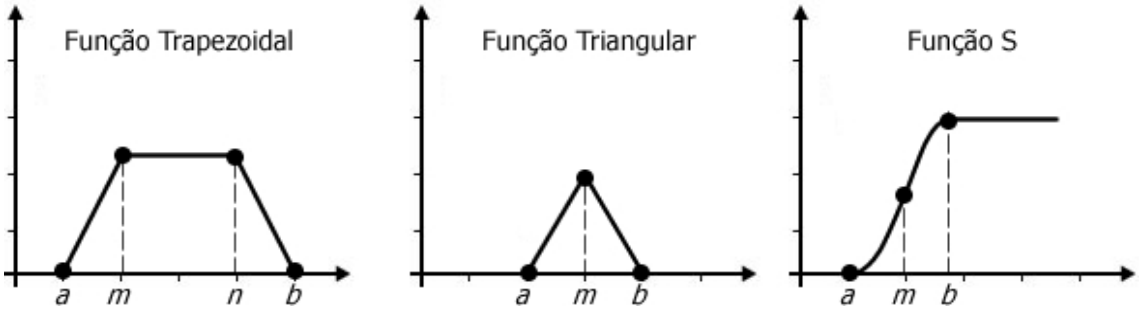

Figura 21: Exemplos de funções de pertinência.
Fonte: Adaptado de Adaptado de KLIR, G.; YUAN, B. (1995)

Um exemplo de conjuntos *fuzzy* por meio da variável temperatura é demonstrado na Figura 22. Quando a temperatura é 20, o grau de pertinência do conjunto *Baixa* é 1 e os outros são 0. Porém, quando a temperatura aumenta para 25, tanto o conjunto *Baixa* como o conjunto *Média* possuem o mesmo grau de pertinência (0.5). Isto determina que a temperatura pertence aos dois conjuntos, considerando que a temperatura seria *Baixa* e *Média*. A seguir são demonstrados as funções de pertinência para cada conjunto:

$$\mu_{baixa}(x)=\begin{cases} 0,\ se\ x < 5 \\ \dfrac{x-5}{10},\ se\ x \in [5,15] \\ 1,\ se\ x \in [15,20] \\ \dfrac{30-x}{10},\ se\ x \in [20,30] \\ 0,\ se\ x > 30 \end{cases}$$

$$\mu_{media}(x)=\begin{cases} 0,\ se\ x \le 20 \\ \frac{x-20}{10},\ se\ x \in [20,30] \\ \frac{40-x}{10},\ se\ x \in [30,40] \\ 0,\ se\ x \ge 40 \end{cases}$$

$$\mu_{alta}(x)=\begin{cases} 0,\ se\ x \le 30 \\ \frac{x-30}{10},\ se\ x \in [30,40] \\ \frac{50-x}{10},\ se\ x \in [40,50] \\ 0,\ se\ x \ge 50 \end{cases}$$

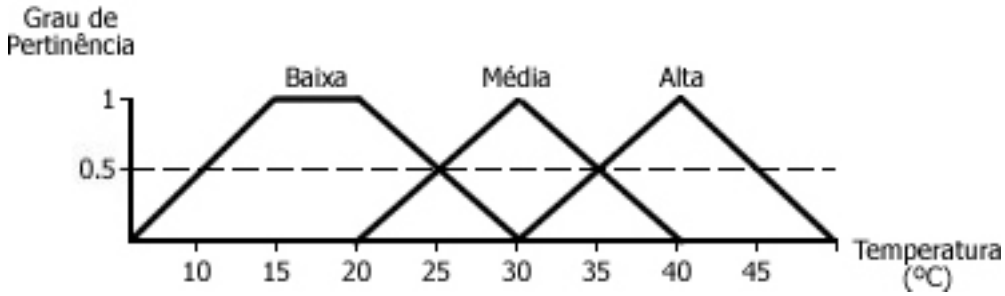

Figura 22: Conjuntos *fuzzy* por meio da variável temperatura
Fonte: Adaptado de COX, E. (2005)

Os graus de pertinência permitem a transição suave entre os conjuntos. Isto possibilita à lógica *fuzzy* cálculos mais precisos em relação ao tratamento de incerteza e conseqüentemente, saídas mais exatas (WANG, 1997).

A fim de realizar este tipo de cálculo, são utilizados os sistemas *fuzzy*. Estes sistemas possuem uma arquitetura funcional genérica (Figura 23) e segue algumas etapas (KLIR; YUAN, 1995):

a) **entrada**: são as variáveis de entrada do sistema contendo valores *crisp*;

b) ***fuzzyficação***: identifica os graus de pertinência relacionados a cada conjunto *fuzzy*;

c) **inferência**: faz a ligação entre os conjuntos difusos de entrada e saída com base em seus respectivos graus de pertinência. Os cálculos são feitos

por meio de regras *fuzzy*, considerando as pertinências dos elementos em relação aos conjuntos definidos;

d) ***defuzzyficação***: a partir das informações da inferência, o sistema identifica os valores *crisp* para cada elemento;

e) **saída**: variáveis de saída contendo valores *crisp*.

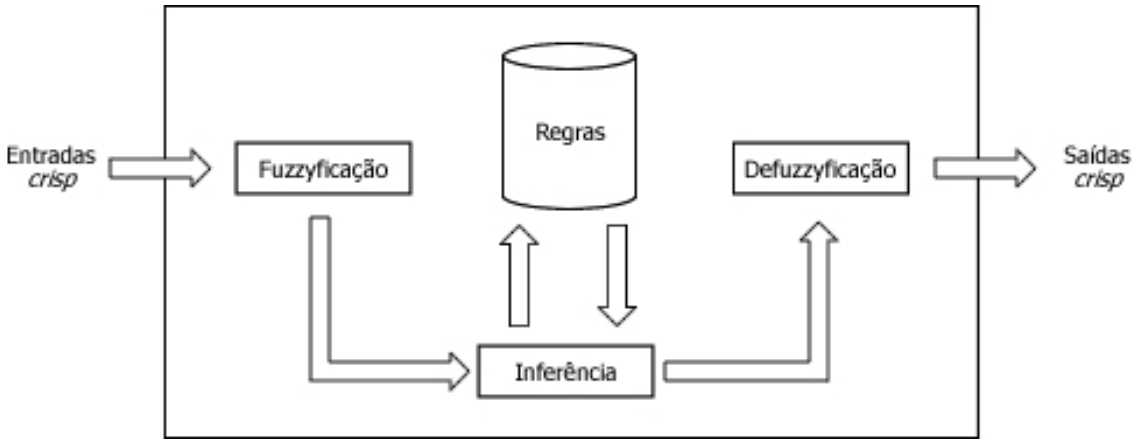

Figura 23. Principais componentes de um sistema *fuzzy*
Fonte: Adaptado de COX, E. (2005)

Dentre suas possíveis aplicações, existe disponível o método de lógica *fuzzy* para a tarefa de clusterização, onde esta auxilia na interpretação dos grupos e divisão correta dos dados entre seus respectivos *clusters*, considerando seus graus de pertinência em relação a cada grupo encontrado pelo algoritmo (BEZDEK et al, 2005).

Existem alguns algoritmos que implementam o modelo *fuzzy* para a tarefa de clusterização e sua escolha depende dos objetivos do domínio da aplicação em relação a descoberta de conhecimento em sua base de dados.

5.2 ALGORITMOS DE LÓGICA *FUZZY* PARA A TAREFA DE CLUSTERIZAÇÃO

Os algoritmos de lógica *fuzzy* para a tarefa de clusterização em *data mining* utilizam a teoria dos conjuntos difusos, a fim de permitir que os elementos distintos de

um grupo ou os dados que estão entre *clusters* sejam classificados com menor taxa de erro (HÖPPNER et al, 1999).

Existem diferentes algoritmos de lógica *fuzzy* para a tarefa de clusterização, onde cada um possui suas próprias características e sua performance depende do tipo de aplicação e do conhecimento do possível número de *clusters*, pois isto é essencial para o algoritmo ter sucesso na identificação correta dos grupos. Considerando isto, deve-se analisar os dados antes de se efetuar o processo de agrupamento (COX, 2005).

No método de lógica *fuzzy* os *clusters* também são chamados de protótipos (*prototypes*), pelo fato de seus elementos não pertencerem totalmente ao grupo, onde a variável que irá definir se um dado pertence a um *cluster* é seu grau de pertinência, considerando que o elemento será assimilado ao grupo que possuir o maior valor de pertinência (BEZDEK et al, 2005).

A seguir serão apresentados alguns destes algoritmos e com o objetivo de facilitar sua descrição, sua aplicação é demonstrada em um gráfico de duas dimensões, permitindo visualizar com maior facilidade as funcionalidades dos métodos.

5.2.1 Algoritmo *Fuzzy C-Means*

O *Fuzzy C-Means* (FCM) é um algoritmo tradicional do método de lógica difusa. Ele foi derivado do algoritmo *Hard C-Means* (ou *K-Means*). A versão final do FCM foi proposta por James C. Bezdek em 1973 (BEZDEK et al, 2005).

Em sua implementação foi criado um parâmetro *fuzzyficador* que determina o grau de *fuzzyficação* entre os *clusters* e quanto maior for este valor, mais os elementos são relacionados entre os grupos. Este valor deve ser balanceado, ou seja, não deve

realizar uma relação *crisp*, como também não deve realizar uma clusterização onde todos os elementos pertencem a todos os grupos (CHEN, 2001).

A fim de proporcionar protótipos de alta qualidade, foram realizados testes a fim de determinar um valor padrão para este parâmetro. Considerando isto, recomenda-se o uso do valor 2, pois este possibilita uma clusterização satisfatória (COX, 2005).

Como o FCM é baseado no *K-Means*, sua tendência é criar protótipos de forma circular. Em determinados casos pode ser vantajoso, porém isto limita a aplicação do algoritmo em casos onde os grupos podem variar de tamanho (HÖPPNER et al, 1999).

O FCM pode ser utilizado tanto para clusterização de dados como de imagens (BEZDEK et al, 2005). A Figura 24 demonstra um exemplo da clusterização pelo FCM, onde são encontrados três grupos circulares de forma satisfatória, onde cada símbolo identifica um *cluster* diferente.

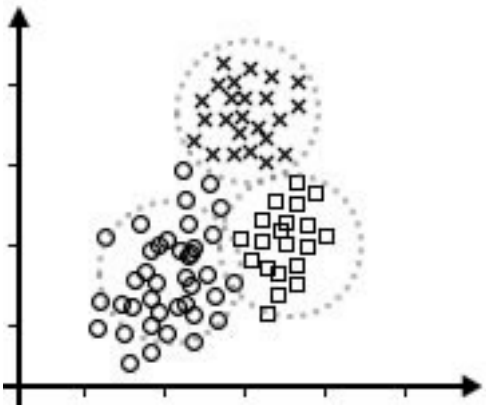

Figura 24. Clusterização *Fuzzy C-Means*
Fonte: Adaptado de HÖPPNER, F. et al (1999)

5.2.2 Algoritmo *Fuzzy C-Varieties*

Desenvolvido por James C. Bezdek e Hans H. Bock em 1981, o *Fuzzy C-Varieties* (FCV) foi criado com o propósito de reconhecer *clusters* em formas de linhas em uma superfície plana (HÖPPNER et al, 1999).

O FCV utiliza pouco processamento, pois para identificar corretamente as linhas não é necessário definir muitos *clusters*. Em testes realizados pelos autores, quanto menor o número de centros dos protótipos, maior será eficiência do algoritmo (BEZDEK et al, 2005).

No entanto, o algoritmo se mostrou ineficiente em casos onde as linhas se cruzavam, identificando erroneamente todos os *clusters*. Outro problema é o fato do algoritmo sempre encontrar protótipos do mesmo tamanho, pelo fato da variável de dimensão dos *clusters* serem as mesmas. Nestes casos, recomenda-se a utilização de outro método (HÖPPNER et al, 1999).

Sua principal aplicação é relacionada com clusterização de imagens que contenham linhas (HÖPPNER et al, 1999). A Figura 25 mostra um reconhecimento de linhas pelo FCV, considerando que cada conjunto de símbolo no gráfico representa um *cluster* encontrado pelo algoritmo.

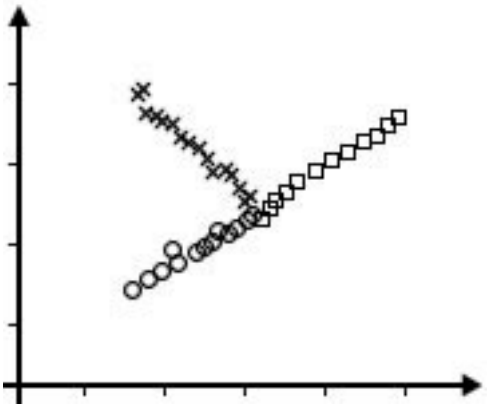

Figura 25. Clusterização por meio do algoritmo *Fuzzy C-Varieties*
Fonte: Adaptado de HÖPPNER, F. et al (1999)

5.2.3 Algoritmo *Fuzzy C-Elliptotypes*

Ao considerar o problema do algoritmo FCV de encontrar *clusters* de tamanhos parecidos, um modelo adaptável foi proposto por Rajesh N. Davé em 1989, o *Fuzzy C-Elliptotypes* (FCE).

Neste algoritmo, cada *cluster* possui sua própria variável de dimensão, que define o tamanho de cada grupo individualmente. Isto possibilita a identificação de protótipos de vários tamanhos, como também uma boa adaptação em relação a diferentes tipos de dados (HÖPPNER et al, 1999).

O modelo identifica com precisão linhas de diversos tamanhos, como também linhas cruzadas. No entanto, o algoritmo perde eficiência ao processar figuras geométricas como quadrados ou retângulos, identificando protótipos inexistentes. Sua principal aplicação é a identificação de linhas em imagens. (BEZDEK et al, 2005).

A Figura 26 apresenta uma aplicação do FCE, onde são identificadas três linhas cruzadas e cada grupo encontrado pelo algoritmo é representado por um símbolo específico.

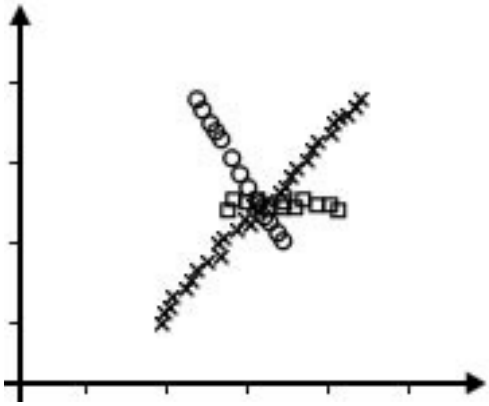

Figura 26. Método de lógica *fuzzy* pelo algoritmo *Fuzzy C-Elliptotypes*
Fonte: Adaptado de BEZDEK, J. et al (2005)

5.2.4 Algoritmo *Fuzzy C-Shells*

Apresentado por Rajesh N. Davé em 1990, o algoritmo *Fuzzy C-Shells* (FCS) tem como principal objetivo identificar protótipos em forma de círculo. Ele possui para cada *cluster* uma variável que contém o raio do círculo e outra variável que contém o ponto central do protótipo (HÖPPNER et al, 1999).

Os círculos reconhecidos pelo algoritmo são de alta qualidade na maioria dos casos onde os dados representam este tipo de abordagem. No entanto, o FCS exige muito processamento e em algumas situações sua aplicação é inviável (BEZDEK et al, 2005).

A Figura 27 demonstra a clusterização por meio do algoritmo FCS, onde três círculos foram encontrados, mesmo quando estes estão unidos. Cada grupo é representado por um símbolo diferente.

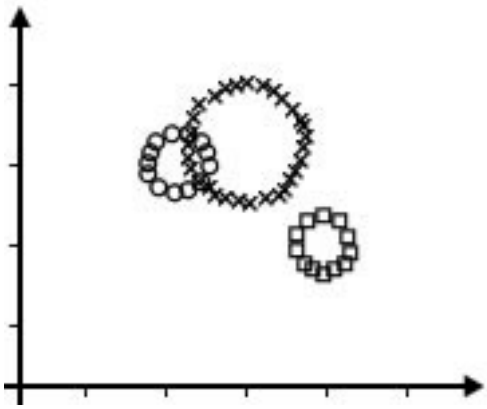

Figura 27. Clusterização por meio do algoritmo *Fuzzy C-Shells*
Fonte: Adaptado de HÖPPNER, F. et al (1999)

Segundo Höppner et al (1999) como sua aplicação é exclusiva para identificação de círculos em imagens, vários outros algoritmos mais específicos foram desenvolvidos com base em suas funções de reconhecimento, como exemplo a identificação de: esferas (*Fuzzy C-Spherical Shells*), elipses[21] (*Fuzzy C-Ellipses*),

[21] Similar a um círculo, no entanto se difere por possuir dois lados mais alongados (STEWART, 2006).

elipsóides[22] (*Fuzzy C-Ellipsoidal Shells*) e protótipos contendo parábolas[23] e linhas (*Fuzzy C-Quadric Shells*).

Além disso, também foi desenvolvido um método adaptativo para o algoritmo FCS, o *Adaptive Fuzzy C-Shells*, sendo capaz de reconhecer tanto círculos como elipses (BEZDEK et al, 2005).

5.2.5 Algoritmo Gustafson-Kessel

Desenvolvido em 1979 por Donald E. Gustafson e William C. Kessel, o algoritmo Gustafson-Kessel (GK) é uma extensão do algoritmo FCM. No entanto, o GK utiliza uma matriz de covariância[24] *fuzzy* para calcular as relações entre os atributos da base de dados e utiliza-la na atualização da distância (GUSTAFSON; KESSEL, 1979).

O diferencial do GK em relação ao FCM é que esta matriz permite ao algoritmo adaptar-se a diferentes formas de *clusters*, contrariando os círculos criados pelo FCM. Isto possibilita a formação de protótipos de formas e tamanhos variados. No entanto, considerando os algoritmos de identificação de figuras geométricas em imagens, o algoritmo GK não consegue o mesmo desempenho, pois sua aplicação é mais genérica (BEZDEK et al, 2005).

Os *clusters* possuem suas próprias matrizes de covariância *fuzzy*, permitindo a cada grupo possuir características de forma e tamanho independentes. Considerando isto, o GK necessita de maior processamento em relação ao FCM. Porém, esta necessidade não altera sua aplicação, visto que sua capacidade de adaptação a diferentes dados compensa esta característica (HÖPPNER et al, 1999). A Figura 28 demonstra

---

[22] Uma elipse tridimensional (STEWART, 2006).
[23] Metade de uma elipse (STEWART, 2006).
[24] Matriz contendo as medidas de afastamento (distância) entre um grupo de elementos (GUSTAFSON; KESSEL, 1979).

uma aplicação com o algoritmo GK, onde existem três grupos de tamanhos variados e sua clusterização é feita com sucesso e a identificação de cada *cluster* é feito por um símbolo específico.

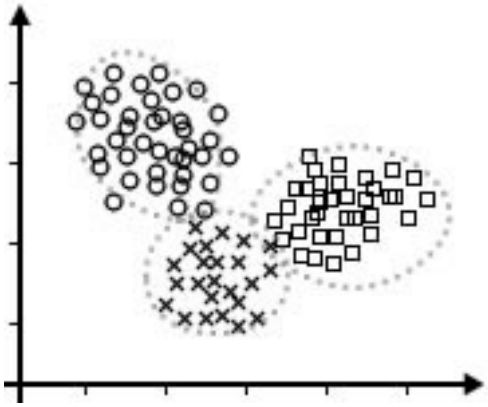

Figura 28. Clusterização Gustafson-Kessel
Fonte: Adaptado de BEZDEK, J. et al (2005)

Considerando que o objetivo geral desta pesquisa é a implementação do algoritmo Gustafson-Kessel, suas características, bem como funcionalidades específicas, serão apresentadas com detalhes a fim de possibilitar seu entendimento.

6 O ALGORITMO GUSTAFSON-KESSEL

Em 1979, na *IEEE Conference on Decision and Control* na cidade de San Diego, Califórina, Donald E. Gustafson e William C. Kessel publicaram o artigo *Fuzzy Clustering with Fuzzy Covariance Matrix*. Neste, eles descrevem uma modificação do tradicional algoritmo FCM. A modificação descrita no artigo foi intitulada como Gustafson-Kessel (GK), devido ao nome de seus autores.

A principal alteração em relação ao FCM foi a troca da distância euclidiana por outra que encontra com maior precisão os grupos existentes. Considerando isto, os autores adotaram a distância de Mahalanobis, a qual implementa uma matriz de covariância entre os atributos disponíveis na base de dados. Esta matriz possui a função de calcular a relação entre as diferentes propriedades, a fim de possibilitar maior flexibilidade ao determinar os grupos encontrados (BEZDEK et al, 2005).

A matriz de covariância permite ao algoritmo Gustafson-Kessel encontrar *clusters* de formas geométricas independentes, ou seja, cada grupo possui suas próprias características de dimensões. Por isso, os resultados gerados pelo GK são, em geral, superiores em relação aos algoritmos tradicionais e ao FCM (GUSTAFSON; KESSEL, 1979).

Este algoritmo implementa o parâmetro *fuzzyficador* (*m*), o qual determina a *fuzzyficação* entre os elementos e seus protótipos. Se o valor de *m* for 1, não existirá esta relação de *fuzzyficação* entre dados e grupos, o que gera uma clusterização tradicional, onde cada elemento pertence exclusivamente a um *cluster*. A medida em que este valor é incrementado, a *fuzzyficação* entre os protótipos aumenta (COX, 2005).

Normalmente, é utilizado o valor 2 para este parâmetro, pois desenvolve uma *fuzzyficação* satisfatória entre elementos e *clusters*, onde os registros pertencem um

pouco a cada grupo. No entanto, as características específicas de cada dado determinarão a qual *cluster* este irá pertencer (HÖPPNER et al, 1999).

O algoritmo Gustafson-Kessel permite apenas a utilização de valores numéricos, pelo fato de todo o processo de clusterização realizar somente operações matemáticas. Se for necessário a utilização de atributos nominais, estes devem ser convertidos em valores decimais (BEZDEK et al, 2005).

Antes de começar os cálculos, deve-se escolher o número de *clusters* que se deseja identificar e determinar o parâmetro de *fuzzyficação m*. A fim de finalizar a clusterização, deve-se definir o grau de erro (ε), que será comparado com o erro calculado ao final de cada iteração do algoritmo.

O erro calculado pelo algoritmo consiste na variação entre as duas últimas pertinências entre todos os elementos e grupos. Este valor é comparado com o número definido pelo grau do erro e o algoritmo encerra sua execução se todas as pertinências forem menor que o valor do grau do erro (GUSTAFSON; KESSEL, 1979).

A matriz de pertinências *U*, que contém as pertinências dos elementos em relação aos grupos, deve ser criada antes de se iniciar os cálculos. Como os primeiros valores desta matriz não interferem no resultado final, os graus iniciais podem ser randômicos, desde que respeitem a propriedade de soma das pertinências dada por (GUSTAFSON; KESSEL, 1979):

$$\sum_{i=1}^{c} u_{ij} = 1 \tag{1}$$

Onde:

a) $\boldsymbol{c}$: número total de *clusters*;

b) $\boldsymbol{u_{ij}}$: grau de pertinência entre o i-ésimo *cluster* e o j-ésimo elemento.

Esta propriedade (1) determina que a soma dos graus de pertinência de cada elemento *j* em relação aos *c clusters* deva ser igual a 1, sendo adotada sempre que as pertinências forem recalculadas.

Posteriormente, dado um vetor de dados[25] de tamanho *p* $[x_1, x_2, x_3, ..., x_p]$, onde cada elemento possua um vetor transposto[26] de *N* dimensões[27] $[x_{p1}, x_{p2}, x_{p3}, ..., x_{pN}]^T$, deve-se minimizar a seguinte função objetivo (GUSTAFSON; KESSEL, 1979):

$$E = \sum_{i=1}^{c} \sum_{j=1}^{p} (u_{ij})^m d^2(x_j, c_i) \quad (2)$$

Onde:

a) ***E***: valor a ser minimizado, o qual determina os centros dos *clusters* e os graus de pertinências dos dados;

b) ***c***: número total de *clusters*;

c) ***p***: número total de elementos;

d) u_{ij}: grau de pertinência entre o i-ésimo *cluster* e o j-ésimo elemento;

e) ***m***: parâmetro de *fuzzyficação*;

f) d^2: distância entre o j-ésimo elemento *x* e o i-ésimo *cluster c*.

Utilizando-se desta equação (2), é possível executar a clusterização pelo método de lógica *fuzzy*, onde cada algoritmo possui suas próprias funções de distância entre elementos e *clusters*. No caso do algoritmo Gustafson-Kessel, cinco fases principais estão envolvidos no processo, os quais devem ser executados na seguinte ordem (GUSTAFSON; KESSEL, 1979):

a) calcular os centros dos *clusters*;

b) computar as matrizes de covariância *fuzzy*;

---

[25] Conjunto de dados agrupados seqüencialmente. Pode ser representado pela posição em linha ou transposta (POOLE; MONTEIRO, 2004).

[26] Inverte a posição do vetor de dados, como por exemplo, se o vetor for horizontal, este será transformado em um vetor vertical (GUSTAFSON; KESSEL, 1979).

[27] Cada atributo disponível na base de dados refere-se a uma dimensão, por exemplo, se a base de dados possuir 5 atributos, significa que apresenta 5 dimensões (BEZDEK et al, 2005).

c) atualizar as distâncias entre os elementos e seus grupos;

d) determinar os novos graus de pertinência;

e) definir a condição de parada do algoritmo.

A fim de compreender melhor o funcionamento do algoritmo Gustafson-Kessel, cada uma das etapas citadas são conceituadas individualmente.

6.1 CÁLCULO DOS CENTROS DOS *CLUSTERS*

O centro de um *cluster* determina a referência para os cálculos de distância em relação aos elementos disponíveis na base de dados. Dependendo do algoritmo, o centro de um grupo pode ser um elemento da base de dados ou um ponto distinto calculado pelo método (BEZDEK et al, 2005).

No método de lógica *fuzzy* é calculado um ponto que determinará o centro de cada *cluster*. Este valor não é um dado existente na base de dados, e sim, um número fictício calculado pelo algoritmo a fim de representar o centro do *cluster*. A equação (3) é utilizada a fim de calcular o centro de cada grupo existente na clusterização (GUSTAFSON; KESSEL, 1979):

$$c_i = \frac{\sum_{j=1}^{p} (u_{ij})^m x_j}{\sum_{j=1}^{p} (u_{ij})^m} \tag{3}$$

Onde:

a) $\boldsymbol{c_i}$: centro do i-ésimo *cluster*;

b) $\boldsymbol{p}$: número total de elementos;

c) $\boldsymbol{u_{ij}}$: grau de pertinência entre o i-ésimo *cluster* e o j-ésimo elemento;

d) $\boldsymbol{m}$: parâmetro de *fuzzyficação*;

e) $\boldsymbol{x_j}$: j-ésimo elemento.

Os centros são calculados individualmente para cada *cluster*. A equação (3) realiza o somatório dos valores resultantes da multiplicação de todos os graus de pertinência e os vetores de dados dos elementos, considerando que os graus são elevados ao parâmetro *fuzzyficador*. Após isto, o resultado é dividido pelo somatório de todos os graus de pertinências elevados ao parâmetro de *fuzzyficação*.

Cada centro encontrado possuirá o mesmo número de dimensões dos elementos, ou seja, se a base de dados possuir *N* dimensões, o resultado de cada centro calculado será um vetor transposto de *N* dimensões $[c_{i1}, c_{i2}, c_{i3}, ..., c_{iN}]^T$. Finalizado o processo de atualização de todos os centros, deve-se encontrar as matrizes de covariância *fuzzy* de cada *cluster*.

6.2 CÁLCULO DAS MATRIZES DE COVARIÂNCIA *FUZZY*

As matrizes de covariância determinam a relação entre os atributos disponíveis na base de dados e os valores encontrados determinaram a dimensão e a forma de cada *cluster*, pois cada grupo possuirá sua própria matriz de covariância. Esta propriedade torna o algoritmo Gustafson-Kessel bastante flexível, pois as matrizes de covariância permitem aos grupos encontrados se adaptarem aos dados disponíveis (HÖPPNER et al, 1999).

Ao atualizar os centros de todos *clusters*, as matrizes de covariância *fuzzy* podem ser calculadas de acordo com a equação (4) (GUSTAFSON; KESSEL, 1979):

$$F_i = \frac{\sum_{j=1}^{p} (u_{ij})^m (x_j - c_i)(x_j - c_i)^T}{\sum_{j=1}^{p} (u_{ij})^m} \quad (4)$$

Onde:

a) $\boldsymbol{F_i}$: matriz de covariância *fuzzy* do i-ésimo *cluster*;

b) $\boldsymbol{p}$: número total de elementos;

c) $\boldsymbol{u_{ij}}$: grau de pertinência entre o i-ésimo *cluster* e o j-ésimo elemento;

d) $\boldsymbol{m}$: parâmetro de *fuzzyficação*;

e) $\boldsymbol{x_j}$: j-ésimo elemento;

f) $\boldsymbol{c_i}$: centro do i-ésimo *cluster*.

A matriz de covariância de cada grupo é calculada pela multiplicação das pertinências e a subtração do vetor de dados e o centro do *cluster*, considerando que os graus são elevados aos parâmetro *fuzzyficador*. O valor obtido é multiplicado pela mesma subtração entre o vetor de dados e o centro, no entanto na forma transposta. Ao final, é feito um somatório destes valores.

A parte superior da equação (4) gera uma matriz *N*x*N*, onde *N* é a dimensão da base de dados. Esta matriz é divida pela soma das pertinências elevadas ao parâmetro de *fuzzyficação*, tendo-se como resultado obtido a matriz de covariância *fuzzy* do i-ésimo *cluster*.

Ao atualizar todas as matrizes, pode-se computar as novas distâncias de cada elemento da base de dados em relação aos centros dos *clusters* encontrados.

6.3 ATUALIZAÇÃO DAS DISTÂNCIAS ENTRE ELEMENTOS E *CLUSTERS*

O cálculo da distância define uma medida entre cada ponto e os grupos encontrados. Este valor é de total importância para o algoritmo, pois os números computados serão utilizados a fim de atualizar as pertinências dos elementos disponíveis na base de dados (BEZDEK et al, 2005).

A distância utilizada pelo algoritmo Gustafson-Kessel é a de Mahalanobis, a qual executa seus cálculos por meio de matrizes de covariância. A equação (5) define a atualização das distâncias (GUSTAFSON; KESSEL, 1979):

$$d^2(x_j, c_i) = (x_j - c_i)^{\mathrm{T}} M_i (x_j - c_i) \tag{5}$$

Onde:

a) $\boldsymbol{d^2}$: distância entre o j-ésimo elemento *x* e o i-ésimo *cluster c*;

b) $\boldsymbol{x_j}$: j-ésimo elemento;

c) $\boldsymbol{c_i}$: centro do i-ésimo *cluster*;

d) $\boldsymbol{M_i}$: matriz de covariância modificada do i-ésimo *cluster*.

A distância é calculada pela multiplicação da subtração entre o vetor de dados e o centro pela matriz de covariância modificada. Após isto, o resultado é multiplicado pela subtração entre o vetor de dados e o centro, no entanto na forma transposta.

A matriz de covariância *Mi* deve ser ajustada a dimensão e a forma do *cluster*, definida pela equação (6) (GUSTAFSON; KESSEL, 1979):

$$M_i = \sqrt[N]{\det(F_i)}\, F_i^{-1} \tag{6}$$

Onde:

a) $\boldsymbol{N}$: número de dimensões da base de dados;

b) $\boldsymbol{F_i}$: matriz de covariância *fuzzy* do i-ésimo *cluster*.

A multiplicação entre a raiz (de ordem *N*) do determinante da matriz de covariância *fuzzy* e a matriz de covariância *fuzzy* inversa resultam na matriz de covariância modificada. Seu resultado será utilizado no cálculo da distância.

Finalizado os cálculos, pode ocorrer de alguma distância entre um dado e um grupo for igual a zero. Neste caso, a pertinência deste elemento em relação a este

cluster deve ser igual a 1 e todos os outros graus de pertinência devem ser igual a zero (HÖPPNER et al, 1999).

Os valores obtidos definirão uma distância em relação a todos os grupos existentes para cada elemento. Considerando estes valores, pode-se atualizar a matriz de pertinência *U*, a qual definirá quanto um elemento pertence a cada *cluster*.

6.4 ATUALIZAÇÃO DOS GRAUS DE PERTINÊNCIA

A identificação do quanto cada elemento pertence aos *clusters* encontrados são os graus de pertinência e devem respeitar a propriedade (1), ou seja, a soma de todas as pertinências de um elemento deve ser igual a 1 (BEZDEK et al, 2005).

No entanto, não existe necessidade de aplicar esta propriedade neste cálculo, visto que os procedimentos matemáticos utilizados pela equação (7) a fim de atualizar as pertinências geram a soma correta dos valores. A atualização das pertinências é dada por (GUSTAFSON; KESSEL, 1979):

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}^2}{d_{kj}^2} \right)^{\frac{1}{m-1}}} \tag{7}$$

Onde:

a) u_{ij}: grau de pertinência entre o i-ésimo *cluster* e o j-ésimo elemento;

b) c: número total de *clusters*;

c) d^2_{ij}: distância entre o j-ésimo elemento e o i-ésimo *cluster*;

d) d^2_{kj}: distância entre o j-ésimo elemento e o k-ésimo *cluster*;

e) m: parâmetro de *fuzzyficação*.

O cálculo executa um somatório da distância do atual *cluster* dividido por todas as outras distâncias. Cada valor encontrado deve ser elevado a 1 dividido pelo

grau de *fuzzyficação* menos 1. Após isto, o valor 1 é dividido pela soma obtida resultando na pertinência deste elemento em relação ao grupo.

Ao final da atualização de todas as pertinências dos elementos, deve-se calcular uma condição de parada, a fim de terminar a execução do algoritmo.

6.5 CÁLCULO DA CONDIÇÃO DE PARADA

A condição de parada deve finalizar o processamento do algoritmo. O método padrão adotado por Gustafson-Kessel é o cálculo do erro entre pertinências. No entanto, outro método pode ser adotado, o do número de iterações, o qual o usuário determina a quantidade de iterações do algoritmo (BEZDEK et al, 2005).

Porém, o cálculo de erro entre os graus de pertinência é satisfatório, visto que o algoritmo após várias iterações apresenta pouca modificação dos valores de pertinência, o qual registra o fim dos cálculos de clusterização. A utilização deste cálculo é recomendado, pois na maioria dos casos reduz a quantidade de ciclos que o algoritmo necessitaria a fim de finalizar a clusterização. A equação (8) descreve o cálculo do erro e a condição de parada (GUSTAFSON; KESSEL, 1979):

$$\|U^{l} - U^{l-1}\| \leq \varepsilon \tag{8}$$

Onde:

a) $\boldsymbol{U}$: matriz de pertinências;

b) $\boldsymbol{l}$: número de iterações atual;

c) $\boldsymbol{\varepsilon}$: valor do erro;

Considerando que todos os valores da subtração entre os graus de pertinência atuais e anteriores forem menor ou igual ao erro, a condição torna-se verdadeira. Assim, o algoritmo finaliza sua execução e a última matriz de pertinências

será o resultado final. No entanto, o algoritmo permanecerá em processamento enquanto a equação (8) for falsa.

Finalizada a compreensão das funcionalidades do Gustafson-Kessel, algumas aplicações podem exemplificar as diversas utilizações da clusterização pelo método de lógica *fuzzy* por meio deste algoritmo.

7 ALGUNS EXEMPLOS DE USO DO ALGORITMO GUSTAFSON-KESSEL PARA CLUSTERIZAÇÃO

O método de lógica *fuzzy* para clusterização não é tão comum quanto os algoritmos tradicionais, os quais são de implementação consideravelmente mais simples e de menor tempo de execução, se comparado aos que são baseados em lógica difusa.

No entanto, os algoritmos de lógica *fuzzy* possuem maiores estudos por parte dos especialistas da área, onde estes implementam novas soluções ou melhoramentos nestes tipos de algoritmo. A seguir serão abordadas algumas aplicações por meio do algoritmo de lógica *fuzzy* Gustafson-Kessel.

7.1 PREVISÃO NÃO-LINEAR DOS PREÇOS DE TRONCOS DE EUCALIPTO BASEADA EM UMA ABORDAGEM NEUROEVOLUTIVA

Este artigo foi publicado na revista de Gestão e Produção em 2007 na cidade de São Carlos, São Paulo. Foi desenvolvido por Leandro dos S. Coelho, Wesley V. da Silva e Roberto M. Protil, sendo que o primeiro faz parte do Grupo Produtrônica, Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica do Paraná (PUCPR) e os outros autores são do Programa de Pós-Graduação em Administração da PUCPR.

A aplicação desenvolvida consiste em uma rede neural para previsões de séries temporais. Neste processo, o algoritmo Gustafson-Kessel foi utilizado para agrupar os cálculos da rede neural, a fim de otimizá-los e possibilitar resultados mais precisos (COELHO; SILVA; PROTIL, 2007).

O projeto foi proposto para previsão dos preços de troncos de eucalipto para celulose e serraria. A Figura 29 demonstra um exemplo de cálculo feito pelo sistema, estimando de forma satisfatória os preços dos troncos de eucalipto (COELHO; SILVA; PROTIL, 2007).

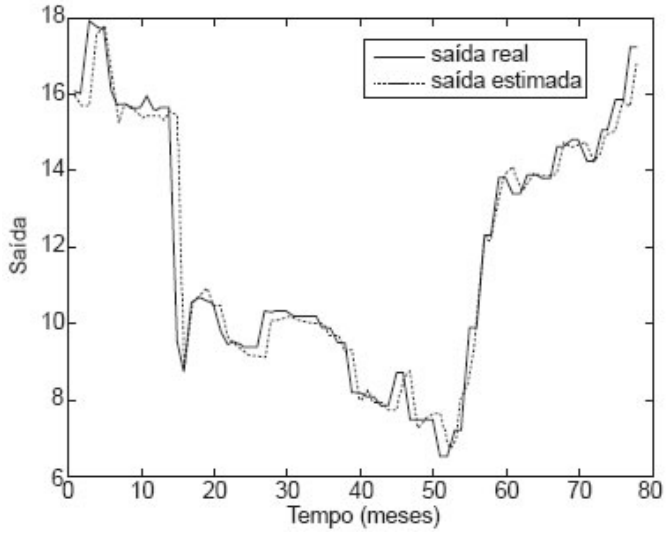

Figura 29. Previsão dos preços dos troncos de eucalipto gerado pela aplicação
Fonte: COELHO, L.; SILVA, W.; PROTIL, R. (2007)

Nos testes realizados, o tempo de processamento do algoritmo GK foi menor em relação ao seu uso em paralelo com outras soluções. No entanto, somente o algoritmo GK não foi suficiente para ajustar com precisão os resultados gerados, sendo necessário sua combinação com algoritmos evolutivos[28], a fim de possibilitar melhores resultados (COELHO; SILVA; PROTIL, 2007).

7.2 DETECTANDO *CLUSTERS* DE DIFERENTES FORMAS GEOMÉTRICAS EM EXPRESSÕES GENÉTICAS

Este projeto foi desenvolvido por Dae-Won Kim, Kwang H. Lee e Doheon Lee e foi publicado em 2005 na revista *Bioinformatics* da Universidade de Oxford, Inglaterra e consiste em um agrupamento de expressões de dados genéticos, a fim de se

[28] Algoritmos baseados no processo de evolução biológica, similares aos algoritmos genéticos (BÄCK; FOGEL; MICHALEWICZ, 1997).

possibilitar aos especialistas da área maior facilidade de análise dos diferentes tipos genéticos.

Vários algoritmos foram testados a fim de se verificar quais poderiam dividir corretamente os grupos de genes em uma base de dados. Entre os utilizados estão os algoritmos *K-Means*, *Fuzzy C-Means* e Gustafson-Kessel (KIM; LEE; LEE, 2005).

Em seus testes, o algoritmo GK mostrou-se muito superior em relação aos outros métodos, dividindo com extrema perfeição os grupos de genes da base de dados (Figura 30). No entanto, em testes em grandes bases seu tempo de processamento foi alto em relação aos outros métodos, devido ao cálculo da matriz de covariância para cada elemento (KIM; LEE; LEE, 2005).

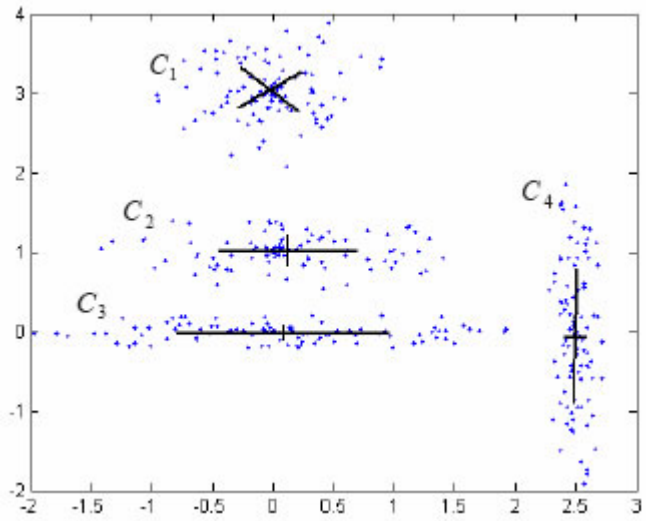

Figura 30. Identificação de quatro grupos por meio do GK
Fonte: KIM, D.; LEE, K.; LEE, D. (2005)

7.3 ANÁLISE DE ALGORITMOS DE CLUSTERIZAÇÃO *FUZZY* PARA SUGESTÃO DE SUPERVISORES DE TESES

Este projeto foi desenvolvido em 2005 como Dissertação de Mestrado do Curso de Ciência da Computação e Sistemas de Informação da Universidade de Tecnologia da Malásia pela estudante Azrina B. Suhaimi.

Em seu trabalho, Suhaimi (2005) desenvolveu um protótipo de clusterização para auxiliar na escolha de supervisores e examinadores que irão fazer a análise das teses dos estudantes da universidade. A aplicação divide os supervisores em grupos conforme seus conhecimentos e aplica os diferentes tipos de teses disponíveis a estes examinadores a fim de escolher os melhores para cada área de aplicação.

Nos testes com os algoritmos de lógica *fuzzy* para clusterização *Fuzzy C-Means* e Gustafson-Kessel, foi constatado que os algoritmos tiveram resultados satisfatórios por meio de diferentes parâmetros de *fuzzyficação*. A Figura 31 demonstra que o algoritmo FCM teve uma performance melhor que o GK, pelo fato dos testes serem realizados em pequenas bases (SUHAIMI, 2005).

Index(*q*)	2.0	2.1	2.2	2.3	2.4	2.5
FCM	31.4%	29.0%	28.2%	24.8%	22.7%	22.6%
GK	21.2%	21.7%	24.6%	16.0%	21.0%	13.8%

Figura 31. Comparação de resultados entre FCM e GK
Fonte: SUHAIMI, A. (2005)

No entanto, Suhaimi (2005) considerou que os dois algoritmos tiveram alto tempo de processamento por processar dados complexos e seriam necessários outros testes a fim de testar suas reais capacidades de clusterização, além de testar outros métodos a fim de escolher o de maior eficiência.

7.4 MÉTODOS DE CLUSTERIZAÇÃO *FUZZY* PARA IDENTIFICAR E MODELAR ESTRATÉGIAS DE CONTROLE NÃO-LINEARES

Este método foi publicado em 2001 no *Journal of Systems and Control Engineering* na cidade de Londres, Inglaterra e foi desenvolvido por Ioannis S. Akkizidis e Geoff N. Roberts. O projeto consiste na identificação e modelagem de ações de controle de sistemas não-lineares[29], os quais consideram a descrição entre entradas e saídas para seu funcionamento. A modelagem destes sistemas se torna complicada quando está envolvido um conhecimento muito específico.

Neste caso, a lógica *fuzzy* está presente a fim de facilitar a compreensão e modelagem destes tipos de sistema. A aplicação dos autores tem como propósito gerar as regras do sistema difuso de acordo com as entradas e saídas (Figura 32). O algoritmo Gustafson-Kessel foi utilizado para agrupar conjuntos semelhantes e gerar regras do sistema (AKKIZIDIS; ROBERTS, 2001).

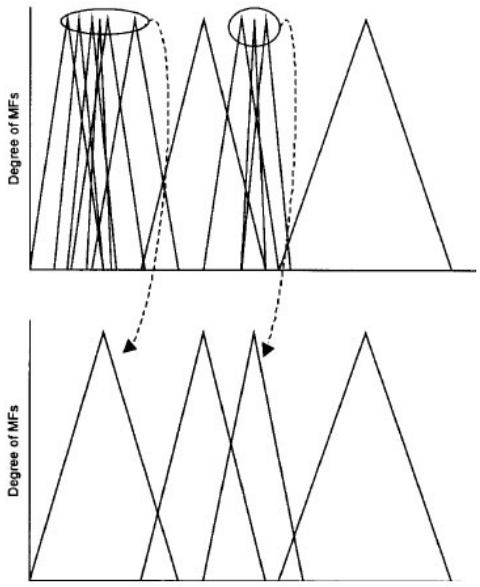

Figura 32. Regras de entrada agrupadas por meio do algoritmo GK
Fonte: AKKIZIDIS, I.; ROBERTS, G. (2001)

[29] Sistemas que não adotam um padrão específico, onde suas características são alteradas ao longo do tempo (POOLE; MONTEIRO, 2004).

Considerando que o projeto é específico a sistemas não-lineares, o algoritmo GK se mostrou o mais eficiente na detecção das similaridades entre os conjuntos do sistema. No entanto, os autores consideram o problema da definição do número de *clusters*, onde este método não possui a característica de definir este parâmetro automaticamente (AKKIZIDIS; ROBERTS, 2001).

7.5 RECONHECIMENTO DE BATIMENTOS CARDÍACOS UTILIZANDO UMA REDE HÍBRIDA NEURO-*FUZZY*

Este artigo demonstra uma aplicação de redes híbridas neuro-*fuzzy* a fim de reconhecer gráficos de batimentos cardíacos e agrupá-los de acordo com o tipo. Foi desenvolvido por Stanislaw Osowski e Tran H. Linh e foi publicado em 2001 no jornal *IEEE Transactions on Biomedical Engineering*, na cidade de Urbana nos Estados Unidos.

O reconhecimento de batimentos cardíacos em Unidades de Terapia Intensiva (UTI) é extremamente importante para o tratamento do paciente e deve ser feito em tempo real. O projeto identifica os gráficos das ondas do batimento cardíaco do paciente (Figura 33) e as classifica por meio de uma rede híbrida neuro-*fuzzy* por meio do algoritmo Gustafson-Kessel (OSOWSKI; LINH, 2001).

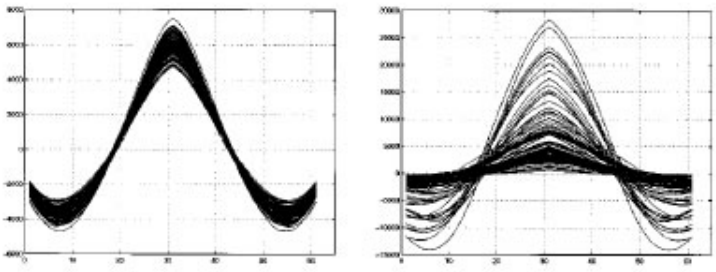

Figura 33. Exemplos de gráficos de ondas de batimento cardíaco
Fonte: OSOWSKI, S.; LINH, T. (2001)

A nova solução proposta pelo GK em conjunto com a rede neural se mostrou eficiente no reconhecimento de batimentos normais e de tipos diferentes. As principais características do método foram a boa taxa de reconhecimento e a alta performance (OSOWSKI; LINH, 2001).

Após as diferentes aplicações do algoritmo Gustafson-Kessel, será descrita a implementação deste método de lógica *fuzzy* no módulo de clusterização da *Shell* Orion *Data Mining Engine*, considerando todos os passos envolvidos no processo de desenvolvimento do algoritmo.

8 O ALGORITMO DE LÓGICA *FUZZY* GUSTAFSON-KESSEL NA TAREFA DE CLUSTERIZAÇÃO DA *SHELL* ORION *DATA MINING ENGINE*

O projeto da *Shell* Orion *Data Mining Engine* tem como objetivo o desenvolvimento de uma ferramenta gratuita, a fim de auxiliar o processo de descoberta de conhecimento em base de dados, como também, disponibilizar ao meio acadêmico informações sobre *data mining* e a modelagem matemática de diferentes algoritmos.

A implementação do algoritmo de lógica *fuzzy* Gustafson-Kessel no módulo de clusterização da *Shell* Orion é uma nova contribuição a este projeto, o qual já possui algumas tarefas e métodos de *data mining* disponíveis.

A fim de avaliar as funcionalidades de cada algoritmo, bases de dados foram utilizadas de acordo com a preferência de seu respectivo desenvolvedor. Neste trabalho optou-se pelo uso de uma base de dados médica, a qual demonstra informações de pacientes com sepse.

8.1 BASE DE DADOS

A base de dados utilizada foi fornecida pelo professor Dr. Felipe Dal Pizzol, do curso acadêmico de medicina da UNESC e contém registros de pacientes com Sepse da UTI do Hospital de Clínicas da cidade de Porto Alegre, Rio Grande do Sul.

A sepse é uma doença que causa uma resposta inflamatória quando o paciente possui uma infecção grave. Este problema pode estar somente em um orgão do doente, porém é provocada uma resposta com inflamação em todo organismo, na tentativa de combater a doença, prejudicando as funções de outros órgãos do corpo

humano. Considerando isto, a sepse também é conhecida comumente como infecção generalizada (CECIL; GOLDMAN; AUSIELLO, 2005).

A base contém um total de 96 registros, onde cada elemento possui 45 atributos contendo informações do paciente e de exames realizados durante a internação.

A base de dados necessitou passar pela etapa de pré-processamento, pois esta possuía atributos nominais. Considerando que o algoritmo Gustafson-Kessel trabalha apenas com valores numéricos, estes dados foram transformados a fim de permitir sua clusterização.

Os atributos nominais referiam-se a informações do paciente (*sexo*, *gravidad* e *desfecho*) e a resposta de perguntas. Nestes dados foi assimilado um número identificador, a fim de possibilitar a execução do algoritmo e sua posterior identificação.

Além disso, existiam atributos nulos, os quais foram substituídos pelo valor 0. No entanto, a base não possuía nenhum valor inconsistente, o que possibilitou a clusterização de todos os registros.

As características de cada atributo estão descritas na Tabela 3, considerando que as descrições técnicas necessárias para compreensão de alguns atributos foram retiradas do livro dos autores Russell L. Cecil, Lee Goldman e Dennis A. Ausiello, intitulado como *Cecil: tratado de medicina interna*.

Tabela 3. Base de dados de pacientes com sepse

Atributo	Descrição	Valor
caso	Número do caso registrado pelo hospital.	Número inteiro
idade	Idade do paciente.	Número inteiro
sexo	Sexo do paciente.	1 para masculino e 2 para feminino
gravidad	Gravidade em que o paciente chegou ao hospital.	1 para leve, 2 para grave e 3 para estado de choque
ira	Se foi registrado insuficiência renal aguda (perda das funções dos rins) durante a internação.	1 para sim e 2 para não
hepatico	Se o paciente possuía hepatite (inflamação no fígado).	1 para sim e 2 para não
vm	Se o paciente utilizou ventilação mecânica (aparelho que auxilia a respiração) durante a internação.	1 para sim e 2 para não
fio2, *fio21*, *fio22* e *fio23*	Se utilizou ventilação mecânica, qual a fração inspirada de oxigênio pelo paciente (em porcentagem). Os quatro atributos representam respectivamente os valores dos quatro primeiros dias de internação.	Número decimal
neurolog	Se o caso é neurológico, ou seja, se afetou o sistema nervoso central	1 para sim e 2 para não
vasopres	Se o paciente utilizou vasopressor (substância com o objetivo de aumentar a pressão arterial)	1 para sim e 2 para não
nora, *nora1*, *nora2* e *nora3*	Se utilizou vasopressor, qual a quantidade de noradrenalina utilizada (substância que mantém a pressão arterial estabilizada). Os quatro atributos representam respectivamente os valores dos quatro primeiros dias de internação.	Número decimal
apacheII	Grau de gravidade em que o paciente chegou na UTI. Este valor determina a previsão de sobrevivência do doente.	Número inteiro
mods, *mods2*, *mods3* e *mods4*	Grau de disfunção de múltiplos órgãos do paciente. Os quatro atributos representam respectivamente os valores dos quatro primeiros dias de internação.	Número inteiro
tempodesfecho	Tempo de desfecho do paciente (em dias).	Número inteiro
desfecho	Desfecho do paciente.	1 para óbito e 2 para cura
tba1, *tba2*, *tba3* e *tba4*	Marcador de oxidação de lipídios (alterações dos lipídios pelo oxigênio).	Número decimal
carbonyl, *carb2*, *carb3* e *carb4*	Marcador de oxidação de proteínas (alterações das proteínas pelo oxigênio).	Número decimal
sodcategorica	Categoria do superóxido-dismutase (enzima que realiza a dismutação do radical superóxido em peróxido de hidrogênio e oxigênio).	1 para *sod* maior que 5.5 e 2 para *sod* menor que 5.5
sod, *sod2*, *sod3* e *sod4*	Marcador do superóxido-dismutase. Os quatro atributos representam respectivamente os valores dos quatro primeiros dias de internação.	Número decimal
cat1, *cat2*, *cat3* e *cat4*	Quantidade de catalase (enzima que transforma o peróxido de hidrogênio em água e oxigênio) produzida pelo paciente. Os quatro atributos representam respectivamente os valores dos quatro primeiros dias de internação.	Número decimal
xo1, *xo2*, *xo3* e *xo4*	Quantidade de xantina oxidase (enzima que catalisa hipoxantina com oxigênio, produzindo ácido úrico e o radical superóxido) produzida pelo paciente. Os quatro atributos representam respectivamente os valores dos quatro primeiros dias de internação.	Número decimal

A fim de desenvolver o algoritmo Gustafson-Kessel, esta pesquisa adotou uma metodologia com o objetivo de compreender os temas abordados, descrever a modelagem matemática do método e implementá-lo no módulo de clusterização da *Shell* Orion *Data Mining Engine*.

8.2 METODOLOGIA

As seguintes etapas foram adotadas pela metodologia do desenvolvimento do algoritmo Gustafson-Kessel na *Shell* Orion: levantamento bibliográfico; modelagem do módulo do algoritmo Gustafson-Kessel; demonstração matemática do algoritmo Gustafson-Kessel; implementação e realização de testes.

A pesquisa bibliográfica deste trabalho fundamentou-se na descrição e compreensão de todos os temas envolvidos, como o processo de *data mining*, a tarefa de clusterização, o método de lógica *fuzzy* e o funcionamento do algoritmo Gustafson-Kessel.

Os estudos realizados tiveram como foco principal o entendimento do método de lógica *fuzzy* e do algoritmo Gustafson-Kessel, o qual envolveu um período maior em relação as outras etapas, devido a complexidade de suas fórmulas matemáticas e a ausência de bibliografia que explicassem detalhadamente as etapas envolvidas no processo de clusterização por meio deste método.

8.2.1 Modelagem do Módulo do Algoritmo Gustafson-Kessel

A modelagem do módulo do algoritmo Gustafson-Kessel na *Shell* Orion *Data Mining Engine* utilizou os padrões da *Unified Modeling Language*[30] (UML), a fim de desenvolver os diagramas de caso de uso, seqüência e atividades. O ambiente de desenvolvimento adotado para a modelagem dos diagramas foi o Netbeans[31] 6.0.1.

O diagrama de caso de uso demonstra as funções a serem realizadas pelo usuário no módulo do algoritmo e as atividades do sistema (Figura 34):

a) **informar parâmetros de entrada**: o usuário informa a quantidade de *clusters*, o valor do parâmetro de *fuzzyficação*, a quantidade máxima de iterações e o valor do erro. Após isso, o usuário solicita ao sistema a execução do algoritmo;

b) **executar o algoritmo**: no momento em que a solicitação do usuário é enviada, o sistema inicia a clusterização por meio do algoritmo Gustafson-Kessel e gera os resultados.

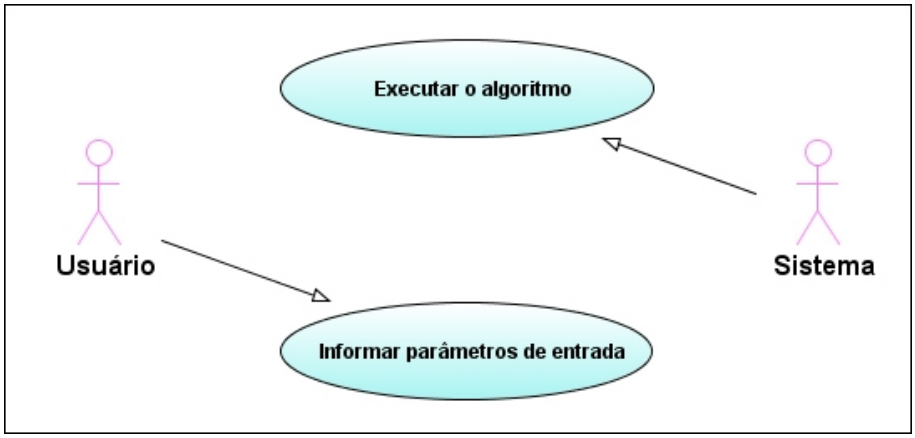

Figura 34. Diagrama de caso de uso do módulo do algoritmo Gustafson-Kessel

A Figura 35 demonstra o diagrama de seqüência, que descreve as interações entre os elementos usuário, algoritmo Gustafson-Kessel e resultados, durante um

[30] A liguagem de modelagem unificada tem como objetivo auxiliar a arquitetura das classes e dos métodos do sistema, a fim de facilitar sua compreensão (PENDER, 2004).
[31] Netbeans disponível em (http://www.netbeans.org).

período de tempo. O usuário solicita a interface inicial do método Gustafson-Kessel, a fim de inserir os parâmetros de entrada. Após isso, o algoritmo é executado e gera os resultados, retornando-os ao usuário.

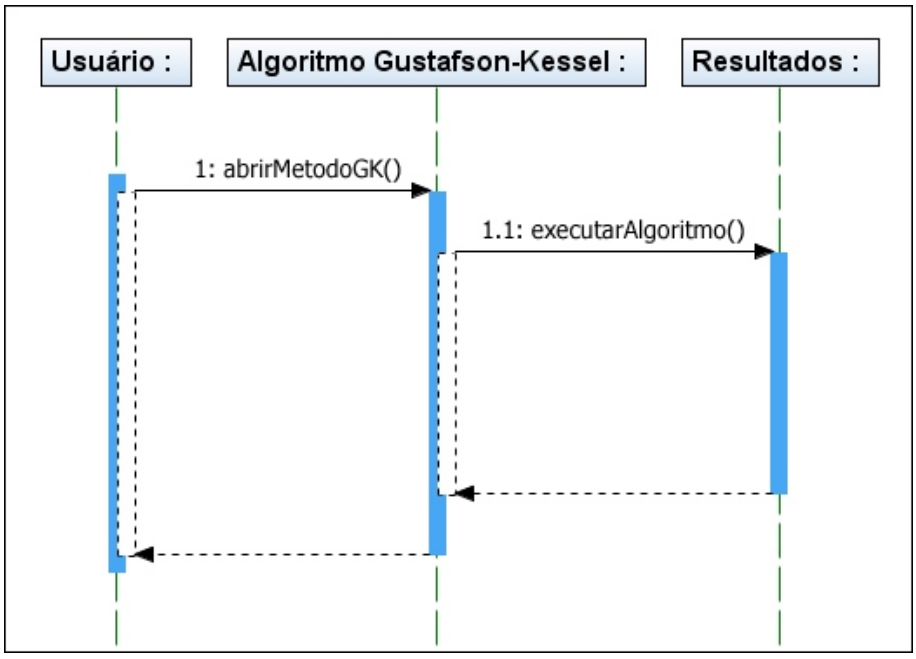

Figura 35. Diagrama de seqüência do módulo do algoritmo Gustafson-Kessel

A representação do fluxo de tarefas do usuário é demonstrado no diagrama de atividades, que descreve uma série de acontecimentos de acordo com as ações executadas pelo usuário (Figura 36):

a) **informa parâmetros de entrada**: o usuário informa os parâmetros de entrada necessários;

b) **solicita execução do algoritmo**: a execução do algoritmo é solicitada pelo usuário;

c) **processa o algoritmo Gustafson-Kessel**: o sistema inicia a execução do algoritmo;

d) **gera os resultados do algoritmo**: os resultados são gerados pelo sistema em forma de texto, gráfico e grupos;

e) **visualiza os resultados**: o usuário visualiza os diferentes resultados do algoritmo Gustafson-Kessel gerados pelo sistema.

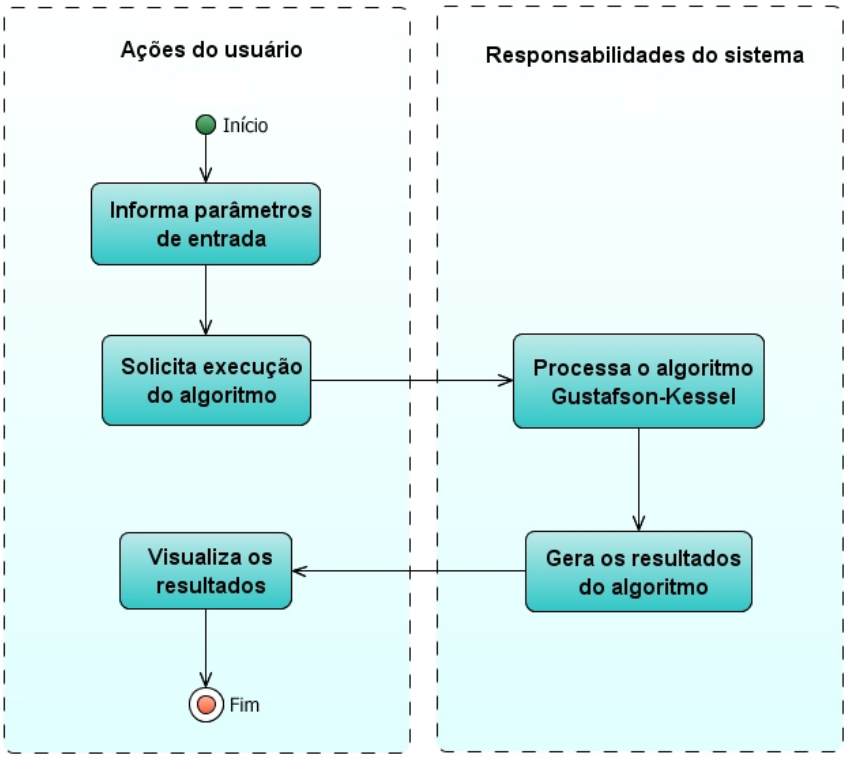

Figura 36. Diagrama de atividades do módulo do algoritmo Gustafson-Kessel

8.2.2 Demonstração Matemática do Algoritmo Gustafson-Kessel

Nesta etapa realizou-se a modelagem matemática do algoritmo Gustafson-Kessel, a fim de compreender os cálculos executados durante o processo de clusterização por meio deste método de lógica *fuzzy*.

Os conceitos e formalismos matemáticos utilizados na modelagem foram baseados principalmente no artigo dos autores do algoritmo, Donald E. Gustafson e William C. Kessel, intitulado como *Fuzzy clustering with fuzzy covariance matrix*, publicado em 1979 em uma conferência sobre controle e decisão na cidade de San Diego, Califórina.

Os cálculos demonstrados nesta seção são parciais, pois para compreender o algoritmo não é preciso apresentar a modelagem completa, visto que esta é muito extensa. No entanto, se houver necessidade de verificar todos os cálculos efetuados, estes estão demonstrados no Apêndice A.

Na demonstração dos cálculos utilizou-se uma base de dados que foi gerada com valores randômicos e possui quatro elementos (x_1, x_2, x_3, x_4), sendo que cada um destes contém três atributos (a, b, c). A Tabela 4 demonstra os valores desta base de dados.

Tabela 4. Base de dados utilizada na modelagem matemática do algoritmo

Atributos	x_1	x_2	x_3	x_4
a	0	1	1	0
b	0.6	0.4	0.9	0.7
c	1.2	1.3	1.3	1.0

Definidos os dados de entrada, deve-se escolher os valores dos parâmetros do algoritmo. Nesta etapa da pesquisa foram utilizados os seguintes valores:

a) **quantidade de *clusters***: especifica o número de grupos que será encontrado. Neste caso, optou-se pelo valor 2;

b) **parâmetro de *fuzzyficação* (*m*)**: determina o grau de *fuzzyficação* entre os elementos. Foi definido o número padrão do algoritmo, o valor 2;

c) **taxa de erro (ε)**: estipula a parada do algoritmo. Neste caso, foi utilizado o valor 10^{-05} (0.00001) apenas para exemplificar os cálculos de erro;

Inicialmente, deve-se atribuir os primeiros valores da matriz de pertinências *U*. Os valores inseridos (Tabela 5) também foram aleatórios, considerando a propriedade de soma das pertinências envolvidas:

$$\sum_{i=1}^{c} u_{ij} = 1$$

Tabela 5. Matriz de pertinência utilizada na modelagem matemática do algoritmo

Pertinências	x_1	x_2	x_3	x_4
c_1	0.67	0.41	0.24	0.5
c_2	0.33	0.59	0.76	0.5

1. Na primeira fase do algoritmo calcula-se o centro dos *clusters*, os quais devem ser encontrados por meio da seguinte equação:

$$c_i = \frac{\sum_{j=1}^{p} (u_{ij})^m x_j}{\sum_{j=1}^{p} (u_{ij})^m}$$

Nesta equação, é feito um somatório de todos os vetores de dados disponíveis, onde cada vetor é multiplicado por sua respectiva pertinência em relação ao atual *cluster*, elevada ao grau de *fuzzyficação*. Todos estes valores são divididos pela soma de todas as pertinências envolvidas elevadas ao grau de *fuzzyficação*.

Considerando isto, tem-se com a substituição dos valores relacionados ao primeiro *cluster*:

$$c_1 = \frac{(u_{11})^m \cdot x_1 + (u_{12})^m \cdot x_2 + (u_{13})^m \cdot x_3 + (u_{14})^m \cdot x_4}{(u_{11})^m + (u_{12})^m + (u_{13})^m + (u_{14})^m}$$

$$c_1 = \frac{0.67^2 \cdot \begin{bmatrix} 0 \\ 0.6 \\ 1.2 \end{bmatrix} + 0.41^2 \cdot \begin{bmatrix} 1 \\ 0.4 \\ 1.3 \end{bmatrix} + 0.24^2 \cdot \begin{bmatrix} 1 \\ 0.9 \\ 1.3 \end{bmatrix} + 0.5^2 \cdot \begin{bmatrix} 0 \\ 0.7 \\ 1.0 \end{bmatrix}}{0.67^2 + 0.41^2 + 0.24^2 + 0.5^2}$$

A operação de multiplicação dos vetores de dados pelos seus graus de pertinência envolve todos os valores de cada vetor:

$$c_1 = \frac{0.4489 \cdot \begin{bmatrix} 0 \\ 0.6 \\ 1.2 \end{bmatrix} + 0.1681 \cdot \begin{bmatrix} 1 \\ 0.4 \\ 1.3 \end{bmatrix} + 0.0576 \cdot \begin{bmatrix} 1 \\ 0.9 \\ 1.3 \end{bmatrix} + 0.25 \cdot \begin{bmatrix} 0 \\ 0.7 \\ 1.0 \end{bmatrix}}{0.4489 + 0.1681 + 0.0576 + 0.25}$$

$$c_1 = \frac{\begin{bmatrix} 0 \\ 0.26934 \\ 0.53868 \end{bmatrix} + \begin{bmatrix} 0.1681 \\ 0.06724 \\ 0.21853 \end{bmatrix} + \begin{bmatrix} 0.0576 \\ 0.05184 \\ 0.07488 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.175 \\ 0.25 \end{bmatrix}}{0.4489 + 0.1681 + 0.0576 + 0.25}$$

A adição de vetores envolve a soma dos valores que estão em uma mesma posição. Neste caso, cada linha é somada separadamente. Considerando isto, obtém-se:

$$c_1 = \frac{\begin{bmatrix} 0.2257 \\ 0.56342 \\ 1.08209 \end{bmatrix}}{0.9246}$$

Por fim, os valores do vetor obtido são divididos individualmente pela soma dos graus de pertinência:

$$c_1 = \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix}$$

Executando o mesmo procedimento para o segundo *cluster*, encontra-se como resultado o seguinte vetor:

$$c_2 = \begin{bmatrix} 0.72061 \\ 0.70016 \\ 1.23314 \end{bmatrix}$$

Os vetores obtidos nos cálculos contêm os valores que determinam os diferentes centros de cada *cluster*, considerando que cada atributo da base de dados possuirá um número. A Tabela 6 demonstra os centros obtidos:

Tabela 6. Centro dos *clusters* encontrados

Atributos	c_1	c_2
a	0.24411	0.72061
b	0.60937	0.70016
c	1.17033	1.23314

2. Após determinar os centros dos *clusters*, deve-se calcular as matrizes de covariância *fuzzy* de cada *cluster*, por meio da equação:

$$F_i = \frac{\sum_{j=1}^{p} (u_{ij})^m (x_j - c_i)(x_j - c_i)^{\mathrm{T}}}{\sum_{j=1}^{p} (u_{ij})^m}$$

Nesta equação executa-se um somatório dos graus de pertinência elevados ao parâmetro de *fuzzyficação*. Este valor é multiplicado pela subtração entre os dados e os centros.

A próxima multiplicação envolve a mesma subtração entre os dados e os centros, porém na forma transposta. Esta propriedade de matrizes determina a inversão de linhas por colunas e quando aplicada em vetores, estes são invertidos de direção. Neste caso, como tem-se vetores verticais, estes são transformados em vetores horizontais.

Considerando os valores do primeiro *cluster* e substituindo seus respectivos valores na equação, tem-se:

$$F_1 = \frac{\begin{array}{l}(u_{11})^m \cdot (x_1 - c_1) \cdot (x_1 - c_1)^{\mathrm{T}} + \\ (u_{12})^m \cdot (x_2 - c_1) \cdot (x_2 - c_1)^{\mathrm{T}} + \\ (u_{13})^m \cdot (x_3 - c_1) \cdot (x_3 - c_1)^{\mathrm{T}} + \\ (u_{14})^m \cdot (x_4 - c_1) \cdot (x_4 - c_1)^{\mathrm{T}}\end{array}}{(u_{11})^m + (u_{12})^m + (u_{13})^m + (u_{14})^m}$$

$$F_1 = \frac{\begin{array}{l}0.67^2 \cdot \left(\begin{bmatrix} 0.0 \\ 0.6 \\ 1.2 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right) \cdot \left([0.0, 0.6, 1.2] - [0.24411, 0.60937, 1.17033] \right) + \\ 0.41^2 \cdot \left(\begin{bmatrix} 1.0 \\ 0.4 \\ 1.3 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right) \cdot \left([1.0, 0.4, 1.3] - [0.24411, 0.60937, 1.17033] \right) + \\ 0.24^2 \cdot \left(\begin{bmatrix} 1.0 \\ 0.9 \\ 1.3 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right) \cdot \left([1.0, 0.9, 1.3] - [0.24411, 0.60937, 1.17033] \right) + \\ 0.50^2 \cdot \left(\begin{bmatrix} 0.0 \\ 0.7 \\ 1.0 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right) \cdot \left([0.0, 0.7, 1.0] - [0.24411, 0.60937, 1.17033] \right)\end{array}}{0.67^2 + 0.41^2 + 0.24^2 + 0.50^2}$$

Após a substituição dos números, subtrai-se os valores dos vetores individualmente conforme suas posições:

$$F_1 = \frac{\begin{array}{l} 0.67^2 \cdot \begin{bmatrix} -0.24411 \\ -0.00937 \\ 0.02967 \end{bmatrix} \cdot [-0.24411, -0.00937, 0.02967] + \\ 0.41^2 \cdot \begin{bmatrix} 0.75589 \\ -0.20937 \\ 0.12967 \end{bmatrix} \cdot [0.75589, -0.20937, 0.12967] + \\ 0.24^2 \cdot \begin{bmatrix} 0.75589 \\ 0.29063 \\ 0.12967 \end{bmatrix} \cdot [0.75589, 0.29063, 0.12967] + \\ 0.50^2 \cdot \begin{bmatrix} -0.24411 \\ 0.09063 \\ -0.17033 \end{bmatrix} \cdot [-0.24411, 0.09063, -0.17033] \end{array}}{0.67^2 + 0.41^2 + 0.24^2 + 0.50^2}$$

Efetuando a multiplicação dos graus de pertinência com o primeiro vetor encontrado, obtém-se:

$$F_1 = \frac{\begin{array}{l} 0.4489 \cdot \begin{bmatrix} -0.24411 \\ -0.00937 \\ 0.02967 \end{bmatrix} \cdot [-0.24411, -0.00937, 0.02967] + \\ 0.1681 \cdot \begin{bmatrix} 0.75589 \\ -0.20937 \\ 0.12967 \end{bmatrix} \cdot [0.75589, -0.20937, 0.12967] + \\ 0.0576 \cdot \begin{bmatrix} 0.75589 \\ 0.29063 \\ 0.12967 \end{bmatrix} \cdot [0.75589, 0.29063, 0.12967] + \\ 0.25 \cdot \begin{bmatrix} -0.24411 \\ 0.09063 \\ -0.17033 \end{bmatrix} \cdot [-0.24411, 0.09063, -0.17033] \end{array}}{0.4489 + 0.1681 + 0.0576 + 0.25}$$

$$F_1 = \frac{\begin{bmatrix} -0.10958 \\ -0.0042 \\ 0.01332 \end{bmatrix} \cdot [-0.24411, -0.00937, 0.02967] + \begin{bmatrix} 0.12707 \\ -0.03519 \\ 0.0218 \end{bmatrix} \cdot [0.75589, -0.20937, 0.12967] + \begin{bmatrix} 0.04354 \\ 0.01674 \\ 0.00747 \end{bmatrix} \cdot [0.75589, 0.29063, 0.12967] + \begin{bmatrix} -0.06103 \\ 0.02266 \\ -0.04258 \end{bmatrix} \cdot [-0.24411, 0.09063, -0.17033]}{0.9246}$$

A criação da matriz acontece nesta etapa, onde utiliza-se a propriedade de multiplicação de matrizes. Esta operação determina que uma matriz *A* (*m* x *n*) somente pode ser multiplicada por uma matriz *B* (*n* x *p*). A matriz resultante será uma matriz (*m* x *p*).

É feito um somatório da multiplicação entre cada número de uma linha de *A* e cada valor de uma coluna de *B*, onde cada resultado será um elemento da nova matriz. Este processo se repete entre todas as linhas de *A* e todas as colunas de *B*. A Figura 37 exemplifica graficamente uma multiplicação entre uma matriz *A* (3x2) e uma matriz *B* (2x3).

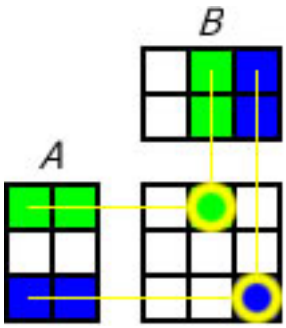

Figura 37. Exemplo gráfico da multiplicação de matrizes
Fonte: Adaptado de POOLE, D.; MONTEIRO, M. S. (2004)

Considerando que o vetor vertical é uma matriz (3x1) e o vetor horizontal é uma matriz (1x3), sua multiplicação resultará em uma matriz (3x3):

$$F_1 = \frac{\begin{bmatrix} 0.02675 & 0.00103 & -0.00325 \\ 0.00103 & 0.00004 & -0.00012 \\ -0.00325 & -0.00012 & 0.0004 \end{bmatrix} + \begin{bmatrix} 0.09605 & -0.0266 & 0.01648 \\ -0.0266 & 0.00737 & -0.00456 \\ 0.01648 & -0.00456 & 0.00283 \end{bmatrix} + \begin{bmatrix} 0.03291 & 0.01265 & 0.00565 \\ 0.01265 & 0.00487 & 0.00217 \\ 0.00565 & 0.00217 & 0.00097 \end{bmatrix} + \begin{bmatrix} 0.0149 & -0.00553 & 0.01039 \\ -0.00553 & 0.00205 & -0.00386 \\ 0.01039 & -0.00386 & 0.00725 \end{bmatrix}}{0.9246}$$

Somando individualmente cada elemento das matrizes encontradas, tem-se:

$$F_1 = \frac{\begin{bmatrix} 0.17061 & -0.01845 & 0.02927 \\ -0.01845 & 0.01433 & -0.00638 \\ 0.02927 & -0.00638 & 0.01144 \end{bmatrix}}{0.9246}$$

Por fim, encontra-se a matriz de covariância *fuzzy* por meio da divisão de cada elemento da matriz pela soma das pertinências:

$$F_1 = \begin{bmatrix} 0.18452 & -0.01996 & 0.03165 \\ -0.01996 & 0.0155 & -0.0069 \\ 0.03165 & -0.0069 & 0.01238 \end{bmatrix}$$

Aplicando as mesmas operações ao segundo *cluster*, tem-se:

$$F_2 = \begin{bmatrix} 0.20133 & 0.00852 & 0.04818 \\ 0.00852 & 0.04322 & 0.00086 \\ 0.04818 & 0.00086 & 0.01389 \end{bmatrix}$$

3. A próxima fase consiste no cálculo das distâncias dos elementos em relação aos *clusters*. A fim de atingir este objetivo, é utilizada a seguinte equação:

$$d^2(x_j, c_i) = (x_j - c_i)^{\mathrm{T}} M_i (x_j - c_i)$$

Onde:

$$M_i = \sqrt[N]{\det(F_i)}\, F_i^{-1}$$

Considerando a equação acima, deve-se encontrar primeiro o valor da matriz de covariância modificada M_i. Inicialmente, calcula-se o determinante da matriz de covariância *fuzzy* a fim de facilitar o cálculo da matriz inversa.

3.1. O cálculo do determinante de uma matriz de (3x3) pode ser obtido por meio da equação:

$$A=\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \tag{9}$$

$$\det(A)=(a\cdot e\cdot i)+(b\cdot f\cdot g)+(c\cdot d\cdot h)-(c\cdot e\cdot g)-(b\cdot d\cdot i)-(a\cdot f\cdot h)$$

Substituindo os valores da matriz de covariância *fuzzy* do primeiro *cluster*, tem-se:

$$\begin{aligned}\det(F_1)=&(F_{1(1,1)}\cdot F_{1(2,2)}\cdot F_{1(3,3)})+(F_{1(1,2)}\cdot F_{1(2,3)}\cdot F_{1(3,1)})+\\&(F_{1(1,3)}\cdot F_{1(2,1)}\cdot F_{1(3,2)})-(F_{1(1,3)}\cdot F_{1(2,2)}\cdot F_{1(3,1)})-\\&(F_{1(1,2)}\cdot F_{1(2,1)}\cdot F_{1(3,3)})-(F_{1(1,1)}\cdot F_{1(2,3)}\cdot F_{1(3,2)})\end{aligned}$$

$$\begin{aligned}\det(F_1)=&(0.18452\cdot 0.0155\cdot 0.01238)+(-0.01996\cdot -0.0069\cdot 0.03165)+\\&(0.03165\cdot -0.01996\cdot -0.0069)-(0.03165\cdot 0.0155\cdot 0.03165)-\\&(-0.01996\cdot -0.01996\cdot 0.01238)-(0.18452\cdot -0.0069\cdot -0.0069)\end{aligned}$$

$$\begin{aligned}\det(F_1)=&3.54075428\,e^{-05}+0.43589646\,e^{-05}+0.43589646\,e^{-05}-\\&1.552669875\,e^{-05}-0.4932211808\,e^{-05}-0.87849972\,e^{-05}\end{aligned}$$

$$\det(F_1)=1.4881564242\,e^{-05}$$

3.2. Após encontrar o determinante, deve-se calcular a matriz inversa da matriz de covariância *fuzzy*. A equação seguinte demonstra o cálculo da matriz inversa:

$$A^{-1}=\frac{1}{|A|}\operatorname{adj}(A) \tag{10}$$

Onde:

a) $|\boldsymbol{A}|$: determinante da matriz A;

b) **adj(*A*)**: matriz adjunta da matriz A.

Esta equação divide a adjunta de *A* pelo determinante de *A*. A matriz adjunta contêm os determinantes de todas as matrizes menores possíveis de *A*. Considerando a matriz de exemplo *A* da equação (9), sua adjunta pode ser obtida pela equação (11):

$$\mathrm{adj}(A)=\begin{bmatrix} \begin{vmatrix} e & f \\ h & i \end{vmatrix} & \begin{vmatrix} c & b \\ i & h \end{vmatrix} & \begin{vmatrix} b & c \\ e & f \end{vmatrix} \\ \begin{vmatrix} f & d \\ i & g \end{vmatrix} & \begin{vmatrix} a & c \\ g & i \end{vmatrix} & \begin{vmatrix} c & a \\ f & d \end{vmatrix} \\ \begin{vmatrix} d & e \\ g & h \end{vmatrix} & \begin{vmatrix} b & a \\ h & g \end{vmatrix} & \begin{vmatrix} a & b \\ d & e \end{vmatrix} \end{bmatrix} \tag{11}$$

Onde a determinante de uma matriz (2x2) é descrita pela equação (12):

$$B=\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$
$$|B|=(a\cdot d)-(b\cdot c) \tag{12}$$

Substituindo os valores da matriz de covariância *fuzzy* do primeiro *cluster*, obtém-se:

$$\mathrm{adj}(F_1)=\begin{bmatrix} \begin{vmatrix} F_{1(2,2)} & F_{1(2,3)} \\ F_{1(3,2)} & F_{1(3,3)} \end{vmatrix} & \begin{vmatrix} F_{1(1,3)} & F_{1(1,2)} \\ F_{1(3,3)} & F_{1(3,2)} \end{vmatrix} & \begin{vmatrix} F_{1(1,2)} & F_{1(1,3)} \\ F_{1(2,2)} & F_{1(2,3)} \end{vmatrix} \\ \begin{vmatrix} F_{1(2,3)} & F_{1(2,1)} \\ F_{1(3,3)} & F_{1(3,1)} \end{vmatrix} & \begin{vmatrix} F_{1(1,1)} & F_{1(1,3)} \\ F_{1(3,1)} & F_{1(3,3)} \end{vmatrix} & \begin{vmatrix} F_{1(1,3)} & F_{1(1,1)} \\ F_{1(2,3)} & F_{1(2,1)} \end{vmatrix} \\ \begin{vmatrix} F_{1(2,1)} & F_{1(2,2)} \\ F_{1(3,1)} & F_{1(3,2)} \end{vmatrix} & \begin{vmatrix} F_{1(1,2)} & F_{1(1,1)} \\ F_{1(3,2)} & F_{1(3,1)} \end{vmatrix} & \begin{vmatrix} F_{1(1,1)} & F_{1(1,2)} \\ F_{1(2,1)} & F_{1(2,2)} \end{vmatrix} \end{bmatrix}$$

$$\mathrm{adj}(F_1)=\begin{bmatrix} \begin{vmatrix} 0.0155 & -0.0069 \\ -0.0069 & 0.01238 \end{vmatrix} & \begin{vmatrix} 0.03165 & -0.01996 \\ 0.01238 & -0.0069 \end{vmatrix} & \begin{vmatrix} -0.01996 & 0.03165 \\ 0.0155 & -0.0069 \end{vmatrix} \\ \begin{vmatrix} -0.0069 & -0.01996 \\ 0.01238 & 0.03165 \end{vmatrix} & \begin{vmatrix} 0.18452 & 0.03165 \\ 0.03165 & 0.01238 \end{vmatrix} & \begin{vmatrix} 0.03165 & 0.18452 \\ -0.0069 & -0.01996 \end{vmatrix} \\ \begin{vmatrix} -0.01996 & 0.0155 \\ 0.03165 & -0.0069 \end{vmatrix} & \begin{vmatrix} -0.01996 & 0.18452 \\ -0.0069 & 0.03165 \end{vmatrix} & \begin{vmatrix} 0.18452 & -0.01996 \\ -0.01996 & 0.0155 \end{vmatrix} \end{bmatrix}$$

Considerando que cada elemento da adjunta é um determinante de uma matriz (2x2), deve-se calcular os valores separadamente:

$$\text{adj}(F_1)=$$

$$\begin{bmatrix} (0.00019189)-(0.0004761) & (-0.000218385)-(-0.0002471048) & (0.000137724)-(0.000490575) \\ (-0.000218385)-(-0.0002471048) & (0.0022843576)-(0.0010017255) & (-0.000631734)-(-0.001273188) \\ (0.000137724)-(0.000490575) & (-0.000631734)-(-0.001273188) & (0.00286006)-(0.0003984016) \end{bmatrix}$$

$$\text{adj}(F_1)=\begin{bmatrix} 0.00014428 & 0.0000287198 & -0.000352851 \\ 0.0000287198 & 0.0012826351 & 0.000641454 \\ -0.000352851 & 0.000641454 & 0.0024616584 \end{bmatrix}$$

Após isto, substitui-se os valores encontrados na equação da matriz inversa:

$$F_1^{-1}=\frac{1}{|F_1|}\text{adj}(F_1)$$

$$F_1^{-1}=\frac{\begin{bmatrix} 0.00014428 & 0.0000287198 & -0.000352851 \\ 0.0000287198 & 0.0048383355 & 0.000641454 \\ -0.000352851 & 0.000641454 & 0.0024616584 \end{bmatrix}}{1.4881564242\,e^{-05}}$$

Por fim, basta dividir a adjunta pelo determinante, obtendo a seguinte matriz inversa:

$$F_1^{-1}=\begin{bmatrix} 9.69522 & 1.92989 & -23.71061 \\ 1.92989 & 86.18953 & 43.10394 \\ -23.71061 & 43.10394 & 165.41664 \end{bmatrix}$$

3.3. Ao encontrar o determinante e a matriz inversa, pode-se determinar a matriz de covariância modificada M_i. Substituindo os valores, obtém-se:

$$M_1=\sqrt[N]{\det(F_1)}\,F_1^{-1}$$

$$M_1=\sqrt[3]{1.4881564242\,e^{-05}}\cdot\begin{bmatrix} 9.69522 & 1.92989 & -23.71061 \\ 1.92989 & 86.18953 & 43.10394 \\ -23.71061 & 43.10394 & 165.41664 \end{bmatrix}$$

Neste caso a raiz será cúbica, pois a base de dados utilizada possui três dimensões. Considerando isto, encontra-se:

$$M_1=0.024597040\cdot\begin{bmatrix} 9.69522 & 1.92989 & -23.71061 \\ 1.92989 & 86.18953 & 43.10394 \\ -23.71061 & 43.10394 & 165.41664 \end{bmatrix}$$

Por fim, a matriz de covariância modificada é encontrada pela multiplicação do resultado da raiz cúbica do determinante pelos elementos da matriz inversa:

$$M_1 = \begin{bmatrix} 0.23847 & 0.04747 & -0.58321 \\ 0.04747 & 2.12001 & 1.06023 \\ -0.58321 & 1.06023 & 4.06876 \end{bmatrix}$$

3.4. Após encontrar a matriz de covariância modificada, pode-se calcular as distâncias dos elementos em relação ao primeiro *cluster*. Considerando o primeiro elemento (x_1), tem-se:

$$d^2(x_1, c_1) = (x_1 - c_1)^{\mathrm{T}} M_1 (x_1 - c_1)$$

$$d^2(x_1, c_1) = \left([0.0, 0.6, 1.2] - [0.24411, 0.60937, 1.17033] \right) \cdot M_1 \cdot \left(\begin{bmatrix} 0.0 \\ 0.6 \\ 1.2 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right)$$

Subtraindo os seguintes vetores, encontra-se:

$$d^2(x_1, c_1) = [-0.24411, -0.00937, 0.02967] \cdot M_1 \cdot \begin{bmatrix} -0.24411 \\ -0.00937 \\ 0.02967 \end{bmatrix}$$

A matriz de covariância modificada é uma matriz (3x3) e considerando o último vetor como uma matriz (3x1), obtém-se com sua multiplicação um vetor vertical:

$$d^2(x_1, c_1) = [-0.24411, -0.00937, 0.02967] \cdot \begin{bmatrix} -0.07596 \\ 0 \\ 0.25315 \end{bmatrix}$$

Considerando o vetor horizontal como uma matriz (1x3) e o vetor vertical como uma matriz (3x1), encontra-se uma matriz (1x1). Este único valor encontrado será a distância do elemento em relação ao atual *cluster*:

$$d^2(x_1, c_1) = 0.02605$$

Realizando o mesmo procedimento para os outros elementos, obtém-se:

$$d^2(x_2, c_1) = 0.11068$$

$$d^2(x_3, c_1) = 0.37018$$

$$d^2(x_4, c_1) = 0.06634$$

Aplicando as equações no segundo *cluster*, encontra-se suas respectivas distâncias em relação aos elementos da base de dados:

$$d^2(x_1, c_2) = 0.29364$$

$$d^2(x_2, c_2) = 0.07313$$

$$d^2(x_3, c_2) = 0.03327$$

$$d^2(x_4, c_2) = 0.11254$$

4. Após encontrar os valores das distâncias, pode-se executar a próxima fase do algoritmo, a atualização das pertinências, definida pela seguinte equação:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}^2}{d_{kj}^2} \right)^{\frac{1}{m-1}}}$$

O somatório da divisão das distâncias calcula a pertinência atualizada. Aplicando está formula ao primeiro *cluster* e ao elemento inicial, tem-se:

$$u_{11} = \frac{1}{\left(\frac{d_{11}^2}{d_{11}^2} \right)^{\frac{1}{m-1}} + \left(\frac{d_{11}^2}{d_{12}^2} \right)^{\frac{1}{m-1}}}$$

$$u_{11} = \frac{1}{\left(\frac{0.02605}{0.02605} \right)^{\frac{1}{2-1}} + \left(\frac{0.02605}{0.29364} \right)^{\frac{1}{2-1}}}$$

$$u_{11} = \frac{1}{1 + 0.08871}$$

$$u_{11} = 0.91852$$

Considerando o mesmo processo para os outros elementos e *clusters*, encontra-se as outras pertinências, demonstradas na Tabela 7:

Tabela 7. Matriz de pertinências atualizada

Pertinências	x_1	x_2	x_3	x_4
c_1	0.91852	0.39786	0.08246	0.62914
c_2	0.08148	0.60214	0.91754	0.37086

5. Encontrando as pertinências atualizadas, deve-se calcular o erro do algoritmo com o objetivo de parar sua execução. Isto é determinado pela seguinte condição:

$$\|U^l - U^{l-1}\| \leq \varepsilon$$

Substituindo as duas matrizes de pertinências, a anterior e a atual, tem-se:

$$\left\| \begin{bmatrix} 0.91852 & 0.39786 & 0.08246 & 0.62914 \\ 0.08148 & 0.60214 & 0.91754 & 0.37086 \end{bmatrix} - \begin{bmatrix} 0.67 & 0.41 & 0.24 & 0.5 \\ 0.33 & 0.59 & 0.76 & 0.5 \end{bmatrix} \right\| \leq 10^{-05}$$

Subtraindo os valores, encontra-se:

$$\left\| \begin{bmatrix} 0.24852 & -0.01214 & -0.15754 & 0.12914 \\ -0.24852 & 0.01214 & 0.15754 & -0.12914 \end{bmatrix} \right\| \leq 0.00001$$

Considerando os valores absolutos (positivos), obtém-se o resultado final:

$$\begin{bmatrix} 0.24852 & 0.01214 & 0.15754 & 0.12914 \\ 0.24852 & 0.01214 & 0.15754 & 0.12914 \end{bmatrix} \leq 0.00001$$

Neste caso nenhum valor é menor que o erro, o que resultaria em uma próxima iteração do algoritmo, até que esta condição de parada fosse verdadeira. Para isso, seria necessário que todos os valores de pertinência fossem menores que o valor do erro determinado nos parâmetros de entrada. Se o algoritmo terminasse sua execução neste momento, os valores de pertinência de cada elemento e os centros dos *clusters* seriam seu resultado final e o processo de clusterização terminaria.

A compreensão de todos estes cálculos realizados pelo algoritmo de lógica *fuzzy* Gustafson-Kessel possibilitou sua implementação no módulo de clusterização da

Shell Orion. Após isto, foram realizados alguns testes a fim de avaliar o método desenvolvido.

8.2.3 Implementação e Realização de Testes

O algoritmo Gustafson-Kessel foi implementado no módulo de clusterização da *Shell* Orion *Data Mining Engine* por meio da linguagem de programação Java e do ambiente de programação Netbeans 6.0.1.

A fim de realizar os testes na *Shell* Orion, é necessário especificar o tipo de banco de dados. Como a ferramenta possibilita a conexão com diferentes *drivers*, nesta pesquisa optou-se pelo uso do *Hypersonic Structured Query Language Database*[32] (HSQLDB) 1.8.0, o qual é gratuito e não necessita de instalação no sistema, bastando apenas executá-lo.

Após especificar a conexão com o banco, deve-se efetuar os testes com uma base específica, a fim de apresentar as funcionalidades do algoritmo implementado. Nesta pesquisa utilizou-se os registros de pacientes com sepse da UTI do Hospital de Clínicas de Porto Alegre, Rio Grande do Sul. Os dados foram inseridos no HSQLDB a fim de possibilitar seu acesso pela *Shell* Orion.

Efetuada a inserção dos dados, deve-se conectar ao *driver* do banco pelo menu da *Shell* Orion *Arquivo*, submenu *Conectar.* Após isto, pode-se acessar o algoritmo pelo menu da etapa de *Data Mining*, submenu da tarefa de *Clusterização*, método de lógica *Fuzzy* e algoritmo *Gustafson-Kessel* (Figura 38).

[32] HSQLDB disponível em (http://hsqldb.org).

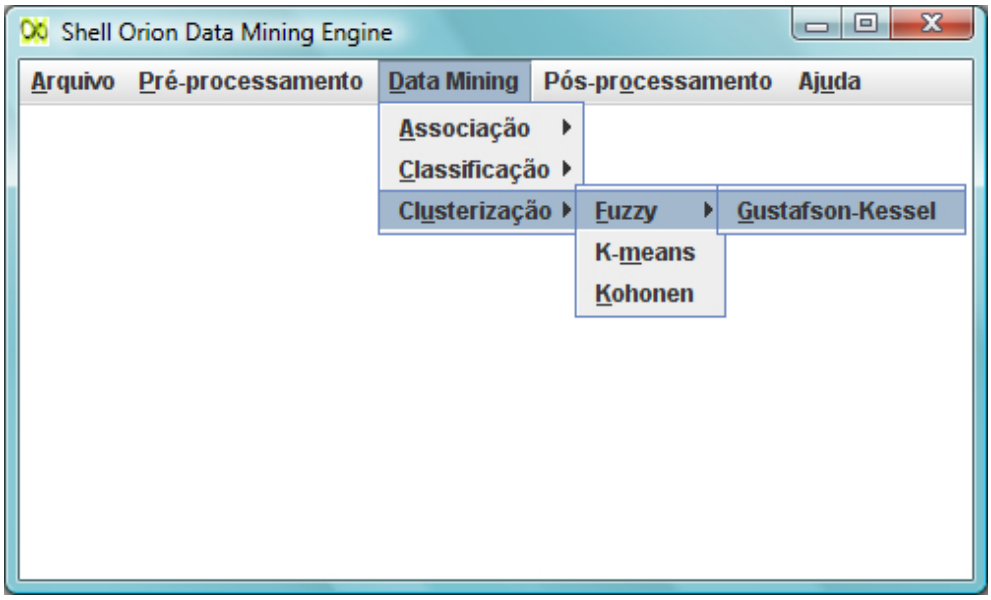

Figura 38. Acesso do algoritmo Gustafson-Kessel na *Shell* Orion

A clusterização por meio do algoritmo de lógica *fuzzy* Gustafson-Kessel necessita que alguns parâmetros de entrada sejam definidos, a fim de possibilitar a determinação correta dos grupos (Figura 39):

a) **quantidade de *clusters***: quantidade de grupos que o algoritmo irá encontrar;

b) **parâmetro de *fuzzyficação***: o grau de *fuzzyficação* dos elementos em relação aos seus *clusters*;

c) **quantidade de iterações**: a quantidade de ciclos que o algoritmo irá executar. Se for escolhido o valor zero, o algoritmo irá executar um número ilimitado de iterações e sua condição de parada será exclusivamente a taxa de erro;

d) **taxa de erro**: erro aceitável dos resultados determinados pelo algoritmo. Esta opção determina a condição de parada do método;

e) **atributos de entrada**: campos da base de dados que serão utilizados como valores de entrada nos cálculos de clusterização pelo algoritmo Gustafson-Kessel.

A definição dos parâmetros de entrada demonstra que por meio de uma base de dados, pode-se obter diferentes resultados, considerando que os valores podem ser selecionados de acordo com a preferência do usuário.

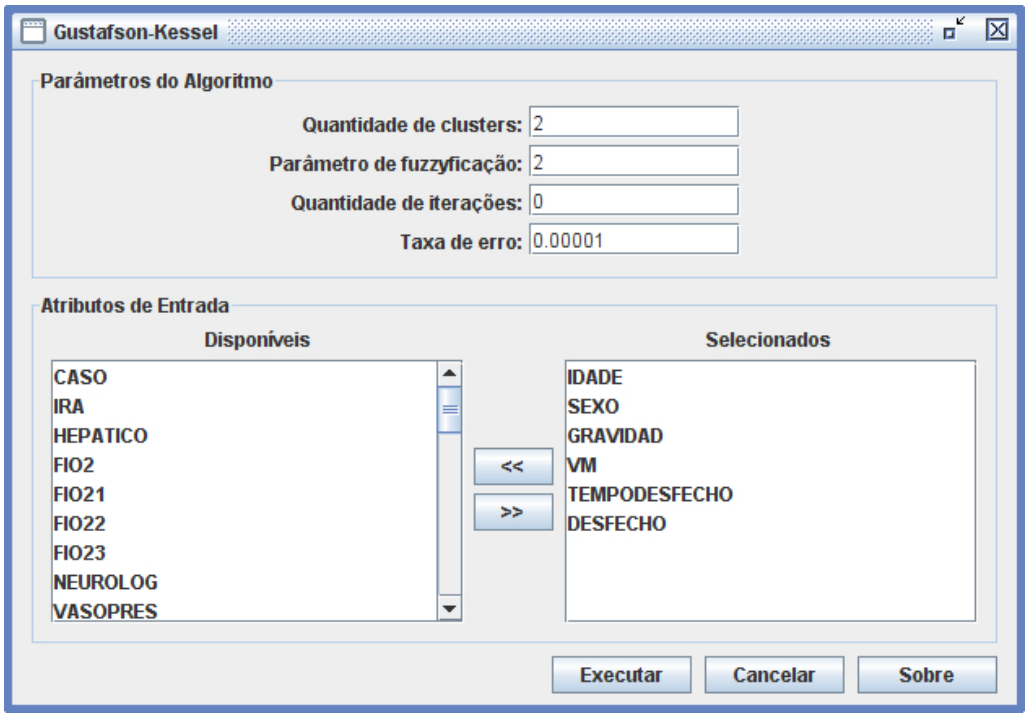

Figura 39. Algoritmo Gustafson-Kessel da *Shell* Orion *Data Mining Engine*

Após executar o algoritmo, os resultados podem ser visualizados de diferentes maneiras. A Figura 40 demonstra o resumo da clusterização pelo algoritmo Gustafson-Kessel, onde são apresentados os parâmetros e atributos de entrada, informações sobre os grupos, os centros dos *clusters* e o atributo de saída, que pode ser alterado por meio da opção *atributo de saída*.

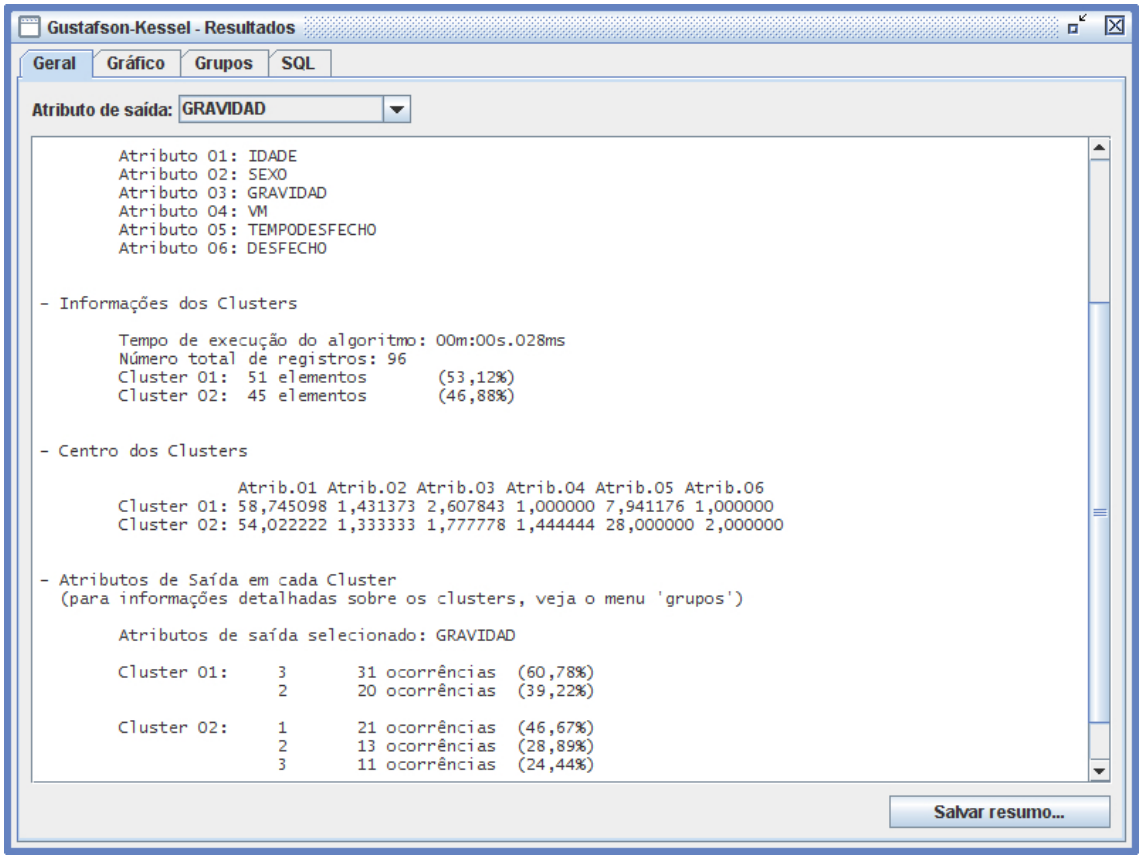

Figura 40. Resumo da clusterização por meio do algoritmo Gustafson-Kessel

Pode-se visualizar que o algoritmo encontrou os dois grupos. De acordo com o atributo de saída selecionado (*gravidad*), o primeiro *cluster* possui dados de pacientes com características graves e em estado de choque (valores 2 e 3), enquanto o segundo apresenta os 3 tipos de estados disponíveis na base (valores 1, 2 e 3). Estes dados obtidos pela clusterização podem ser exportados em um arquivo de texto, por meio do botão *Salvar resumo*.

Os resultados também podem ser visualizados em forma gráfica (Figura 41), onde é demonstrado os grupos encontrados por meio de *Principal Component Analysis*[33] (PCA). Este método realiza a transformação de uma base de dados contendo *n* dimensões em uma matriz de 2 dimensões, por meio de sucessivas decomposições dos dados, com o objetivo de possibilitar a projeção dos elementos em um gráfico.

[33] Guia sobre as diferentes implementações do PCA disponível em (http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf).

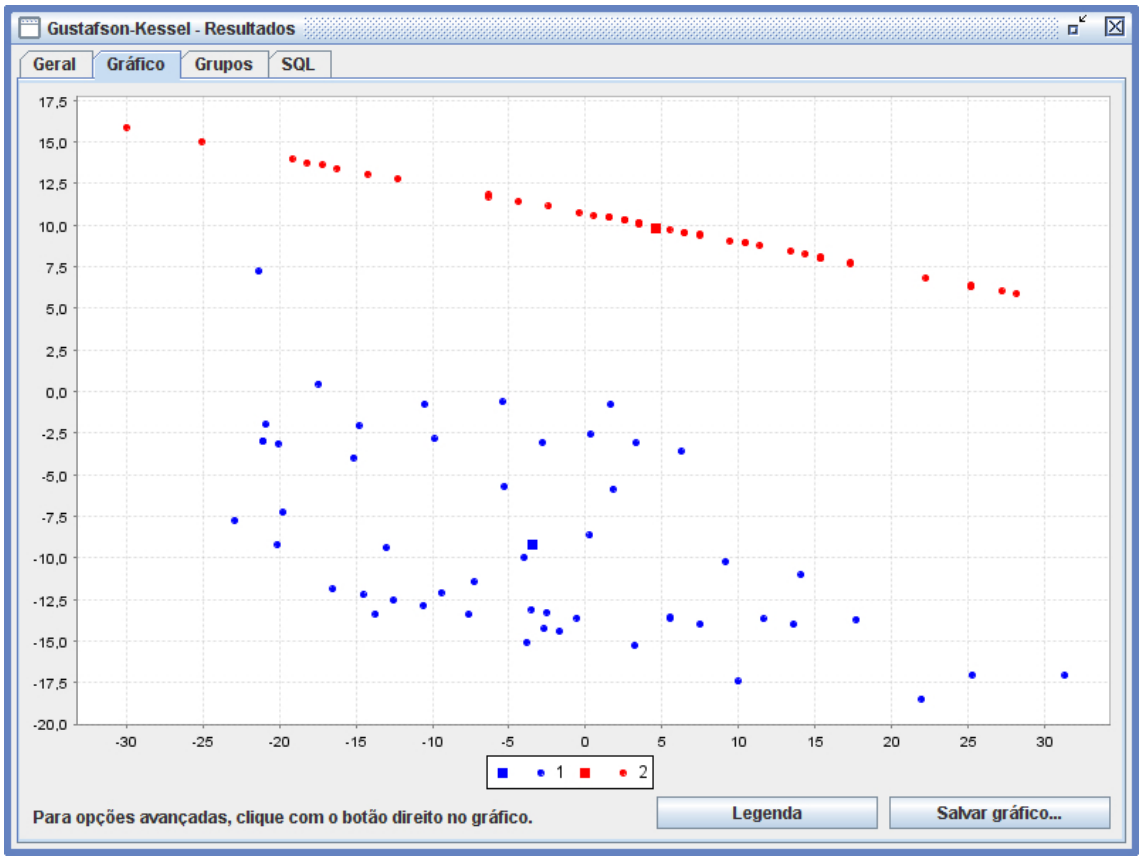

Figura 41. Gráfico construído por meio de PCA pelo algoritmo Gustafson-Kessel

A análise dos resultados também pode ser feita por meio de uma estrutura de árvore. Esta funcionalidade foi desenvolvida com o objetivo de facilitar a navegação entre os diferentes *clusters* encontrados, a fim de possibilitar a análise individual dos dados existentes. O início da árvore contém os grupos encontrados e a quantidade de elementos.

Ao expandir um *cluster* e algum dos elementos, pode-se visualizar o conteúdo de cada dado, considerando os valores inseridos nos atributos de entrada. Também pode-se visualizar o número do elemento em relação ao banco de dados, por meio da opção *Mostrar número do elemento,* e o grau de pertinência deste elemento em relação aos grupos encontrados, utilizando a opção *Mostrar pertinências*. A Figura 42 demonstra a estrutura de árvore gerada pelo algoritmo.

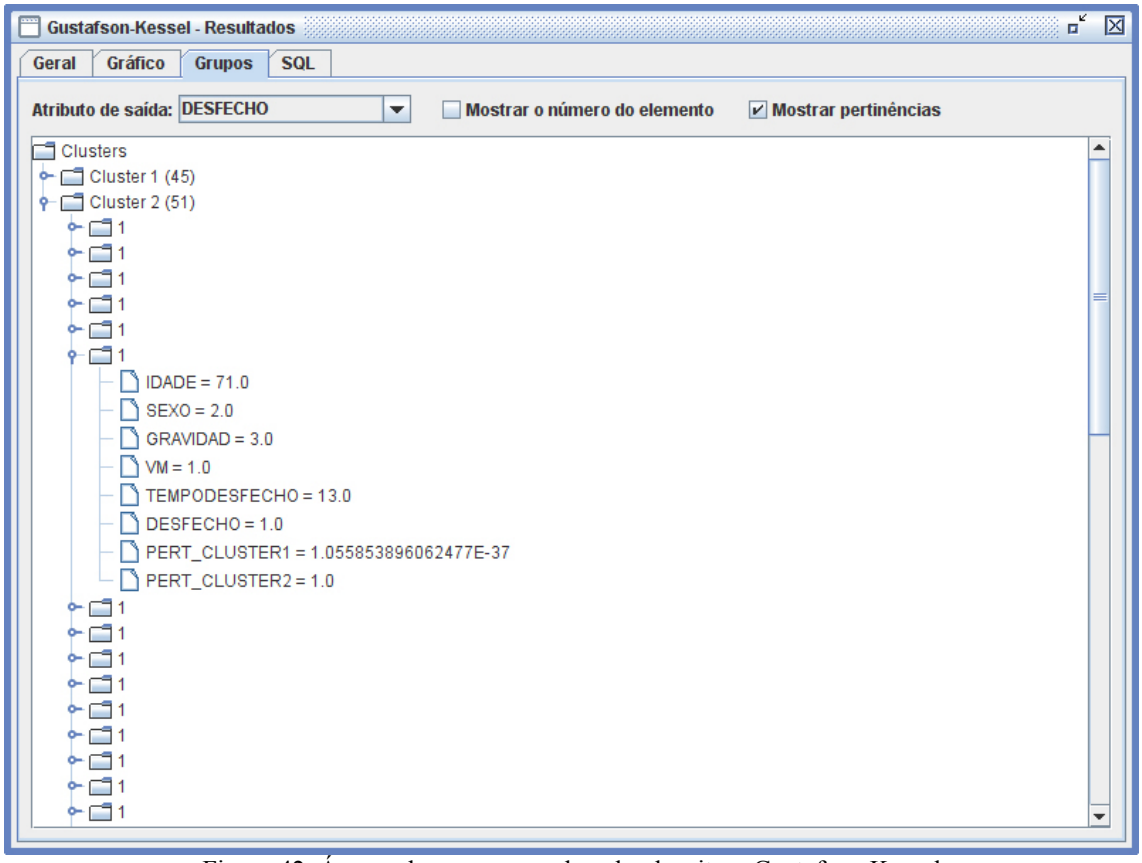

Figura 42. Árvore de grupos gerada pelo algoritmo Gustafson-Kessel

Os resultados obtidos podem ser exportados no formato de arquivo *Structured Query Language* (SQL), o que permite importar posteriormente os grupos obtidos para outras tarefas de *data mining*, como por exemplo a classificação, a fim de aumentar a descoberta de conhecimento na base de dados. A Figura 43 apresenta a tela de exportação do arquivo SQL, que pode ser inicializada pelo botão *Salvar SQL*.

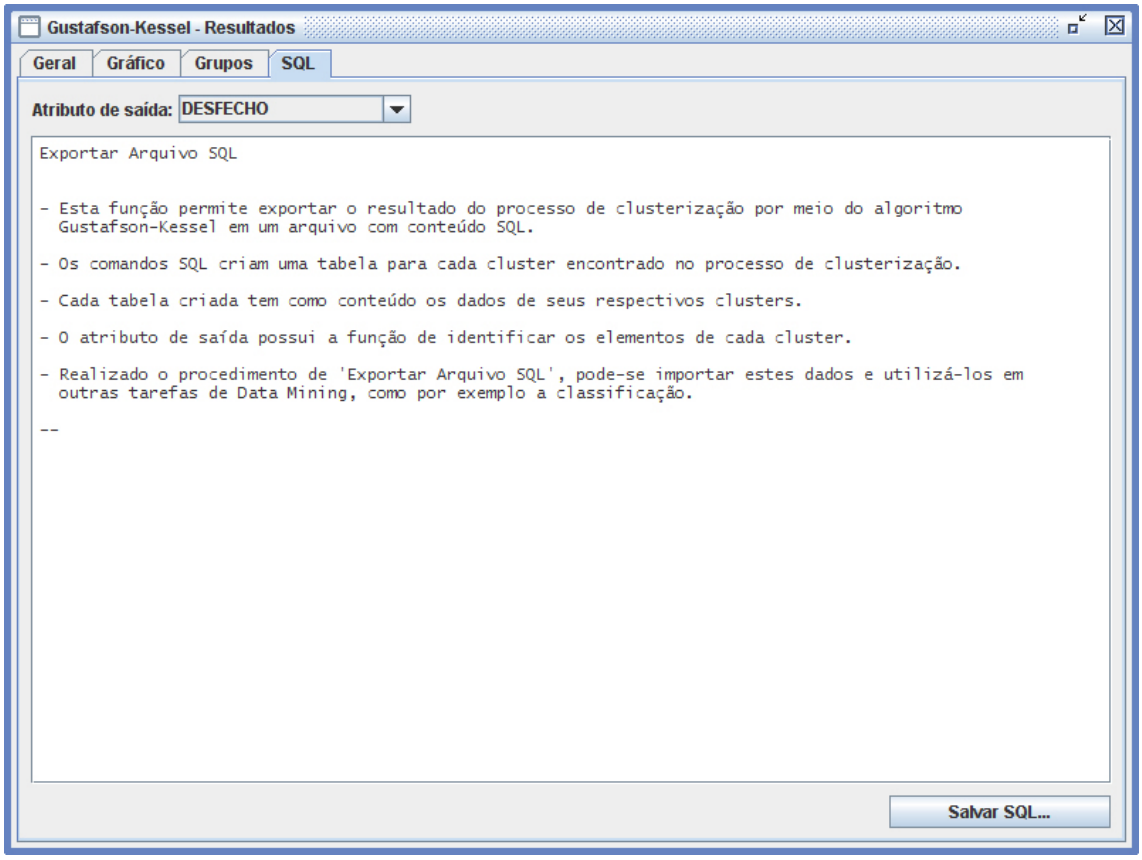

Figura 43. Exportar arquivo SQL

O atributo de saída pode ser escolhido pelo usuário, determinando o valor que representará os dados no arquivo gerado. A Figura 44 demonstra parte do conteúdo SQL criado pela operação de exportação.

```
-- Create table CLUSTER2

CREATE TABLE CLUSTER2(
        LABEL VARCHAR(4) NOT NULL,
        IDADE DOUBLE NOT NULL,
        SEXO DOUBLE NOT NULL,
        GRAVIDAD DOUBLE NOT NULL,
        VM DOUBLE NOT NULL,
        TEMPODESFECHO DOUBLE NOT NULL,
        DESFECHO DOUBLE NOT NULL
);

-- Insert data

INSERT INTO CLUSTER2 VALUES ('1', 75.0, 1.0, 3.0, 1.0, 7.0, 1.0);
INSERT INTO CLUSTER2 VALUES ('1', 57.0, 1.0, 3.0, 1.0, 4.0, 1.0);
INSERT INTO CLUSTER2 VALUES ('1', 71.0, 1.0, 3.0, 1.0, 3.0, 1.0);
INSERT INTO CLUSTER2 VALUES ('1', 41.0, 1.0, 2.0, 1.0, 9.0, 1.0);
INSERT INTO CLUSTER2 VALUES ('1', 69.0, 1.0, 3.0, 1.0, 3.0, 1.0);
```

Figura 44. Conteúdo do arquivo SQL

Considerando a possibilidade de não existir a matriz inversa durante a execução do algoritmo, o módulo interrompe sua execução e apresenta uma tela de erro (Figura 45). No entanto, o usuário pode optar por realizar novamente a clusterização ou visualizar os resultados obtidos até o momento.

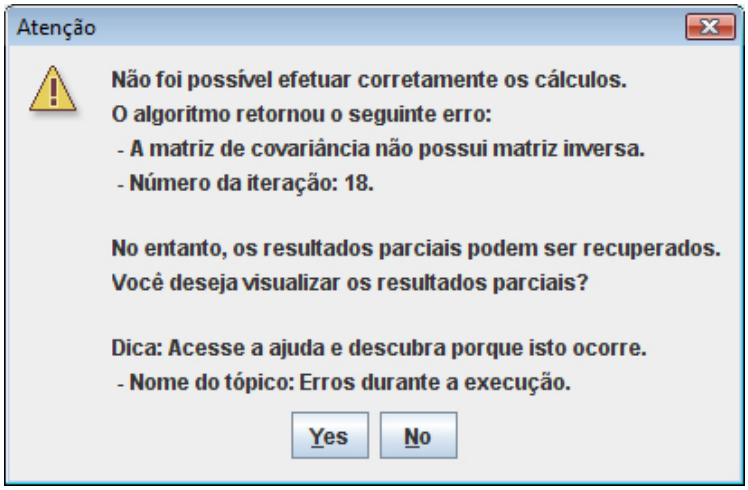

Figura 45: Erro de matriz inversa durante a execução do algoritmo.

Além destas funções, um arquivo de ajuda disponibiliza toda a documentação necessária a fim de facilitar a utilização do algoritmo Gustafson-Kessel e pode ser acessada pelo menu *Ajuda*, submenu *Conteúdo da ajuda*, tarefa de *Clusterização*, método de lógica *fuzzy* e algoritmo *Gustafson-Kessel*. A interface de ajuda é demonstrada na Figura 46.

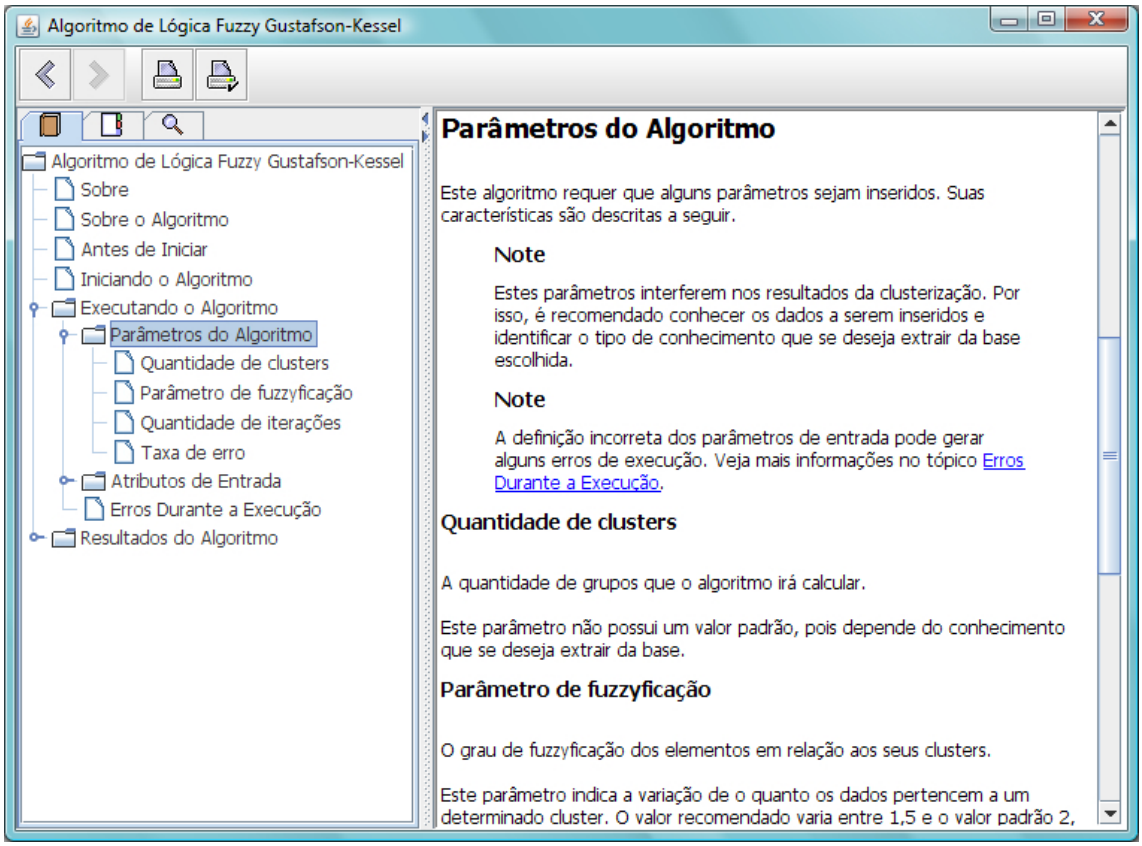

Figura 46. Documentação de ajuda do algoritmo Gustafson-Kessel na *Shell* Orion

Finalizada a implementação do algoritmo Gustafson-Kessel na *Shell* Orion, foram efetuados alguns testes a fim de verificar os resultados gerados e os tempos de processamento da aplicação desenvolvida.

8.3 RESULTADOS OBTIDOS

Os resultados foram obtidos por meio da tarefa de clusterização pelo algoritmo de lógica *fuzzy* Gustafson-Kessel na *Shell* Orion *Data Mining Engine*, considerando o funcionamento do módulo desenvolvido, análise dos grupos gerados, desempenho do algoritmo, integridade do método e usabilidade da aplicação.

Na realização dos testes do algoritmo Gustafson-Kessel para a clusterização dos dados, utilizou-se um microcomputador com sistema operacional Windows Vista Home Premium, processador Intel Centrino Core 2 Duo 1.5 GHz e 2 GB de memória RAM.

Inicialmente, verificou-se as capacidades do algoritmo em relação a descoberta de grupos na base de dados de pacientes com sepse, com o objetivo de verificar os resultados obtidos.

8.3.1 *Clusters* Gerados pelo Algoritmo Gustafson-Kessel

O principal conhecimento gerado pelos algoritmos de clusterização são os grupos encontrados na base de dados. A fim de verificar as capacidades do algoritmo Gustafson-Kessel e exemplificar sua utilização, o processo de clusterização foi executado na base de dados de pacientes com sepse, com os seguintes parâmetros de entrada:

a) **quantidade de *clusters***: 2;

b) **parâmetro de *fuzzyficação***: 2;

c) **quantidade de iterações**: 0 (ilimitado);

d) **taxa de erro**: 0.00001;

e) **atributos de entrada**: *idade*, *sexo*, *gravidad* e *tempodesfecho*.

O objetivo deste teste foi determinar o desfecho dos pacientes, de acordo com os atributos de entrada *idade*, *sexo*, *gravidad* e *tempodesfecho*. Considerando isto, o atributo de saída selecionado foi *desfecho*. Os resultados apresentados pelo algoritmo estão descritos na Tabela 8.

Tabela 8. *Clusters* distintos identificados pelo algoritmo Gustafson-Kessel

Cluster	**Quantidade de elementos**	**Porcentagem dos elementos**	**Desfecho**
1	45	46.88%	2
2	51	53.12%	1

A Figura 47 demonstra o gráfico gerado pela aplicação, onde pode-se perceber que existe uma distância entre os dois grupos, o que facilita a identificação das características que diferenciam os elementos.

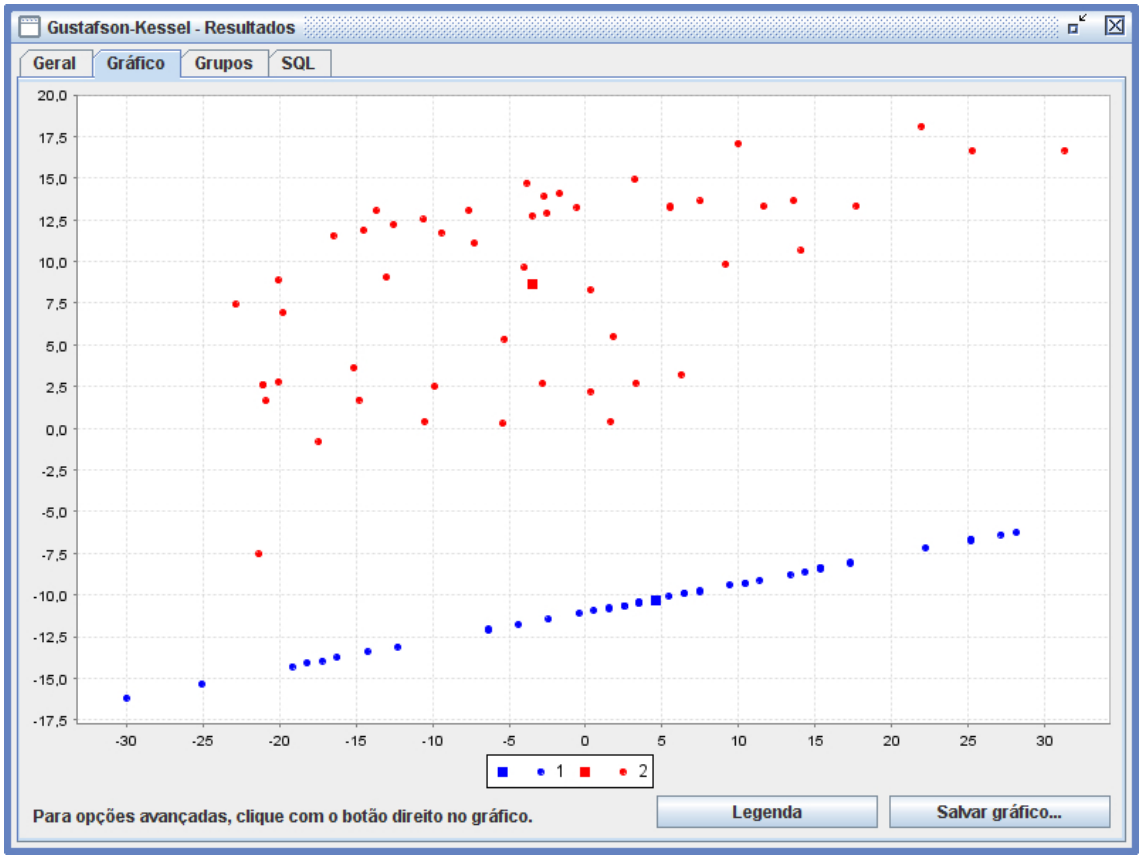

Figura 47. Gráfico contendo dois grupos distintos

Os resultados demonstram que o algoritmo obteve êxito em sua clusterização, pois os dois *clusters* foram encontrados com precisão, onde o primeiro (azul) agrupa os pacientes curados e o segundo (vermelho) os pacientes que não se curaram.

Em outro teste realizado com os mesmos parâmetros, porém com atributos de entrada diferentes (*ira*, *vm*, *vasopres*, *apacheII*, *tba1* e *carbonyl*), tentou-se determinar o *desfecho* do paciente. No entanto, os resultados (Tabela 9) não foram precisos como os obtidos anteriormente, pelo fato dos atributos selecionados serem muito variáveis, pois se referem a exames feitos pelos pacientes.

Tabela 9. *Clusters* encontrados pelo algoritmo Gustafson-Kessel

Cluster	Quantidade de elementos	Porcentagem dos elementos	Desfecho
1	20	20.83%	2
2	51	53.13%	1
	25	26.04%	2

A Figura 48 demonstra os dois grupos encontrados e pode-se perceber que os dados estão espalhados pelo gráfico, o que dificulta a identificação dos *clusters*. O primeiro grupo (azul) ocupa o espaço inferior esquerdo, com alguns elementos um pouco distantes e o segundo (vermelho) encontra-se fragmentado pelo gráfico.

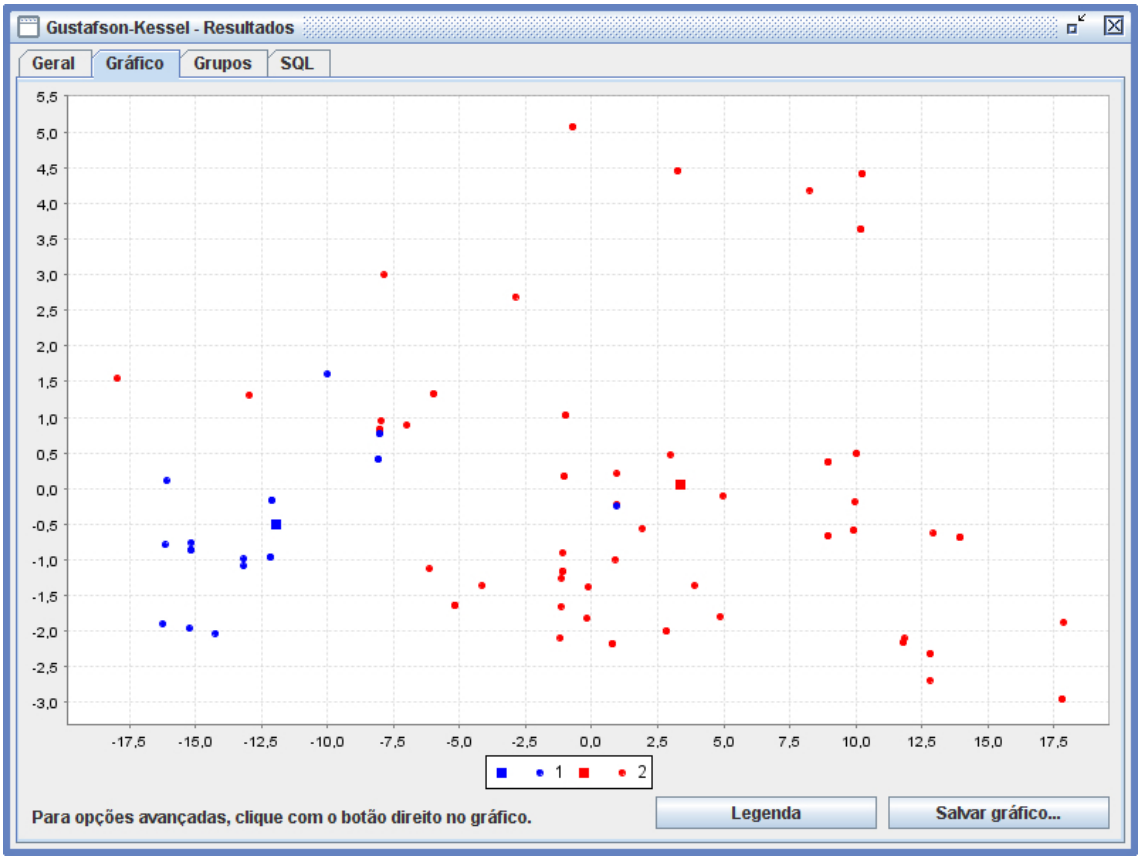

Figura 48. Gráfico contendo grupos de difícil identificação

Considerando isto, chegou-se a conclusão que alguns atributos de entrada podem influenciar diretamente nos resultados do algoritmo e deve-se determinar estes atributos com exatidão, conforme os objetivos de descoberta de conhecimento envolvidos.

Concluindo o funcionamento correto do algoritmo, realizou-se testes com diferentes parâmetros de entrada a fim de verificar os tempos de processamento utilizados pelo módulo desenvolvido.

8.3.2 Tempos de processamento do Algoritmo Gustafson-Kessel

Os testes de desempenho foram feitos por meio de uma base de dados com valores aleatórios, contendo 5000 registros e 5 atributos. A avaliação dos tempos de processamento durante a clusterização dos elementos foi realizada utilizando-se diferentes valores nos parâmetros: quantidade de *clusters*; número de iterações; quantidade de atributos de entrada. A taxa de erro não foi determinada nestes testes, pois o objetivo era avaliar o tempo de execução de um número fixo de iterações. A Tabela 10 demonstra os tempos de execução de acordo com os parâmetros de entrada selecionados.

Tabela 10. Resultados de testes com *fuzzyficação* padrão

Quantidade de *clusters*	**Parâmetro de *fuzzyficação***	**Número de iterações**	**Atributos de entrada**	**Tempo de processamento**
5	2	100	2	00m:03s.747ms
5	2	100	3	00m:04s.859ms
5	2	100	4	00m:06s.822ms
5	2	100	5	00m:09s.340ms
6	2	100	2	00m:04s.280ms
6	2	100	3	00m:06s.169ms
6	2	100	4	00m:08s.619ms
6	2	100	5	00m:11s.345ms
8	2	100	2	00m:06s.646ms
8	2	100	3	00m:08s.961ms
8	2	100	4	00m:12s.108ms
8	2	100	5	00m:16s.459ms
10	2	100	2	00m:09s.298ms
10	2	100	3	00m:12s.277ms
10	2	100	4	00m:16s.523ms
10	2	100	5	00m:21s.432ms

Considerando o valor padrão 2 para o parâmetro de *fuzzyficação*, pode-se concluir que a quantidade de *clusters* e o número de atributos selecionados interferem diretamente no tempo de processamento do algoritmo, principalmente no caso da base utilizada possuir muitos dados e/ou registros.

Os resultados do tempo de processamento do algoritmo Gustafson-Kessel foram satisfatórios, visto que utilizou-se um número fixo de iterações. Por meio do parâmetro taxa de erro, os tempos são melhores (Tabela 11), considerando as situações em que o algoritmo executou menos ciclos para identificar os grupos.

Tabela 11. Resultados de testes utilizando o parâmetro taxa de erro

Quantidade de *clusters*	**Parâmetro de *fuzzyficação***	**Número de iterações**	**Taxa de erro**	**Atributos de entrada**	**Tempo de processamento**
5	2	18	0.00001	2	00m:00s.804ms
5	2	33	0.00001	3	00m:01s.716ms
5	2	100	0.00001	4	00m:06s.885ms
5	2	55	0.00001	5	00m:05s.213ms
6	2	59	0.00001	2	00m:02s.691ms
6	2	85	0.00001	3	00m:05s.283ms
6	2	100	0.00001	4	00m:08s.547ms
6	2	54	0.00001	5	00m:06s.301ms
8	2	76	0.00001	2	00m:05s.270ms
8	2	100	0.00001	3	00m:08s.912ms
8	2	100	0.00001	4	00m:12s.117ms
8	2	56	0.00001	5	00m:09s.183ms
10	2	71	0.00001	2	00m:06s.650ms
10	2	100	0.00001	3	00m:12s.119ms
10	2	100	0.00001	4	00m:16s.213ms
10	2	56	0.00001	5	00m:12s.078ms

Outros valores de *fuzzyficação* foram testados (1.5 e 2.5) e os grupos encontrados foram os mesmos do parâmetro padrão, porém os tempos de processamento foram muito superiores aos obtidos com o valor anterior (Tabela 12).

Tabela 12. Resultados de testes com outros valores de *fuzzyficação*

Quantidade de *clusters*	**Parâmetro de *fuzzyficação***	**Número de iterações**	**Atributos de entrada**	**Tempo de processamento**
5	1.5	100	2	00m:13s.078ms
5	2.5	100	2	00m:18s.440ms
5	1.5	100	3	00m:24s.841ms
5	2.5	100	3	00m:28s.316ms
5	1.5	100	4	00m:37s.251ms
5	2.5	100	4	00m:40s.948ms
5	1.5	100	5	00m:51s.373ms
5	2.5	100	5	00m:56s.673ms
10	1.5	100	5	01m:46s.536ms
10	2.5	100	5	02m:09s.236ms

Ao analisar os resultados de tempo de processamento, pode-se considerar as seguintes observações em relação a alguns parâmetros de entrada:

a) **quantidade de *clusters***: determinar com precisão o número de grupos que se deseja encontrar, visto que além de reduzir o período de execução, os resultados obtidos são superiores;

b) **parâmetro de *fuzzyficação***: utilizar o valor padrão 2, pois além de ter um tempo de processamento menor, os resultados gerados pelo algoritmo são similares a outros valores utilizados para este parâmetro;

c) **taxa de erro**: considerar a condição de parada pela taxa de erro, visto que ao atingir o valor determinado, os resultados encontrados são satisfatórios e não existe necessidade de executar outras iterações;

d) **número de atributos de entrada**: inserir apenas os elementos relevantes à determinação dos grupos, pois quanto mais atributos selecionados, maior será o tempo de processamento.

Após a verificação do desempenho do módulo desenvolvido, foi realizada uma comparação com a ferramenta *Clustering Toolbox*, que também possui o algoritmo Gustafson-Kessel implementado, a fim de conferir os valores obtidos.

8.3.3 Comparação com outra Aplicação

A fim de comparar os resultados do módulo desenvolvido, foi utilizado uma ferramenta chamada *Clustering Toolbox*[34], implementada para Matlab[35]. Esta extensão foi desenvolvida por Janos Abonyi, Balazs Balasko e Balazs Feil do Departamento de Engenharia de Processos da Universidade de Veszprém, na Hungria.

A base utilizada nos testes foi a de dados médicos de pacientes com sepse. Os seguintes parâmetros de entrada foram determinados:

a) **quantidade de *clusters***: 3;

b) **parâmetro de *fuzzyficação***: 2;

c) **quantidade de iterações**: 0 (ilimitado);

d) **taxa de erro**: 0.00001;

e) **atributos de entrada**: *gravidad*, *tba1* e *carbonyl*.

Os resultados (Tabela 13) demonstram que as duas ferramentas desenvolveram grupos idênticos, concluindo que ambas implementam corretamente o algoritmo Gustafson-Kessel.

---

[34] *Clustering Toolbox* disponível em (http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=7486).

[35] O Matlab é uma aplicação comercial que implementa diversas operações matemáticas. Nos testes realizados, foi utilizada a versão de demonstração, disponível em (http://www.mathworks.com).

Tabela 13. Resultados de testes de comparação

Ferramenta	Iterações	*Cluster* 1	*Cluster* 2	*Cluster* 3
Shell Orion	49	21 elementos (21,88%)	33 elementos (34,38%)	42 elementos (43,75%)
Clustering Toolbox	66	42 elementos (43,75%)	21 elementos (21,88%)	33 elementos (34,38%)

Neste teste notou-se que o módulo implementado na *Shell* Orion se demonstrou mais simples no método de inserção de dados, por possuir telas de definição de parâmetros e demonstração dos resultados (Figura 49), diferentemente da ferramenta *Clustering Toolbox*, que deve ser toda executada por meio de comandos no Matlab (Figura 50).

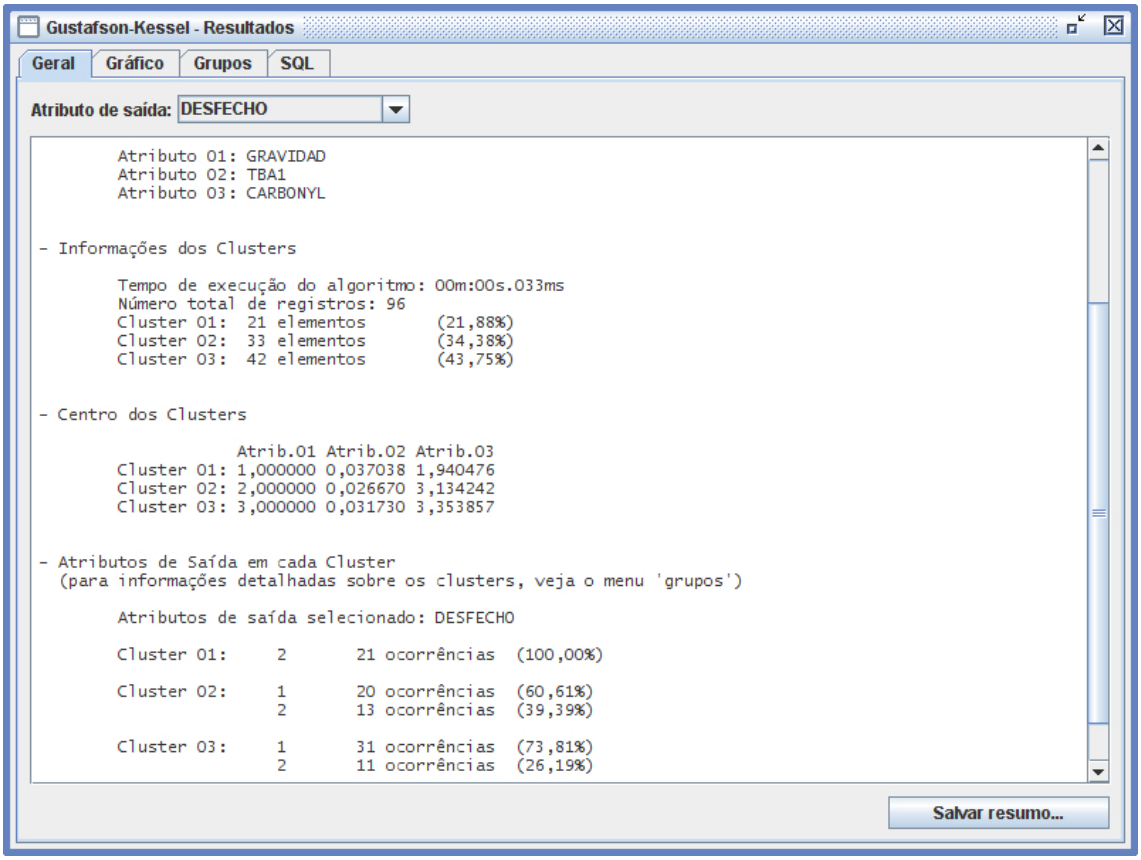

Figura 49. Resultados de testes na *Shell* Orion

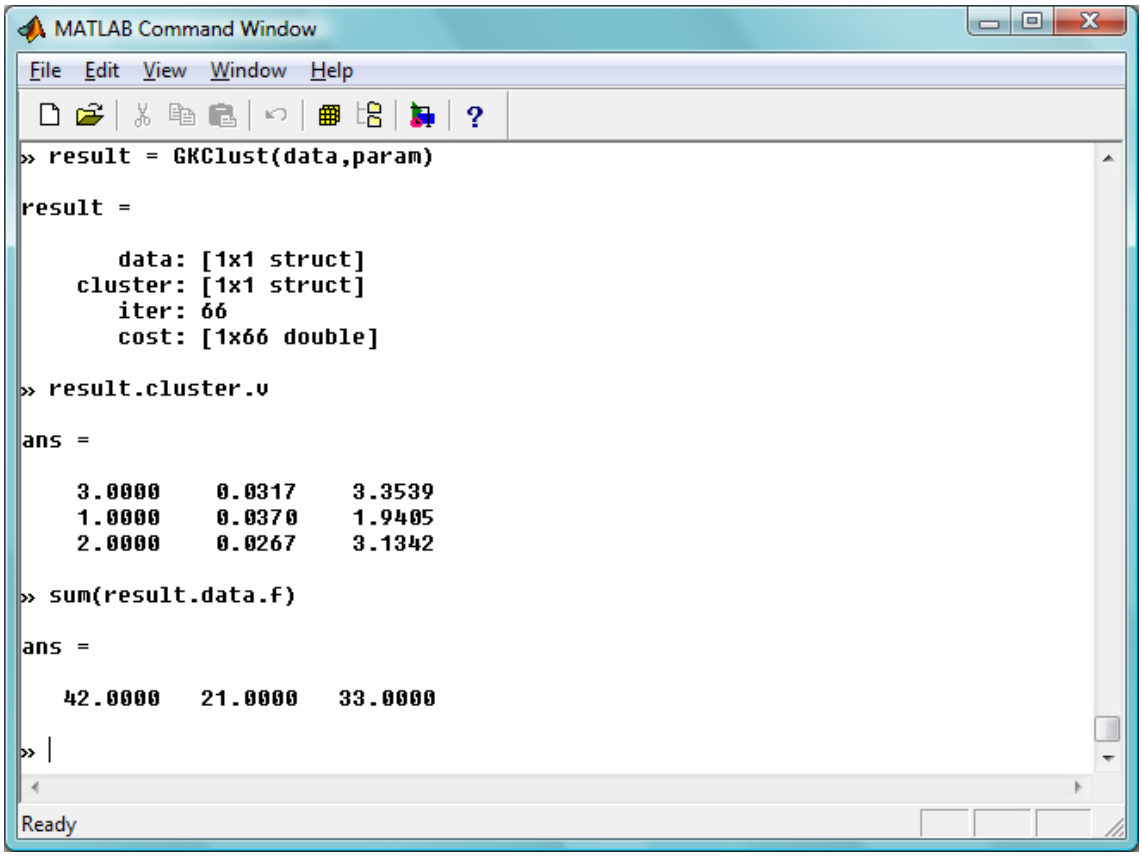

Figura 50. Resultados de testes na ferramenta *Clustering Toolbox*

Os comandos executados acima na ferramenta *Clustering Toolbox* se referem a: realizar a clusterização (*result = GKClust(data,param)*), onde *data* são os dados e *param* os parâmetros de entrada; verificar os centros dos grupos obtidos (*result.cluster.v*); somar os elementos de cada *cluster* (*sum(result.data.f)*).

Pode-se verificar que os centros calculados pelas duas ferramentas são idênticos, além de encontrarem a mesma quantidade de elementos em cada grupo. No entanto, os resultados obtidos pelo módulo na *Shell* Orion encontram-se analisados e descritos em um resumo, enquanto na *Clustering Toolbox* estes precisam ser compreendidos a fim de identificar os grupos e seus respectivos dados.

Considerando todos os testes realizados, pode-se concluir que os resultados obtidos foram satisfatórios, além de confirmar o funcionamento de todas as opções disponíveis no módulo implementado.

CONCLUSÃO

A complexa tarefa de descoberta de novos conhecimentos em bases de dados pode ser simplificada utilizando-se os conceitos de *data mining*. Considerando suas diversas tarefas e métodos, esta técnica possibilita as organizações vantagem competitiva no que se refere a exploração de diferentes informações e no auxílio a tomada de decisão.

Entre as tarefas existentes, esta pesquisa fundamentou-se no entendimento da clusterização, especificamente sobre o algoritmo Gustafson-Kessel, o qual possibilita a identificação de grupos de diferentes formas e dimensões, por implementar a teoria da lógica *fuzzy*, que permite aos elementos pertencerem a diversos grupos simultaneamente, aumentando a precisão no particionamento dos dados.

Durante a pesquisa, algumas dificuldades foram encontradas, especialmente com relação ao entendimento do método de lógica *fuzzy* para clusterização e ao funcionamento do algoritmo Gustafson-Kessel, os quais possuíam carência de bibliografia que auxiliasse no estudo da teoria de clusterização *fuzzy* e na construção do modelo matemático do algoritmo.

No entanto, estas dificuldades foram superadas e os objetivos desta pesquisa em relação a compreensão do método de lógica *fuzzy* para a tarefa de clusterização, a modelagem matemática do algoritmo Gustafson-Kessel e sua implementação no módulo de clusterização da *Shell* Orion foram alcançados.

Após a compreensão do método e a implementação do algoritmo Gustafson-Kessel na *Shell* Orion, foram executados alguns testes os quais demonstraram seu funcionamento correto, particionando os grupos corretamente devido a utilização das matrizes de covariância *fuzzy*, as quais determinam as dimensões de cada *cluster*.

Concluiu-se também que o tempo de processamento foi satisfatório, mesmo quando executado em bases de dados com muitos elementos. No entanto, sua performance depende exclusivamente dos parâmetros de entrada, onde estes devem ser determinados com precisão.

Considerando a pesquisa realizada, são descritas algumas sugestões de trabalhos futuros com o objetivo de dar continuidade ao projeto da *Shell* Orion *Data Mining Engine*:

a) implementar novos algoritmos de clusterização pelo método de lógica *fuzzy*, como por exemplo o *Fuzzy C-Means*;
b) desenvolver outras etapas relacionadas a descoberta de conhecimento em bases de dados, como por exemplo, finalidades de pré e pós processamento, a fim de possibilitar respectivamente a preparação dos dados e a análise dos resultados;
c) implementar algoritmos referentes a outros métodos de clusterização, como por exemplo, baseados em densidade;
d) desempenhar uma pesquisa que utilize os resultados obtidos na clusterização pelo algoritmo de lógica *fuzzy* Gustafson-Kessel em outras tarefas de *data mining*, como por exemplo a classificação.

REFERÊNCIAS

AKKIZIDIS, Ioannis S.; ROBERTS, Geoff N. **Fuzzy clustering methods for identifying and modelling of non-linear control strategies**. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, v. 215, n. 5, p. 437-452, 2001.

BÄCK, Thomas; FOGEL, David B.; MICHALEWICZ, Zbigniew. **Handbook of Evolutionary Computation**. New York: Taylor & Francis Group, 1997.

BERRY, Michael J.; LINOFF, Gordon. **Data mining techniques**: for marketing, sales, and customer relationship management. Indianapolis: Wiley Publishing, 2004.

BEZDEK, James et al. **Fuzzy models and algorithms for pattern recognition and image processing**. New York: Springer, 2005.

BORTOLOTTO, Leandro S. **O Método de Redes Neurais pelo Algoritmo Kohonen para Clusterização na Shell Orion Data Mining Engine**. 2007. 91 f. Trabalho de Conclusão de Curso (Especialização) – Faculdade de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2007.

CALDEIRAS, André M. et al. **Inteligência computacional**: aplicada à administração, economia e engenharia em Matlab. São Paulo: Thomson Learning, 2007.

CASAGRANDE, Diego P. **O Módulo da Tarefa de Associação pelo Algoritmo Apriori no Desenvolvimento Da Shell de Data Mining Orion**. 2005. 79 f. Trabalho de Conclusão de Curso (Especialização) – Faculdade de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2005.

CECIL, Russell L.; GOLDMAN, Lee; AUSIELLO, Dennis A. **Cecil**: tratado de medicina interna. Rio de Janeiro: Elsevier, 2005.

CHEN, Zhengxin. **Data mining and uncertain reasoning**: an integrated approach. New York: Wiley Publishing, 2001.

COELHO, Leandro dos S.; SILVA, Wesley V. da; PROTIL, Roberto M. Previsão não-linear dos preços de troncos de eucalipto baseada em uma abordagem neuroevolutiva. **Gestão e Produção**, São Carlos, v. 14, n. 1, p. 139-154, abr 2007. Disponível em: <http://www.scielo.br/pdf/gp/v14n1/11.pdf>. Acesso em: 12 set 2007.

COX, Earl. **Fuzzy modeling and genetic algorithms for data mining and exploration**. California: Morgan Kaufmann, 2005.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. From Data mining to Knowledge Discovery in Databases. **AI Magazine**, v. 17, n. 3, p. 37-54, 1996. Disponível em: <http://kdnuggets.com/gpspubs/aimag-kdd-overview-1996-Fayyad.pdf>. Acesso em: 12 set 2007.

GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel L. **Data Mining**: um guia prático. Rio de Janeiro: Elsevier, 2005.

GUSTAFSON, Donald E.; KESSEL, William C. **Fuzzy clustering with fuzzy covariance matrix**. Proceedings of the IEEE Control and Decision Conference, San Diego, p. 761-766, jan 1979.

HAN, Jiawei; KAMBER, Micheline. **Data mining**: concepts and techniques. San Francisco: Morgan Kaufmann, 2001.

HAND, David; MANNILA, Heikki; SMYTH, Padhraic. **Principles of Data Mining**. London: The MIT Press, 2001.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. **The elements of statistical learning**: data mining, inference, and prediction. New York: Springer, 2001.

HÖPPNER, Frank et al. **Fuzzy cluster analysis**: methods for classification, data analysis, and image recognition. Chichester: John Wiley & Sons, 1999.

KANTARDZIC, Mehmed. **Data Mining**: Concepts, Models, Methods, and Algorithms. John Wiley & Sons, 2003.

KIM, Dae-Won; LEE, Kwang H.; LEE, Doheon. Detecting clusters of different geometrical shapes in microarray gene expression data. **Bioinformatics**, Oxford, v. 21, n. 9, p. 1927-1934, maio 2005. Disponível em: <http://bioinformatics.oxfordjournals.org/cgi/reprint/bti251v1.pdf>. Acesso em: 21 maio 2007.

KLIR, George J.; YUAN, Bo. **Fuzzy sets and fuzzy logic**: theory and applications. New Jersey: Prentice Hall, 1995.

LUGER, George F. **Inteligência artificial**: estruturas e estratégias para a solução de problemas complexos. Porto Alegre: Bookman, 2004.

MARTINS, Denis P. **A Tarefa de Clusterização pelo Método de Particionamento K-means na Shell Orion Data Mining Engine**. 2007. 77 f. Trabalho de Conclusão de Curso (Especialização) – Faculdade de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2007.

OSOWSKI, Stanislaw; LINH, Tran H. **ECG Beat Recognition Using Fuzzy Hybrid Neural Network**. IEEE Transactions on Biomedical Engineering, v. 48, n. 11, p. 1265-1271, nov 2001.

PAL, Sankar K.; MITRA, Pabitra. **Pattern recognition algorithms for data mining**: scalability, knowledge discovery and soft granular computing. Florida: Chapman & Hall, 2004.

PELEGRIN, Diana C. **A Tarefa de Classificação e o Algoritmo ID3 para Indução de Árvores de Decisão na Shell de Data Mining Orion**. 2005. 92 f. Trabalho de Conclusão de Curso (Especialização) – Faculdade de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2005.

PENDER, Tom. **UML**: a bíblia. Rio de Janeiro: Elsevier, 2004.

POOLE, David; MONTEIRO, Martha S. **Álgebra linear**. São Paulo: Thomson Learning, 2004.

RAIMUNDO, Lidiane R. **O Algoritmo CART pelo Critério de Gini na Tarefa de Classificação da Shell Orion Data Mining Engine**. 2007. 106 f. Trabalho de Conclusão de Curso (Especialização) – Faculdade de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2007.

RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência artificial**. Rio de Janeiro: Elsevier, 2004.

SERRA, Laércio. **A essência do Business Intelligence**. São Paulo: Berkeley Brasil, 2002.

STEWART, James. **Cálculo**. São Paulo: Thomson Learning, 2006.

SUHAIMI, Azrina B. **An Analysis of Fuzzy Clustering Algortihms for Suggestion of Supervisor and Examiner of Thesis Title**. 2005. 163 f. Dissertação de Mestrado – Curso de Ciência da Computação e Sistemas de Informação, Universidade de Tecnologia da Malásia, Skudai, Johor, Malásia. Disponível em: <http://eprints.utm.my/2709/1/AzrinaSuhaimiMCD2005TTT.pdf>. Acesso em: 5 set 2007.

WANG, Li-Xin. **A course in fuzzy systems and control**. London: Prentice Hall, 1997.

WITTEN, Ian H.; FRANK, Eibe. **Data mining**: practical machine learning tools and techniques. San Francisco: Morgan Kaufmann, 2005.

ZADEH, Lotfi A. **Fuzzy Sets**. Information and Control, vol. 8, p. 338-353, 1965.

APÊNDICE A – MODELAGEM MATEMÁTICA COMPLETA

Neste apêndice é apresentada a modelagem matemática completa do algoritmo Gustafson-Kessel, a fim de demonstrar todo o processo envolvido no cálculo de uma iteração do método.

1. Parâmetros de entrada

a) número de *clusters*: 2

b) parâmetro de *fuzzyficação* (*m*): 2

c) taxa de erro (ε): 10^{-05} (0.00001)

d) base de dados (atributos x dados):

	x_1	x_2	x_3	x_4
a	0	1	1	0
b	0.6	0.4	0.9	0.7
c	1.2	1.3	1.3	1.0

e) valores iniciais da matriz de pertinência *U* (*clusters* x dados):

	x_1	x_2	x_3	x_4
c_1	0.67	0.41	0.24	0.5
c_2	0.33	0.59	0.76	0.5

2. Cálculo dos centros:

$$c_i = \frac{\sum_{j=1}^{p} (u_{ij})^m x_j}{\sum_{j=1}^{p} (u_{ij})^m}$$

a) primeiro *cluster*:

$$c_1 = \frac{0.67^2 \cdot \begin{bmatrix} 0 \\ 0.6 \\ 1.2 \end{bmatrix} + 0.41^2 \cdot \begin{bmatrix} 1 \\ 0.4 \\ 1.3 \end{bmatrix} + 0.24^2 \cdot \begin{bmatrix} 1 \\ 0.9 \\ 1.3 \end{bmatrix} + 0.5^2 \cdot \begin{bmatrix} 0 \\ 0.7 \\ 1.0 \end{bmatrix}}{0.67^2 + 0.41^2 + 0.24^2 + 0.5^2}$$

$$c_1 = \frac{\begin{bmatrix} 0 \\ 0.26934 \\ 0.53868 \end{bmatrix} + \begin{bmatrix} 0.1681 \\ 0.06724 \\ 0.21853 \end{bmatrix} + \begin{bmatrix} 0.0576 \\ 0.05184 \\ 0.07488 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.175 \\ 0.25 \end{bmatrix}}{0.4489 + 0.1681 + 0.0576 + 0.25}$$

$$c_1 = \frac{\begin{bmatrix} 0.2257 \\ 0.56342 \\ 1.08209 \end{bmatrix}}{0.9246}$$

$$c_1 = \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix}$$

b) segundo *cluster*:

$$c_2 = \frac{0.33^2 \cdot \begin{bmatrix} 0 \\ 0.6 \\ 1.2 \end{bmatrix} + 0.59^2 \cdot \begin{bmatrix} 1 \\ 0.4 \\ 1.3 \end{bmatrix} + 0.76^2 \cdot \begin{bmatrix} 1 \\ 0.9 \\ 1.3 \end{bmatrix} + 0.5^2 \cdot \begin{bmatrix} 0 \\ 0.7 \\ 1.0 \end{bmatrix}}{0.33^2 + 0.59^2 + 0.76^2 + 0.5^2}$$

$$c_2 = \frac{\begin{bmatrix} 0 \\ 0.06534 \\ 0.13068 \end{bmatrix} + \begin{bmatrix} 0.3481 \\ 0.13924 \\ 0.45253 \end{bmatrix} + \begin{bmatrix} 0.5776 \\ 0.51984 \\ 0.75088 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.175 \\ 0.25 \end{bmatrix}}{0.1089 + 0.3481 + 0.5776 + 0.25}$$

$$c_2 = \frac{\begin{bmatrix} 0.9257 \\ 0.89942 \\ 1.58409 \end{bmatrix}}{1.2846}$$

$$c_2 = \begin{bmatrix} 0.72061 \\ 0.70016 \\ 1.23314 \end{bmatrix}$$

c) centro dos *clusters* encontrados:

	c_1	c_2
a	0.24411	0.72061
b	0.60937	0.70016
c	1.17033	1.23314

3. Cálculo da matriz de covariância *fuzzy*:

$$F_i = \frac{\sum_{j=1}^{p} (u_{ij})^m (x_j - c_i)(x_j - c_i)^{\mathrm{T}}}{\sum_{j=1}^{p} (u_{ij})^m}$$

a) primeiro *cluster*:

$$F_1 = \frac{\begin{array}{l} 0.67^2 \cdot \left(\begin{bmatrix} 0.0 \\ 0.6 \\ 1.2 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right) \cdot \left([0.0, 0.6, 1.2] - [0.24411, 0.60937, 1.17033] \right) + \\ 0.41^2 \cdot \left(\begin{bmatrix} 1.0 \\ 0.4 \\ 1.3 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right) \cdot \left([1.0, 0.4, 1.3] - [0.24411, 0.60937, 1.17033] \right) + \\ 0.24^2 \cdot \left(\begin{bmatrix} 1.0 \\ 0.9 \\ 1.3 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right) \cdot \left([1.0, 0.9, 1.3] - [0.24411, 0.60937, 1.17033] \right) + \\ 0.5^2 \cdot \left(\begin{bmatrix} 0.0 \\ 0.7 \\ 1.0 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right) \cdot \left([0.0, 0.7, 1.0] - [0.24411, 0.60937, 1.17033] \right) \end{array}}{0.67^2 + 0.41^2 + 0.24^2 + 0.5^2}$$

$$F_1 = \frac{\begin{array}{l} 0.4489 \cdot \begin{bmatrix} -0.24411 \\ -0.00937 \\ 0.02967 \end{bmatrix} \cdot [-0.24411, -0.00937, 0.02967] + \\ 0.1681 \cdot \begin{bmatrix} 0.75589 \\ -0.20937 \\ 0.12967 \end{bmatrix} \cdot [0.75589, -0.20937, 0.12967] + \\ 0.0576 \cdot \begin{bmatrix} 0.75589 \\ 0.29063 \\ 0.12967 \end{bmatrix} \cdot [0.75589, 0.29063, 0.12967] + \\ 0.25 \cdot \begin{bmatrix} -0.24411 \\ 0.09063 \\ -0.17033 \end{bmatrix} \cdot [-0.24411, 0.09063, -0.17033] \end{array}}{0.4489 + 0.1681 + 0.0576 + 0.25}$$

$$F_1 = \frac{\begin{bmatrix} -0.10958 \\ -0.0042 \\ 0.01332 \end{bmatrix} \cdot [-0.24411, -0.00937, 0.02967] + \begin{bmatrix} 0.12707 \\ -0.03519 \\ 0.0218 \end{bmatrix} \cdot [0.75589, -0.20937, 0.12967] + \begin{bmatrix} 0.04354 \\ 0.01674 \\ 0.00747 \end{bmatrix} \cdot [0.75589, 0.29063, 0.12967] + \begin{bmatrix} -0.06103 \\ 0.02266 \\ -0.04258 \end{bmatrix} \cdot [-0.24411, 0.09063, -0.17033]}{0.9246}$$

$$F_1 = \frac{\begin{bmatrix} 0.02675 & 0.00103 & -0.00325 \\ 0.00103 & 0.00004 & -0.00012 \\ -0.00325 & -0.00012 & 0.0004 \end{bmatrix} + \begin{bmatrix} 0.09605 & -0.0266 & 0.01648 \\ -0.0266 & 0.00737 & -0.00456 \\ 0.01648 & -0.00456 & 0.00283 \end{bmatrix} + \begin{bmatrix} 0.03291 & 0.01265 & 0.00565 \\ 0.01265 & 0.00487 & 0.00217 \\ 0.00565 & 0.00217 & 0.00097 \end{bmatrix} + \begin{bmatrix} 0.0149 & -0.00553 & 0.01039 \\ -0.00553 & 0.00205 & -0.00386 \\ 0.01039 & -0.00386 & 0.00725 \end{bmatrix}}{0.9246}$$

$$F_1 = \frac{\begin{bmatrix} 0.17061 & -0.01845 & 0.02927 \\ -0.01845 & 0.01433 & -0.00638 \\ 0.02927 & -0.00638 & 0.01144 \end{bmatrix}}{0.9246}$$

$$F_1 = \begin{bmatrix} 0.18452 & -0.01996 & 0.03165 \\ -0.01996 & 0.0155 & -0.0069 \\ 0.03165 & -0.0069 & 0.01238 \end{bmatrix}$$

b) segundo *cluster*:

$$F_2 = \frac{\begin{array}{l} 0.33^2 \cdot \left(\begin{bmatrix} 0.0 \\ 0.6 \\ 1.2 \end{bmatrix} - \begin{bmatrix} 0.72061 \\ 0.70016 \\ 1.23314 \end{bmatrix} \right) \cdot \left([0.0, 0.6, 1.2] - [0.72061, 0.70016, 1.23314] \right) + \\ 0.59^2 \cdot \left(\begin{bmatrix} 1.0 \\ 0.4 \\ 1.3 \end{bmatrix} - \begin{bmatrix} 0.72061 \\ 0.70016 \\ 1.23314 \end{bmatrix} \right) \cdot \left([1.0, 0.4, 1.3] - [0.72061, 0.70016, 1.23314] \right) + \\ 0.76^2 \cdot \left(\begin{bmatrix} 1.0 \\ 0.9 \\ 1.3 \end{bmatrix} - \begin{bmatrix} 0.72061 \\ 0.70016 \\ 1.23314 \end{bmatrix} \right) \cdot \left([1.0, 0.9, 1.3] - [0.72061, 0.70016, 1.23314] \right) + \\ 0.5^2 \cdot \left(\begin{bmatrix} 0.0 \\ 0.7 \\ 1.0 \end{bmatrix} - \begin{bmatrix} 0.72061 \\ 0.70016 \\ 1.23314 \end{bmatrix} \right) \cdot \left([0.0, 0.7, 1.0] - [0.72061, 0.70016, 1.23314] \right) \end{array}}{0.33^2 + 0.59^2 + 0.76^2 + 0.5^2}$$

$$F_2 = \frac{\begin{array}{l} 0.1089 \cdot \begin{bmatrix} -0.72061 \\ -0.10016 \\ -0.03314 \end{bmatrix} \cdot [-0.72061, -0.10016, -0.03314] + \\ 0.3481 \cdot \begin{bmatrix} 0.27939 \\ -0.30016 \\ 0.06686 \end{bmatrix} \cdot [0.27939, -0.30016, 0.06686] + \\ 0.5776 \cdot \begin{bmatrix} 0.27939 \\ 0.19984 \\ 0.06686 \end{bmatrix} \cdot [0.27939, 0.19984, 0.06686] + \\ 0.25 \cdot \begin{bmatrix} -0.72061 \\ -0.00016 \\ -0.23314 \end{bmatrix} \cdot [-0.72061, -0.00016, -0.23314] \end{array}}{0.1089 + 0.3481 + 0.5776 + 0.25}$$

$$F_2 = \frac{\begin{bmatrix} -0.07847 \\ -0.01091 \\ -0.00361 \end{bmatrix} \cdot [-0.72061, -0.10016, -0.03314] + \begin{bmatrix} 0.09725 \\ -0.10448 \\ 0.02327 \end{bmatrix} \cdot [0.27939, -0.30016, 0.06686] + \begin{bmatrix} 0.16137 \\ 0.11543 \\ 0.03862 \end{bmatrix} \cdot [0.27939, 0.19984, 0.06686] + \begin{bmatrix} -0.18015 \\ -0.00004 \\ -0.05828 \end{bmatrix} \cdot [-0.72061, -0.00016, -0.23314]}{1.2846}$$

$$F_2 = \frac{\begin{bmatrix} 0.05655 & 0.00786 & 0.0026 \\ 0.00786 & 0.00109 & 0.00036 \\ 0.0026 & 0.00036 & 0.00012 \end{bmatrix} + \begin{bmatrix} 0.02717 & -0.02919 & 0.0065 \\ -0.02919 & 0.03136 & -0.00699 \\ 0.0065 & -0.00699 & 0.00156 \end{bmatrix} + \begin{bmatrix} 0.04509 & 0.03225 & 0.01079 \\ 0.03225 & 0.02307 & 0.00772 \\ 0.01079 & 0.00772 & 0.00258 \end{bmatrix} + \begin{bmatrix} 0.12982 & 0.00003 & 0.042 \\ 0.00003 & 0 & 0.00001 \\ 0.042 & 0.00001 & 0.01359 \end{bmatrix}}{1.2846}$$

$$F_2 = \frac{\begin{bmatrix} 0.25863 & 0.01095 & 0.06189 \\ 0.01095 & 0.05552 & 0.0011 \\ 0.06189 & 0.0011 & 0.01785 \end{bmatrix}}{1.2846}$$

$$F_2 = \begin{bmatrix} 0.20133 & 0.00852 & 0.04818 \\ 0.00852 & 0.04322 & 0.00086 \\ 0.04818 & 0.00086 & 0.01389 \end{bmatrix}$$

c) matrizes de covariância *fuzzy* encontradas:

$$F_1 = \begin{bmatrix} 0.18452 & -0.01996 & 0.03165 \\ -0.01996 & 0.0155 & -0.0069 \\ 0.03165 & -0.0069 & 0.01238 \end{bmatrix} \quad F_2 = \begin{bmatrix} 0.20133 & 0.00852 & 0.04818 \\ 0.00852 & 0.04322 & 0.00086 \\ 0.04818 & 0.00086 & 0.01389 \end{bmatrix}$$

4. Cálculo das distâncias:

$$d^2(x_j, c_i) = (x_j - c_i)^{\mathrm{T}} M_i (x_j - c_i)$$

onde

$$M_i = \sqrt[N]{\det(F_i)}\, F_i^{-1}$$

a) primeiro *cluster*:

- determinante:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$\det(A) = (a \cdot e \cdot i) + (b \cdot f \cdot g) + (c \cdot d \cdot h) - (c \cdot e \cdot g) - (b \cdot d \cdot i) - (a \cdot f \cdot h)$$

$$F_1 = \begin{bmatrix} 0.18452 & -0.01996 & 0.03165 \\ -0.01996 & 0.0155 & -0.0069 \\ 0.03165 & -0.0069 & 0.01238 \end{bmatrix}$$

$$\begin{aligned} \det(F_1) = {} & (0.18452 \cdot 0.0155 \cdot 0.01238) + (-0.01996 \cdot -0.0069 \cdot 0.03165) \\ & + (0.03165 \cdot -0.01996 \cdot -0.0069) - (0.03165 \cdot 0.0155 \cdot 0.03165) \\ & - (-0.01996 \cdot -0.01996 \cdot 0.01238) - (0.18452 \cdot -0.0069 \cdot -0.0069) \end{aligned}$$

$$\begin{aligned} \det(F_1) = {} & 3.54075428\, e^{-05} + 0.43589646\, e^{-05} + 0.43589646\, e^{-05} \\ & - 1.552669875\, e^{-05} - 0.4932211808\, e^{-05} - 0.87849972\, e^{-05} \end{aligned}$$

$$\det(F_1) = 1.4881564242\, e^{-05}$$

- matriz inversa:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$|A| = \det(A)$$

$$A^{-1} = \frac{1}{|A|} \operatorname{adj}(A)$$

$$\text{adj}(A)=\begin{bmatrix} \begin{vmatrix} e & f \\ h & i \end{vmatrix} & \begin{vmatrix} c & b \\ i & h \end{vmatrix} & \begin{vmatrix} b & c \\ e & f \end{vmatrix} \\ \begin{vmatrix} f & d \\ i & g \end{vmatrix} & \begin{vmatrix} a & c \\ g & i \end{vmatrix} & \begin{vmatrix} c & a \\ f & d \end{vmatrix} \\ \begin{vmatrix} d & e \\ g & h \end{vmatrix} & \begin{vmatrix} b & a \\ h & g \end{vmatrix} & \begin{vmatrix} a & b \\ d & e \end{vmatrix} \end{bmatrix}$$

$$B=\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$|B|=(a\cdot d)-(b\cdot c)$$

$$\text{adj}(F_1)=\begin{bmatrix} \begin{vmatrix} 0.0155 & -0.0069 \\ -0.0069 & 0.01238 \end{vmatrix} & \begin{vmatrix} 0.03165 & -0.01996 \\ 0.01238 & -0.0069 \end{vmatrix} & \begin{vmatrix} -0.01996 & 0.03165 \\ 0.0155 & -0.0069 \end{vmatrix} \\ \begin{vmatrix} -0.0069 & -0.01996 \\ 0.01238 & 0.03165 \end{vmatrix} & \begin{vmatrix} 0.18452 & 0.03165 \\ 0.03165 & 0.01238 \end{vmatrix} & \begin{vmatrix} 0.03165 & 0.18452 \\ -0.0069 & -0.01996 \end{vmatrix} \\ \begin{vmatrix} -0.01996 & 0.0155 \\ 0.03165 & -0.0069 \end{vmatrix} & \begin{vmatrix} -0.01996 & 0.18452 \\ -0.0069 & 0.03165 \end{vmatrix} & \begin{vmatrix} 0.18452 & -0.01996 \\ -0.01996 & 0.0155 \end{vmatrix} \end{bmatrix}$$

$$\text{adj}(F_1)=$$

$$\begin{bmatrix} (0.00019189)-(0.0004761) & (-0.000218385)-(-0.0002471048) & (0.000137724)-(0.000490575) \\ (-0.000218385)-(-0.0002471048) & (0.0022843576)-(0.0010017255) & (-0.000631734)-(-0.001273188) \\ (0.000137724)-(0.000490575) & (-0.000631734)-(-0.001273188) & (0.00286006)-(0.0003984016) \end{bmatrix}$$

$$\text{adj}(F_1)=\begin{bmatrix} 0.00014428 & 0.0000287198 & -0.000352851 \\ 0.0000287198 & 0.0012826351 & 0.000641454 \\ -0.000352851 & 0.000641454 & 0.0024616584 \end{bmatrix}$$

$$F_1^{-1}=\frac{\begin{bmatrix} 0.00014428 & 0.0000287198 & -0.000352851 \\ 0.0000287198 & 0.0048383355 & 0.000641454 \\ -0.000352851 & 0.000641454 & 0.0024616584 \end{bmatrix}}{1.4881564242\,e^{-05}}$$

$$F_1^{-1}=\begin{bmatrix} 9.69522 & 1.92989 & -23.71061 \\ 1.92989 & 86.18953 & 43.10394 \\ -23.71061 & 43.10394 & 165.41664 \end{bmatrix}$$

- matriz covariância modificada:

$$M_1=\sqrt[3]{1.4881564242\,e^{-05}}\cdot\begin{bmatrix} 9.69522 & 1.92989 & -23.71061 \\ 1.92989 & 86.18953 & 43.10394 \\ -23.71061 & 43.10394 & 165.41664 \end{bmatrix}$$

$$M_1 = 0.024597040 \cdot \begin{bmatrix} 9.69522 & 1.92989 & -23.71061 \\ 1.92989 & 86.18953 & 43.10394 \\ -23.71061 & 43.10394 & 165.41664 \end{bmatrix}$$

$$M_1 = \begin{bmatrix} 0.23847 & 0.04747 & -0.58321 \\ 0.04747 & 2.12001 & 1.06023 \\ -0.58321 & 1.06023 & 4.06876 \end{bmatrix}$$

- distâncias:

$$d^2(x_1, c_1) = \left([0.0, 0.6, 1.2] - [0.24411, 0.60937, 1.17033] \right) \cdot M_1 \cdot \left(\begin{bmatrix} 0.0 \\ 0.6 \\ 1.2 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right)$$

$$d^2(x_1, c_1) = [-0.24411, -0.00937, 0.02967] \cdot M_1 \cdot \begin{bmatrix} -0.24411 \\ -0.00937 \\ 0.02967 \end{bmatrix}$$

$$d^2(x_1, c_1) = [-0.24411, -0.00937, 0.02967] \cdot \begin{bmatrix} -0.07596 \\ 0 \\ 0.25315 \end{bmatrix}$$

$$d^2(x_1, c_1) = 0.02605$$

$$d^2(x_2, c_1) = \left([1.0, 0.4, 1.3] - [0.24411, 0.60937, 1.17033] \right) \cdot M_1 \cdot \left(\begin{bmatrix} 1.0 \\ 0.4 \\ 1.3 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right)$$

$$d^2(x_2, c_1) = [0.75589, -0.20937, 0.12967] \cdot M_1 \cdot \begin{bmatrix} 0.75589 \\ -0.20937 \\ 0.12967 \end{bmatrix}$$

$$d^2(x_2, c_1) = [0.75589, -0.20937, 0.12967] \cdot \begin{bmatrix} 0.09469 \\ -0.2705 \\ -0.13523 \end{bmatrix}$$

$$d^2(x_2, c_1) = 0.11068$$

$$d^2(x_3, c_1) = \left([1.0, 0.9, 1.3] - [0.24411, 0.60937, 1.17033] \right) \cdot M_1 \cdot \left(\begin{bmatrix} 1.0 \\ 0.9 \\ 1.3 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right)$$

$$d^2(x_3, c_1) = [0.75589, 0.29063, 0.12967] \cdot M_1 \cdot \begin{bmatrix} 0.75589 \\ 0.29063 \\ 0.12967 \end{bmatrix}$$

$$d^2(x_3, c_1) = [0.75589, 0.29063, 0.12967] \cdot \begin{bmatrix} 0.11843 \\ 0.7895 \\ 0.39489 \end{bmatrix}$$

$$d^2(x_3, c_1) = 0.37018$$

$$d^2(x_4, c_1) = \left([0.0, 0.7, 1.0] - [0.24411, 0.60937, 1.17033] \right) \cdot M_1 \cdot \left(\begin{bmatrix} 0.0 \\ 0.7 \\ 1.0 \end{bmatrix} - \begin{bmatrix} 0.24411 \\ 0.60937 \\ 1.17033 \end{bmatrix} \right)$$

$$d^2(x_4, c_1) = [-0.24411, 0.09063, -0.17033] \cdot M_1 \cdot \begin{bmatrix} -0.24411 \\ 0.09063 \\ -0.17033 \end{bmatrix}$$

$$d^2(x_4, c_1) = [-0.24411, 0.09063, -0.17033] \cdot \begin{bmatrix} 0.04543 \\ -0.00004 \\ -0.45458 \end{bmatrix}$$

$$d^2(x_4, c_1) = 0.06634$$

b) segundo *cluster*:

- determinante:

$$F_2 = \begin{bmatrix} 0.20133 & 0.00852 & 0.04818 \\ 0.00852 & 0.04322 & 0.00086 \\ 0.04818 & 0.00086 & 0.01389 \end{bmatrix}$$

$$\begin{aligned} \det(F_2) = {} & (0.20133 \cdot 0.04322 \cdot 0.01389) + (0.00852 \cdot 0.00086 \cdot 0.04818) \\ & + (0.04818 \cdot 0.00852 \cdot 0.00086) - (0.04818 \cdot 0.04322 \cdot 0.04818) \\ & - (0.00852 \cdot 0.00852 \cdot 0.01389) - (0.20133 \cdot 0.00086 \cdot 0.00086) \end{aligned}$$

$$\begin{aligned} \det(F_2) = {} & (12.08635933\, e^{-05}) + (0.0353024496\, e^{-05}) + (0.0353024496\, e^{-05}) \\ & - (10.03271219\, e^{-05}) - (0.1008280656\, e^{-05}) - (0.0148903668\, e^{-05}) \end{aligned}$$

$$\det(F_2) = 2.008533577\, e^{-05}$$

- matriz inversa:

$$\text{adj}(F_2) = \begin{bmatrix} \begin{vmatrix} 0.04322 & 0.00086 \\ 0.00086 & 0.01389 \end{vmatrix} & \begin{vmatrix} 0.04818 & 0.00852 \\ 0.01389 & 0.00086 \end{vmatrix} & \begin{vmatrix} 0.00852 & 0.04818 \\ 0.04322 & 0.00086 \end{vmatrix} \\ \begin{vmatrix} 0.00086 & 0.00852 \\ 0.01389 & 0.04818 \end{vmatrix} & \begin{vmatrix} 0.20133 & 0.04818 \\ 0.04818 & 0.01389 \end{vmatrix} & \begin{vmatrix} 0.04818 & 0.20133 \\ 0.00086 & 0.00852 \end{vmatrix} \\ \begin{vmatrix} 0.00852 & 0.04322 \\ 0.04818 & 0.00086 \end{vmatrix} & \begin{vmatrix} 0.00852 & 0.20133 \\ 0.00086 & 0.04818 \end{vmatrix} & \begin{vmatrix} 0.20133 & 0.00852 \\ 0.00852 & 0.04322 \end{vmatrix} \end{bmatrix}$$

$$\text{adj}(F_2)=$$

$$\begin{bmatrix} (0.0006003258)-(0.0000007396) & (0.0000414348)-(0.0001183428) & (0.0000073272)-(0.0020823396) \\ (0.0000414348)-(0.0001183428) & (0.0027964737)-(0.0023213124) & (0.0004104936)-(0.0001731438) \\ (0.0000073272)-(0.0020823396) & (0.0004104936)-(0.0001731438) & (0.0087014826)-(0.0000725904) \end{bmatrix}$$

$$\text{adj}(F_2)=\begin{bmatrix} 0.0005995862 & -0.000076908 & -0.0020750124 \\ -0.000076908 & 0.0004751613 & 0.0002373498 \\ -0.0020750124 & 0.0002373498 & 0.0086288922 \end{bmatrix}$$

$$F_2^{-1}=\frac{\begin{bmatrix} 0.0005995862 & -0.000076908 & -0.0020750124 \\ -0.000076908 & 0.0004751613 & 0.0002373498 \\ -0.0020750124 & 0.0002373498 & 0.0086288922 \end{bmatrix}}{2.008533577\,e^{-05}}$$

$$F_2^{-1}=\begin{bmatrix} 29.85194 & -3.82906 & -103.30982 \\ -3.82906 & 23.65712 & 11.81707 \\ -103.30982 & 11.81707 & 429.61154 \end{bmatrix}$$

- matriz de covariância modificada:

$$M_2=\sqrt[3]{2.008533577\,e^{-05}}\cdot\begin{bmatrix} 29.85194 & -3.82906 & -103.30982 \\ -3.82906 & 23.65712 & 11.81707 \\ -103.30982 & 11.81707 & 429.61154 \end{bmatrix}$$

$$M_2=0.027182727\cdot\begin{bmatrix} 29.85194 & -3.82906 & -103.30982 \\ -3.82906 & 23.65712 & 11.81707 \\ -103.30982 & 11.81707 & 429.61154 \end{bmatrix}$$

$$M_2=\begin{bmatrix} 0.81146 & -0.10408 & -2.80824 \\ -0.10408 & 0.64307 & 0.32122 \\ -2.80824 & 0.32122 & 11.67801 \end{bmatrix}$$

- distâncias:

$$d^2(x_1,c_2)=\left(\left[0.0,0.6,1.2\right]-\left[0.72061,0.70016,1.23314\right]\right)\cdot M_2\cdot\left(\begin{bmatrix}0.0\\0.6\\1.2\end{bmatrix}-\begin{bmatrix}0.72061\\0.70016\\1.23314\end{bmatrix}\right)$$

$$d^2(x_1,c_2)=\left[-0.72061,-0.10016,-0.03314\right]\cdot M_2\cdot\begin{bmatrix}-0.72061\\-0.10016\\-0.03314\end{bmatrix}$$

$$d^2(x_1,c_2)=\left[-0.72061,-0.10016,-0.03314\right]\cdot\begin{bmatrix}-0.48126\\-0.00005\\1.60446\end{bmatrix}$$

$$d^2(x_1, c_2) = 0.29364$$

$$d^2(x_2, c_2) = \left(\left[1.0, 0.4, 1.3\right] - \left[0.72061, 0.70016, 1.23314\right]\right) \cdot M_2 \cdot \left(\begin{bmatrix} 1.0 \\ 0.4 \\ 1.3 \end{bmatrix} - \begin{bmatrix} 0.72061 \\ 0.70016 \\ 1.23314 \end{bmatrix}\right)$$

$$d^2(x_2, c_2) = \left[0.27939, -0.30016, 0.06686\right] \cdot M_2 \cdot \begin{bmatrix} 0.27939 \\ -0.30016 \\ 0.06686 \end{bmatrix}$$

$$d^2(x_2, c_2) = \left[0.27939, -0.30016, 0.06686\right] \cdot \begin{bmatrix} 0.0702 \\ -0.20063 \\ -0.10022 \end{bmatrix}$$

$$d^2(x_2, c_2) = 0.07313$$

$$d^2(x_3, c_2) = \left(\left[1.0, 0.9, 1.3\right] - \left[0.72061, 0.70016, 1.23314\right]\right) \cdot M_2 \cdot \left(\begin{bmatrix} 1.0 \\ 0.9 \\ 1.3 \end{bmatrix} - \begin{bmatrix} 0.72061 \\ 0.70016 \\ 1.23314 \end{bmatrix}\right)$$

$$d^2(x_3, c_2) = \left[0.27939, 0.19984, 0.06686\right] \cdot M_2 \cdot \begin{bmatrix} 0.27939 \\ 0.19984 \\ 0.06686 \end{bmatrix}$$

$$d^2(x_3, c_2) = \left[0.27939, 0.19984, 0.06686\right] \cdot \begin{bmatrix} 0.01816 \\ 0.12091 \\ 0.06039 \end{bmatrix}$$

$$d^2(x_3, c_2) = 0.03327$$

$$d^2(x_4, c_2) = \left(\left[0.0, 0.7, 1.0\right] - \left[0.72061, 0.70016, 1.23314\right]\right) \cdot M_2 \cdot \left(\begin{bmatrix} 0.0 \\ 0.7 \\ 1.0 \end{bmatrix} - \begin{bmatrix} 0.72061 \\ 0.70016 \\ 1.23314 \end{bmatrix}\right)$$

$$d^2(x_4, c_2) = \left[-0.72061, -0.00016, -0.23314\right] \cdot M_2 \cdot \begin{bmatrix} -0.72061 \\ -0.00016 \\ -0.23314 \end{bmatrix}$$

$$d^2(x_4, c_2) = \left[-0.72061, -0.00016, -0.23314\right] \cdot \begin{bmatrix} 0.06998 \\ 0.00001 \\ -0.69902 \end{bmatrix}$$

$$d^2(x_4, c_2) = 0.11254$$

c) distâncias encontradas:

	x_1	x_2	x_3	x_4
c_1	0.02605	0.11068	0.37018	0.06634
c_2	0.29364	0.07313	0.03327	0.11254

5. Cálculo dos graus de pertinência:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}^2}{d_{kj}^2} \right)^{\frac{1}{m-1}}}$$

a) primeiro *cluster*:

$$u_{11} = \frac{1}{\left(\frac{0.02605}{0.02605} \right)^{\frac{1}{2-1}} + \left(\frac{0.02605}{0.29364} \right)^{\frac{1}{2-1}}}$$

$$u_{11} = \frac{1}{1 + 0.08871}$$

$$u_{11} = 0.91852$$

$$u_{12} = \frac{1}{\left(\frac{0.11068}{0.11068} \right)^{\frac{1}{2-1}} + \left(\frac{0.11068}{0.07313} \right)^{\frac{1}{2-1}}}$$

$$u_{12} = \frac{1}{1 + 1.51347}$$

$$u_{12} = 0.39786$$

$$u_{13} = \frac{1}{\left(\frac{0.37018}{0.37018} \right)^{\frac{1}{2-1}} + \left(\frac{0.37018}{0.03327} \right)^{\frac{1}{2-1}}}$$

$$u_{13} = \frac{1}{1 + 11.12654}$$

$$u_{13} = 0.08246$$

$$u_{14} = \frac{1}{\left(\frac{0.06634}{0.06634} \right)^{\frac{1}{2-1}} + \left(\frac{0.06634}{0.11254} \right)^{\frac{1}{2-1}}}$$

$$u_{14}=\frac{1}{1+0.58948}$$

$$u_{14}=0.62914$$

b) segundo *cluster*:

$$u_{21}=\frac{1}{\left(\frac{0.29364}{0.02605}\right)^{\frac{1}{2-1}}+\left(\frac{0.29364}{0.29364}\right)^{\frac{1}{2-1}}}$$

$$u_{21}=\frac{1}{11.27217+1}$$

$$u_{21}=0.08148$$

$$u_{22}=\frac{1}{\left(\frac{0.07313}{0.11068}\right)^{\frac{1}{2-1}}+\left(\frac{0.07313}{0.07313}\right)^{\frac{1}{2-1}}}$$

$$u_{22}=\frac{1}{0.66073+1}$$

$$u_{22}=0.60214$$

$$u_{23}=\frac{1}{\left(\frac{0.03327}{0.37018}\right)^{\frac{1}{2-1}}+\left(\frac{0.03327}{0.03327}\right)^{\frac{1}{2-1}}}$$

$$u_{23}=\frac{1}{0.08988+1}$$

$$u_{23}=0.91754$$

$$u_{24}=\frac{1}{\left(\frac{0.11254}{0.06634}\right)^{\frac{1}{2-1}}+\left(\frac{0.11254}{0.11254}\right)^{\frac{1}{2-1}}}$$

$$u_{24}=\frac{1}{1.69641+1}$$

$$u_{24}=0.37086$$

c) pertinências encontradas:

	x_1	x_2	x_3	x_4
c_1	0.91852	0.39786	0.08246	0.62914
c_2	0.08148	0.60214	0.91754	0.37086

6. Cálculo do erro:

$$\|U^l - U^{l-1}\| \leq \varepsilon$$

$$\left\| \begin{bmatrix} 0.91852 & 0.39786 & 0.08246 & 0.62914 \\ 0.08148 & 0.60214 & 0.91754 & 0.37086 \end{bmatrix} - \begin{bmatrix} 0.67 & 0.41 & 0.24 & 0.5 \\ 0.33 & 0.59 & 0.76 & 0.5 \end{bmatrix} \right\| \leq 10^{-05}$$

$$\left\| \begin{bmatrix} 0.24852 & -0.01214 & -0.15754 & 0.12914 \\ -0.24852 & 0.01214 & 0.15754 & -0.12914 \end{bmatrix} \right\| \leq 0.00001$$

$$\begin{bmatrix} 0.24852 & 0.01214 & 0.15754 & 0.12914 \\ 0.24852 & 0.01214 & 0.15754 & 0.12914 \end{bmatrix} \leq 0.00001$$

Considerando que a condição acima é falsa, o algoritmo executaria a próxima iteração.