

SEGMENTAÇÃO DE DADOS EM UM NÚMERO DESCONHECIDO DE GRUPOS
UTILIZANDO ALGORITMO DE COLÔNIA DE FORMIGAS

Dilson Godoi Espenchitt

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS
EM ENGENHARIA CIVIL.

Aprovada por:

Prof. Nelson Francisco Favilla Ebecken, D.Sc.

Prof. Gilberto Carvalho Pereira, D.Sc.

Prof^a. Beatriz de Souza Leite Pires de Lima, D.Sc.

Prof^a. Marta Lima de Queirós Mattoso, D.Sc.

Prof. Mário Antônio Ribeiro Dantas, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

JANEIRO DE 2008

ESPENCHITT, DILSON GODOI

Segmentação de Dados em um Número
Desconhecido de Grupos Utilizando
Algoritmo de Colônia de Formigas [Rio de
Janeiro] 2008

XI,77 p., 29,7 cm (COPPE/UFRJ,
D.Sc., Engenharia Civil, 2008)

Tese - Universidade Federal do Rio de
Janeiro, COPPE

1. Segmentação de Dados
2. Colônia de Formigas
3. Mineração de Dados

I.COPPE /UFRJ II.Título(série)

DEDICATÓRIA

Dedico este trabalho à
minha esposa Thereza que
sempre me apoiou em todos
os momentos.

AGRADECIMENTOS

Agradeço:

À minha esposa Thereza pela ajuda, compreensão e apoio, aos meus filhos Maria Lydia e Antônio Augusto pelo carinho e paciência.

Ao meu orientador, Professor Nelson Ebecken pela acolhida, ensinamentos transmitidos, paciência, estímulo e confiança no trabalho por mim proposto. Com sua orientação precisa, sua experiência e inteligência sempre nos apontam uma direção segura a seguir.

Aos meus pais pela formação que me proporcionaram.

À minha sogra Maria Aparecida pelo suporte dados nas minhas ausências.

Ao Exmo. Sr. Contra-Almirante Bernardo José Pierantoni Gambôa, atual diretor do CASNAV(Centro de Análise de Sistemas Navais) e representando todos os demais, pelo apoio incondicional sem o qual não seria possível a realização do curso.

Aos “Casnavianos”, pelo apoio e incentivo durante a realização do curso, em especial a Sandra Tavares sempre pronta a ajudar nas formatações e ao Coronel Menezes pela disponibilidade em fazer as correções ortográficas.

Aos companheiros do Projeto da Avaliação Operacional das Fragatas Classe Niterói Modernizadas: Jorge Viot, Leo, Mauricio Guedes, Miguel, Lincoln, Tresisan, Cmte Malheiros, Cmte Régula, Cmte Bodini, Fernando Ayres (Papy), Ronaldo, Denise, Silvio, La Marca, Viviane e Bruno, que nos últimos tempos tiveram que aturar minha TPD (Tensão Pré Defesa).

Ao amigo Josir Gomes pela sua abnegação na implementação do algoritmo no software WEKA e pelas soluções propostas, sem esta ajuda a realização deste trabalho seria quase impossível.

Aos amigos Di Benedito, Willian e Renan pela ajuda prestada nas minhas dúvidas no ECLIPSE e em Java.

Aos amigos Gerson e Paulo Sérgio, sempre prontos a mostrar uma luz no final do túnel.

A amiga Sandra Fortes pelo seu suporte nos momentos difíceis.

Aos funcionários da COPPE/PEC, em especial ao Jairo Leite, Elisabeth e Eгна que sempre estavam prontos a atender as solicitações inerentes a vida acadêmica.

A todos os funcionários da COPPE pelo seu trabalho e dedicação em especial a Estela Sampaio pelo seu apoio administrativo.

E, acima de tudo, a Deus, por mais esta etapa vencida.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.).

SEGMENTAÇÃO DE DADOS EM UM NÚMERO DESCONHECIDO DE GRUPOS UTILIZANDO ALGORITMO DE COLÔNIA DE FORMIGAS

Dilson Godoi Espenchitt

Janeiro/2008

Orientador: Nelson Francisco Favilla Ebecken

Programa: Engenharia Civil

O objetivo deste trabalho é propor uma ferramenta que seja de fácil utilização para um usuário não especialista segmentar uma massa de dados em um número desconhecido de grupos.

O paradigma escolhido foi inspirado na teoria de Colônia de Formigas, onde os únicos parâmetros a serem selecionados são o número de formigas que serão usadas e o número de ciclos a serem executados pelo algoritmo não necessitando de nenhum conhecimento prévio da massa de dados. Este algoritmo foi implementado no software WEKA.

Os experimentos mostram que foi alcançado o objetivo principal desse trabalho. A implementação de um Algoritmo de Colônia de Formigas para Agrupamento de Dados no software WEKA, permite a um usuário não especialista executar a tarefa de agrupamento de dados.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

APPLYING ANT COLONY ALGORITHM TO PERFORM DATA
SEGMENTATION WHEN THE AMOUNT OF CLUSTERS IS UNKNOWN

Dilson Godoi Espenchitt

January/2008

Advisor: Nelson Francisco Favilla Ebecken

Department: Civil Engineering

The purpose of this work is to present an user friendly tool proposal for non-specialized users to segment a data set in an unknown number of groups.

The chosen paradigm was inspired by the Ant Colony theory, in which the only parameters required are the specifies number of ants and the number of cycles to be executed, not being necessary any data set previous knowledge. This algorithm was implemented in the WEKA software.

The experience shows that this work main's objective was acquired.

The implementation of an Ant Colony Algorithm for data clustering in WEKA software allows a non-specialist user to execute this task in a simple way.

ÍNDICE

1 - INTRODUÇÃO	1
1.1 - Motivação.....	1
1.2 - Objetivo.....	2
1.3 - Organização da Tese	3
2 - MINERAÇÃO DE DADOS OU DATA MINING.....	4
2.1 - Introdução	4
2.2 - Conceitos e Definições.....	5
2.3 - Descoberta de Conhecimento em Banco de Dados.....	6
2.4 - Bases da Mineração de Dados.....	10
2.5 - Principais Tarefas do <i>Data Mining</i>	10
2.5.1 - Classificação	12
2.5.2 - Associação.....	12
2.5.3 - Agrupamento.....	12
2.5.4 - Previsão	12
3 - AGRUPAMENTO DE DADOS	13
3.1 - Introdução	13
3.2 - Tipos de Atributos em Problemas de Agrupamento	15
3.2.1 - Atributos Numéricos	16
3.2.1.1 - Normalização	16
3.2.1.2 - Medidas de Distância	16
3.2.2 - Atributos Binários	17
3.2.3 - Atributos Nominais	17
3.2.4 - Atributos Ordinais	18
3.2.5 - Atributos Mistos.....	18
3.3 - Características dos Problemas de Agrupamento	18
3.3.1 - Métodos Particionais.....	22
3.3.2 - Métodos Hierárquicos	23
3.3.3 - Métodos Baseados em Densidade	24
3.3.4 - Métodos Baseados em Malhas	27
3.3.5 - Métodos Baseados em Modelos.....	28
3.3.5.1 - Algoritmo <i>Expectation Maximization</i>	30
3.4 - Novas Tendências	31
3.5 - O problema da Definição do Número de Grupos.....	32
4 - ALGORITMOS EVOLUTIVOS	34
4.1 - Introdução	34
4.2 - Colônia de Formigas	37
4.3 - Formigas Reais.....	39
4.4 - Formigas Artificiais	41
4.4.1 - Comparação entre as Formigas Artificiais e as Formigas Reais.....	42
4.4.2 - Desenvolvimento de um Algoritmo <i>ACO</i>	44
4.5 - Algoritmo de Colônia de Formigas para Agrupamento de Dados.....	45
4.6 - Núcleo do Algoritmo.....	46
4.7 - Implementação do Algoritmo de Colônia de Formigas para Agrupamento de Dados.....	48
4.8 - O Software WEKA.....	50

5 - METODOLOGIA PARA SE DETERMINAR OS PARÂMETROS INTERNOS DO ALGORITMO DE COLÔNIA DE FORMIGAS PARA AGRUPAMENTO DE DADOS	52
5.1 - Determinação do Número de Formigas	57
5.2 - Número de Interações	57
5.3 - Formação do <i>Cluster</i>	57
5.4 - Outros Parâmetros	58
5.5 - Bases MultiDimensionais.....	59
6 - ESTUDO DE CASO	61
6.1 - Resultados Apresentados pelo Algoritmo Colônia de Formiga para Agrupamento de Dados	61
6.2 - Resultados Apresentados pelo Algoritmo <i>Expectation Maximization</i>	62
6.3 - Resultados Finais	64
7 - CONCLUSÃO E SUGESTÕES DE TRABALHOS FUTUROS.....	65
REFERÊNCIA BIBLIOGRÁFICA.....	68

ÍNDICE DE FIGURAS

Figura 1 -	Etapas do KDD. Adaptado Han & Kamber, 2006	7
Figura 2 -	Data Mining e Disciplinas Correlatas(COELHO, 2006)	10
Figura 3 -	Taxonomia de Data Mining (COELHO, 2006).....	11
Figura 4 -	Segmentação Aglomerativos e Divisivos(HAN & KAMBLER,2006)	24
Figura 5 -	Árvore de Classificação	29
Figura 6 -	Gaussianas Circulares (Zadrozny, 2006)	30
Figura 7 -	Gaussianas Elípticas (Zadrozny, 2006).....	31
Figura 8 -	Métodos de Agrupamnetos	32
Figura 9 -	Transposição de um Obstáculo (DORIGO, 2006)	41
Figura 10 -	Diagrama de Classe.....	49
Figura 11 -	Arquivo no Formato ARFF	51
Figura 12 -	ClusterTeste.....	52
Figura 13 -	ClusterIdeal	53
Figura 14 -	p02.5_ 4 Elipses	54
Figura 15 -	p04.5 _9Elipses	54
Figura 16 -	p37.5_4Arcos	55
Figura 17 -	p15.5_3 Elipses	55
Figura 18 -	Tela de Inicialização do WEKA	56
Figura 19 -	Seleção do Algoritmo de Clusterização	56
Figura 20 -	Parâmetros Configuráveis	57
Figura 21 -	Parâmetros do Algoritmo EM	63

ÍNDICE DE TABELAS

Tabela 1 -	Ficha Médica dos Pacientes	18
Tabela 2 -	Tipos de Atributo	18
Tabela 3 -	Características dos algoritmos.....	33
Tabela 4 -	Parâmetros do algoritmo	48
Tabela 5 -	Parâmetros Validados.....	53
Tabela 6 -	Resultados dos Experimentos	60
Tabela 7 -	Características das bases de dados	61
Tabela 8 -	Resultados do Algoritmo de Colônia de Formigas para Agrupamento de Dados	62
Tabela 9 -	Resultados do Algoritmo Expectation Maximization	63
Tabela 10 -	Quadro resumo dos resultados	64

1 - INTRODUÇÃO

Nos últimos anos houve um crescimento substancial da quantidade de dados armazenados (SANTOS, 2007), devido ao aumento da facilidade e à redução dos custos para manter esses dados. Isto acontece em todos os campos do conhecimento humano.

Essa imensa quantidade de dados armazenada é inviável de ser analisada por especialistas através de métodos convencionais. Assim, a dificuldade de uma análise mais precisa desses dados colabora para que os mesmos se transformem em apenas um amontoado de itens sem utilidade. Por outro lado, sabe-se que nas grandes quantidades de dados pode existir um enorme potencial de informação, muito embora os conhecimentos contidos nos dados não estejam caracterizados explicitamente, já que sendo esses dados operacionais, não interessam quando estudados individualmente.

Logo, a descoberta de conhecimento em bases de dados vem ganhando grande importância e interesse, pois, as pesquisas nessa área aumentaram e visam à construção de tecnologias mais eficientes para a recuperação de informações, procurando encontrar conhecimentos implícitos que possam ser úteis (FAYYAD *et al.*, 1996). Então, denomina-se esse processo como Mineração de Dados, ou seja, a análise de grandes volumes de dados utilizando técnicas computacionais para extração de conhecimento não trivial e potencialmente útil.

Mas, minerar os dados mantém a tarefa de transformá-los em conhecimento ainda muito árdua. Logo, busca-se avidamente uma forma, científica e não empírica, de extrair conhecimento, mesmo que a base seja muito grande e, logicamente, seus inúmeros componentes não conhecidos.

Uma das soluções pesquisadas é o agrupamento de dados, sendo que, uma ferramenta com uma interface amigável que não exija profundos conhecimentos de matemática, estatística ou informática, para o usuário final tornará esta solução cada vez mais aplicável.

1.1 - MOTIVAÇÃO

A análise de dados e conseqüentemente a descoberta de conhecimentos a partir de uma massa de dados, como já mencionado anteriormente, só é possível se os dados forem analisados de forma conjunta. Portanto, sistemas computacionais eficientes são pré-requisitos para se poder analisar os dados, de forma eficiente.

A maioria das tarefas de Mineração de Dados sofre fortes restrições para serem realizadas por um usuário comum, ou seja, aquele que é especialista na massa de dados que irá ser analisada, mas não tem domínio das ferramentas existentes, pois as mesmas, muitas vezes, requerem um nível de conhecimento técnico em diversas áreas.

Quando vamos determinar o número de grupos para uma base de dados o problema se torna um pouco mais complexo, pois além de diversos parâmetros que temos que ajustar, o que requer conhecimento específico do algoritmo que está sendo utilizado, alguns algoritmos requerem o número (n) de conjuntos (*clusters*) nos quais queremos particionar a nossa massa de dados.

Após usarmos diversos “n”, aplicamos índices semi-empíricos tais como, Calinski e Harabasz, Critério Condorcet, *Cubic Clustering Criterion* e PBM, para ver qual o melhor “n”. Este tipo de classificação não supervisionada encontra vasta literatura e tem sido tema de pesquisa ininterrupta. (MACHADO, 2002, PUNTAR, 2003, MORAES, 2004, ANDRADE, 2004).

Temos também que levar em conta que a clusterização é uma tarefa não supervisionada, o que sempre gera dificuldades para definir o número de clusters existentes nas estruturas dos dados.

Recentemente, informações privilegiadas ou restrições conhecidas, têm sido utilizadas durante a tarefa de clusterização. Ou seja, em alguns casos existem amostras que necessariamente pertencem a determinados clusters, e estas informações auxiliam ou interferem no processo de clusterização, denominando este processo de aprendizado semi-supervisionado. Isso é particularmente importante na área de mineração de textos (DA SILVA, 2006).

O agrupamento de dados continua sendo uma tarefa desafiadora, pois ainda não se encontrou uma resposta definitiva para este problema. Está sempre atraindo os pesquisadores em busca de algoritmos inspirados em novas teorias, tais como sistemas imunológicos artificiais, colônia de formigas, *flock agents*, algoritmos genéticos, algoritmos *swarm*, (LIU *et al*, 2007, AZZAG *et al*, 2007, HANDL, MEYER, 2007, NUANNUAN *et al*, 2005, AZZAG *et al*, 2004, COELHO, EBECKEN, 2001).

1.2 - OBJETIVO

Este trabalho tem o objetivo de propor uma ferramenta que seja de fácil utilização, para um usuário não especialista segmentar uma massa de dados em um

número desconhecido de grupos, colaborando desta forma para a extração de informação e conhecimento dos dados armazenados.

Para isto, buscou-se na literatura um algoritmo que realizasse a tarefa de segmentar os dados sem que fosse necessário selecionar, a priori, um número inicial de conjuntos e que não necessitasse de parâmetros complexos para se ajustar. Além disso, procurou-se um ambiente que fosse o mais amigável possível para o usuário.

O paradigma escolhido foi inspirado na teoria de Colônia de Formigas, onde os parâmetros a serem selecionados são: o número de “formigas” que serão usadas e o número de ciclos que serão executados, não necessitando do conhecimento, a priori, do número de *clusters* que iremos utilizar. Este algoritmo foi inserido no software WEKA, desenvolvido pela Universidade de Waikato, Nova Zelândia. A escolha se deu por ser um software livre, pela simplicidade de uso, e para aproveitar as facilidades de entrada e saída de dados além das interfaces gráficas residentes.

1.3 - ORGANIZAÇÃO DA TESE

Esta tese é composta, além deste capítulo, de outros seis capítulos adicionais que estão organizados da seguinte forma:

- O Capítulo 2 aborda conceitos básicos de Mineração de Dados, para o entendimento dos principais assuntos discutidos na Tese.
- O Capítulo 3 trata de Agrupamento de Dados, que é o assunto principal da Tese. Ele descreve as principais características dos problemas de agrupamento e mostra os principais métodos de segmentação de dados.
- O Capítulo 4 apresenta o princípio dos Algoritmos Evolutivos e os princípios do Algoritmo de Colônia de Formigas para Agrupamento de Dados.
- O Capítulo 5 apresenta metodologia para a determinação dos Parâmetros Internos do Algoritmo de Colônia de Formigas para Agrupamento de Dados, o método para formação de *clusters* e o método para determinação do número de *clusters*.
- O Capítulo 6 apresenta estudos de casos com a comparação entre o Algoritmo de Colônia de Formigas para Agrupamento de Dados e o Algoritmo *Expectation Maximization*.
- O Capítulo 7 apresenta as conclusões e sugestões de trabalhos futuros.

2 - MINERAÇÃO DE DADOS OU DATA MINING

2.1 - INTRODUÇÃO

É a área do conhecimento que surge a partir da interação de ciências principalmente quantitativas, como computação e estatística, para extrair conhecimento de uma grande quantidade de dados armazenado.

Nas últimas décadas observamos que a quantidade de dados a serem armazenados e manipulados cresceu vertiginosamente que a tarefa de extrair algum conhecimento desta massa de dados está ficando cada vez mais árdua.

Segundo COUTINHO, 2003, *Data Mining* (DM) é um processo para extrair informação válida, previamente desconhecida e de máxima abrangência a partir de grandes bases de dados e/ou armazéns de dados, denominados *Datawarehouse* (DW). O DM vai muito além da simples consulta a um banco de dados. Permite ao usuário explorar e inferir informações úteis a partir dos dados, descobrindo relacionamentos escondidos no banco de dados e buscando padrões válidos. É, sem dúvida, um conjunto de técnicas utilizadas, para descobrimento de conhecimento em base de dados robustas (*Knowledge Discovery in Databases- KDD*).

Talvez a definição mais importante de *Data Mining* tenha sido elaborada por FAYYAD *et al.*(1996): "...o processo não-trivial de identificar, em dados, padrões válidos, novos, potencialmente úteis e ultimamente compreensíveis".

Gerar conhecimento a partir desse estado caótico de dados armazenados faz com que o profissional ligado à área de informação, nos dias atuais, seja extremamente requisitado na busca de reconhecimento de padrões válidos, a ponto de se tornarem verdadeiros profissionais de lapidação dos dados. Nunca se ouviu falar tanto em termos, como: extração de conhecimento, descoberta de informação, arqueologia de dados, mineração de dados, mineração de textos (*text mining*), mineração de sites (*web mining*), entre outros.

As origens do DM encontram-se nos idos de 60, quando a análise de dados basicamente era oferecida através de análise estatística. Segundo muitos autores, DM não passava de rotinas estatísticas clássicas, incluindo nesse conjunto de rotinas, a estatística multivariada.

Com o passar dos anos, novas técnicas foram sendo agregadas à estatística. Surgem os conceitos de Lógica Nebulosa (*Fuzzy Logic*), Redes Neurais Artificiais

(*Artificial Neural Networks*), Árvores de decisão (*Decision Tree*). Todavia, no passado, muitos dos pesquisadores não sabiam classificar, com exatidão, em que área de conhecimento essas técnicas poderiam ser agregadas. A melhor tentativa de classificação destas técnicas surge no início da década de 1990, onde muitos pesquisadores concordaram em classificá-las como sendo técnicas de Inteligência Artificial.

Atualmente, costuma-se dizer que DM é uma área de conhecimento interdisciplinar, que envolve uma série de técnicas emprestadas de outras áreas de conhecimento. Para os pesquisadores mais ortodoxos, pode parecer uma heresia chamar DM de área de conhecimento, uma vez que, para eles, seria apenas um conjunto de técnicas que permite, a partir de informações tiradas de uma massa de dados, inferir conhecimentos. Todavia, a especialização de diversos pesquisadores oriundos de outras áreas de conhecimento, tornou-a tão robusta que permite dizer que DM já pode ser considerada uma área de conhecimento.

2.2 - CONCEITOS E DEFINIÇÕES

De acordo com COELHO, 2006, “Um conceito adequado para DM deve necessariamente envolver os termos conhecimento e dado. Neste sentido, os dados representam a matéria prima e o conhecimento é o produto final da indústria do DM”.

Os dados são fatos disponíveis que, ou estão em uma forma estruturada e num ambiente digital, como bancos de dados operacionais, ou podem ser assim codificados. O conhecimento descoberto pelo processo de DM é uma generalização que pode ser obtida por meio dos dados. Esta generalização é também comumente chamada de padrão, no sentido de que é uma estrutura recorrente nos dados analisados. Além disso, dado que o fenômeno gerador dos fatos é ignorado ou apenas parcialmente conhecido, o produto obtido, conhecimento, é oculto. Por isso, usa-se também o termo descoberta de conhecimento para estas atividades.

Em COELHO,2006, são examinados três conceitos, e em seguida analisados:

Data Mining refere-se à extração ou “mineração” de conhecimento a partir de grandes quantidades de dados. (HAN & KAMBER, 2006);

Data Mining é definido como o processo de descoberta de padrões nos dados. O processo precisa ser automático ou (mais usualmente) semi-automático. Os padrões descobertos devem ser significativos, pois devem trazer alguma vantagem, geralmente

no sentido econômico. Os dados são invariavelmente apresentados em quantidades substanciais. (WITTEN & FRANK, 2005);e

Data Mining é um processo iterativo no qual o progresso é definido pela descoberta, através de métodos manuais e/ou automáticos. (WESTPHAL & BLAXTON, 1998).

Os dois primeiros conceitos enfatizam a questão da abundância dos dados disponíveis, que é uma preocupação particular dentro da área. A quantidade de dados é frequentemente tratada como o problema da escalabilidade, onde a escala reflete diretamente o volume dos dados.

Data Mining, enquanto área do conhecimento, tem como um dos objetivos de estudo a eficiência das técnicas, mais especificamente dos algoritmos, diante de grandes volumes de dados. Neste sentido, diversas técnicas clássicas de análise de dados foram recriadas ou adaptadas de maneira a tornarem-se eficientes no aspecto da escalabilidade, ou seja, novos algoritmos foram desenvolvidos de maneira que métodos clássicos pudessem ser aplicados em grandes bancos de dados, de forma eficiente.

Os dois últimos conceitos enfatizam a questão da iteração do processo de *Data Mining*, no sentido que a análise é constante e o processo é desenhado em diversos passos.

Este processo total é preferivelmente chamado de KDD – *Knowledge Discovery in Databases*, ou Descoberta de Conhecimento em Bancos de Dados. O processo é dividido em etapas que vão desde a localização e extração dos dados até a compreensão do conhecimento modelado. A construção do modelo de conhecimento, que é como preferimos definir *Data Mining*, é apenas uma das etapas intermediárias do KDD (BERRY & LINOFF, 2004, BERSON, *et al*, 1999, HAN & KAMBER, 2006, WEISS & INDURKHYA, 1998).

2.3 - DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS

Podemos dizer que a Descoberta de Conhecimento em Banco de Dados (*Knowledge Discovery in Databases* -KDD), consiste em uma metodologia de trabalho que vai da localização e extração dos dados até a compreensão e interpretação do conhecimento modelado, passando pela etapa de construção do modelo de conhecimento – Mineração dos dados. (BERRY & LINOFF, 2004, BERSON, *et al*, 1999, HAN & KAMBER, 2006, WEISS & INDURKHYA, 1998).

O conceito de KDD fica aqui diferenciado do conceito de Mineração de Dados. Convém observar que esta diferenciação não está bem definida na literatura. Para alguns autores, os conceitos podem ser considerados os mesmos. A Mineração de Dados é, nesta análise, apenas uma das fases do KDD que, ao todo, compreende cinco segmentos de trabalho: definição do problema (seleção dos dados); pré-processamento e limpeza dos dados; transformação do formato do arquivo; Mineração de Dados; e finalmente, interpretação e análise de resultados (COELHO, 2006).

A Figura 1 apresenta um esquema com as etapas contidas no processo de KDD, segundo HAN & KAMBER, (2006).

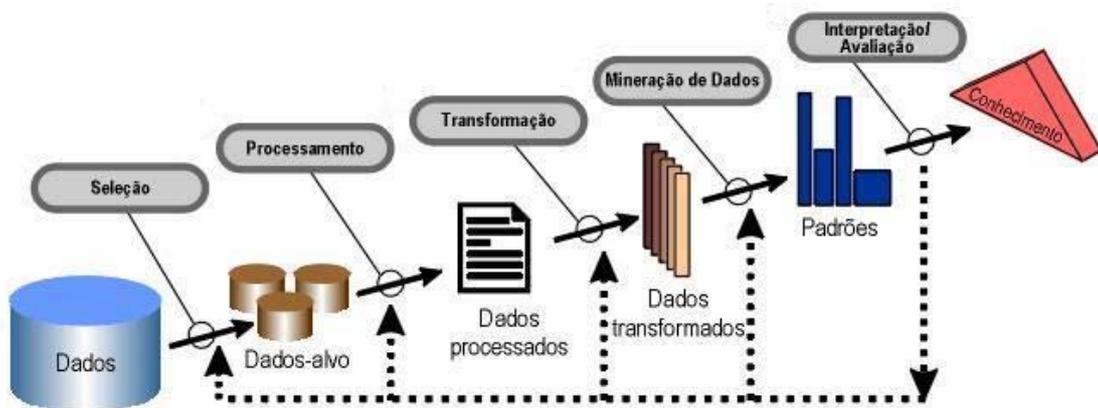


Figura 1 - Etapas do KDD. Adaptado Han & Kamber, 2006

A primeira etapa é a atividade de seleção das informações contidas na fonte de dados. Observe que esta fonte não é necessariamente um banco de dados ou um *Data Warehouse*. É possível extrair conhecimento a partir de um conjunto de dados que esteja em outras fontes, como planilhas, tabelas construídas a partir de questionários, pesquisa na internet, etc. De qualquer maneira, é necessário coletar todos os dados que são considerados relevantes e eliminar aqueles que forem considerados desnecessários. A eliminação pode parecer desnecessária, mas realmente não é. Duas razões principais devem ser consideradas. A primeira está ligada à não relevância de algumas das informações armazenadas para o problema que está sendo analisado. A segunda razão está ligada ao volume de dados existente. Muitas vezes a análise da totalidade dos dados pode ser proibitiva em termos de tempo e da capacidade computacional disponível para o processo. Então, uma seleção dos dados mais relevantes se faz necessária. A título de exemplo, quando a fonte de dados é um *Data Warehouse* é comum descartar uma

grande quantidade de dados, já que a redundância nestes sistemas é, geralmente, muito grande.

A segunda etapa do processo de descoberta de conhecimento é possivelmente a etapa mais importante de todo o processo. O pré-processamento é fundamental para a etapa de modelagem do conhecimento (Mineração de Dados). Um pré-processamento ineficiente pode reduzir ou até eliminar as chances de uma modelagem eficiente (Pyle, 1999).

Nesta etapa, os dados são preparados para a modelagem, resolvendo-se problemas como os de redundância, inconsistência e ausência de valores, sendo por isso chamada de etapa de preparação ou de limpeza. Observe que, se estes problemas não forem resolvidos, ou se o forem, mas de forma incompleta, a etapa de modelagem de dados poderá apresentar resultados insatisfatórios, contraditórios ou inválidos.

Um outro importante aspecto sobre a etapa de preparação de dados é que, na maioria dos casos não há nenhuma maneira de estimar a qualidade do resultado. Quando valores desconhecidos estão sendo substituídos é comum que não se tenha uma estimativa da precisão dos valores que estão sendo substituídos, ou que se tenha uma estimativa muito fraca.

A partir dos dados limpos e pré-processados, a etapa de modelagem só pode ser aplicada se houver uma compatibilidade entre o formato dos dados e os requisitos do software que fará a modelagem. A etapa de transformação é responsável por garantir esta compatibilidade, que geralmente não ocorre naturalmente.

É importante considerar que diferentes técnicas ou diferentes ferramentas computacionais podem ser empregadas na etapa de modelagem, e o formato de entrada dos dados depende das restrições do sistema a ser utilizado. A etapa de transformação parte dos dados limpos e os adapta ao formato especificado pela ferramenta de modelagem que será utilizada. Se for necessário modelar os dados utilizando ferramentas diferentes (o que foi feito para comparação dos resultados, por exemplo), a etapa de transformação tomará sempre como entrada o resultado do pré-processamento (único), gerando diferentes formatos, um para cada uma das ferramentas que será utilizada. (COELHO,2005)

A etapa de Mineração de Dados é a captação e estruturação das características e padrões dos dados que estão sendo analisados em forma de algum modelo matemático-estatístico. Estas características e padrões podem, dentro de limites estimados através de intervalos de confiança e testes de hipóteses, ser utilizadas para

projeções de fatos não observados. Esta etapa realiza, além da construção dos modelos, as avaliações através de estimativas. Pode-se considerar que esta etapa está efetivamente relacionada com a descoberta do conhecimento.

Finalmente, a última etapa, chamada de interpretação, é responsável pela adequação da saída da ferramenta de modelagem às necessidades do usuário. Algumas ferramentas de visualização e de navegação em dados podem ser utilizadas.

Observe que estas atividades têm uma ordem bem estabelecida. Entretanto, em alguma etapa podem ser evidenciados problemas ocorridos em alguma etapa anterior. Assim, o processo de KDD pode incluir a repetição de alguma(s) etapa(s). Um exemplo típico é a etapa de preparação de dados. É possível que algum problema nos dados só venha a ser percebido na fase de interpretação do modelo gerado depois da fase de Mineração de Dados. A preparação de dados teria de ser refeita assim como todas as etapas subsequentes.

Uma outra alteração comum no fluxo do KDD é quando alguma etapa pode ser suprimida como, por exemplo, a etapa de transformação dos dados. Existem algumas ferramentas que fazem todo o processo de KDD, desde a extração dos dados até a interpretação do modelo. Nestes casos não é preciso fazer a transformação dos dados, pois a própria ferramenta o fez no momento da importação dos dados.

Uma diferença significativa entre DM e outras ferramentas de análise está na maneira como exploram as inter-relações entre os dados. As diversas ferramentas de análise disponíveis utilizam um método baseado na verificação, isto é, o usuário constrói hipóteses sobre inter-relações específicas e, então, verifica ou refuta, com o uso do sistema.

Este modelo torna-se dependente da intuição e habilidade do analista em propor hipóteses interessantes, em manipular a complexidade do espaço de atributos e em refinar a análise, baseado nos resultados de consultas ao banco de dados potencialmente complexos. Já o processo de *Data Mining* fica responsável pela geração de hipóteses, garantindo mais rapidez, precisão e completude aos resultados. Mas como se dá essa garantia de rapidez e completude dos resultados?

A resposta pode ser encontrada facilmente. Todavia, a que mais chama a atenção, é a que diz que DM gera por si só um modelo (modelagem matemática), determinando, dessa forma, padrões a partir de dados observados. O perfeito ajuste desse modelo é de fundamental importância no que se refere ao descobrimento de

conhecimento. Se o modelo representa (ou não), o conhecimento buscado dependerá do analista/especialista que interage a todo o momento com os processos de KDD.

É importante ratificar que o processo KDD não é independente de um especialista humano, uma vez que será ele o responsável pela validação, ou não, do conhecimento extraído. Somente o especialista possui o sentimento e a sensibilidade de sua área de conhecimento, tornando-se, deste modo, parte essencial na modelagem do KDD.

2.4 - BASES DA MINERAÇÃO DE DADOS

Segundo COUTINHO, 2003, a Mineração de Dados, de maneira didática, é fundamentada na estatística, na inteligência artificial e no *machine learning*, podendo ainda ser acrescentada de Banco de Dados, Ciência da Informação, Visualização e outras disciplinas, como pode ser visto na Figura 2.

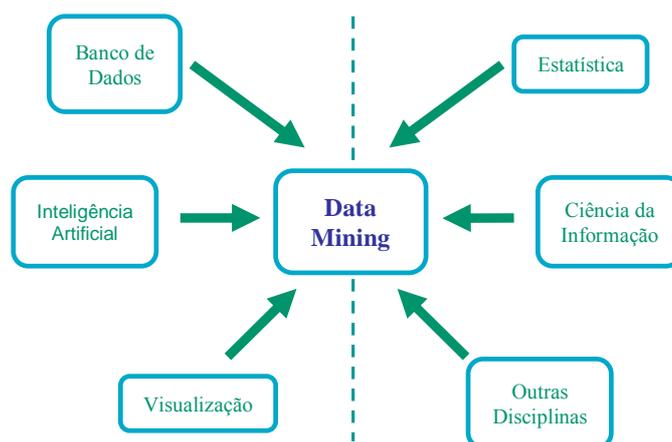


Figura 2 - Data Mining e Disciplinas Correlatas(COELHO, 2006)

O que está sendo chamado de “Outras Disciplinas” na figura acima refere-se a eventuais contribuições vindas da Economia, Biologia ou Psicologia (HAN & KAMBER, 2006).

2.5 - PRINCIPAIS TAREFAS DO *DATA MINING*

Para poder descrever como o *Data Mining* funciona, COELHO, 2006, desenvolveu uma taxonomia que permite isolar os diversos termos encontrados, em grupos dentro de uma hierarquia de conceitos.

Ainda segundo COELHO, 2006, um dos maiores problemas na busca de uma taxonomia como esta é dar nome aos níveis taxonômicos. Em função disto, definiu-se que os algoritmos de *Data Mining* estão classificados no contexto de uma determinada

técnica. A técnica reúne diversos algoritmos que tenham em comum algumas heurísticas ou estratégias de atuação. Os problemas representam um nível acima das técnicas, pois representam um tipo de padrão que se deseja obter. Finalmente, existem as funcionalidades, nas quais os problemas estão inseridos. Na figura 3 é possível observar as duas funcionalidades de *Data Mining*: a Descritiva e a Preditiva (HAN & KAMBLER, 2006), assim como as principais técnicas: Regras de Associação, Cluster, Classificação e Previsão. A tentativa de descrever as técnicas mais importantes tornaria esta estrutura mais confusa, pois existem técnicas que estão relacionadas com mais de um problema. Por exemplo, a técnica de Redes Neurais pode ser utilizada para problemas de Classificação e Previsão (na funcionalidade Preditiva) ou ainda, de Cluster (na funcionalidade Descritiva).

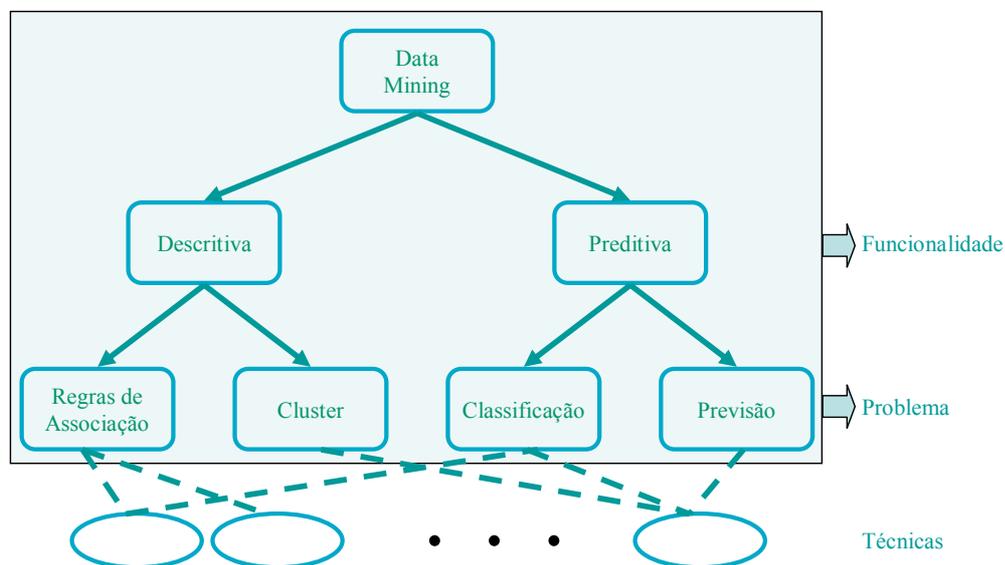


Figura 3 - Taxonomia de Data Mining (COELHO, 2006)

Os problemas estão mais diretamente relacionados com o tipo de modelo que se deseja estabelecer. Enquanto os dois problemas da funcionalidade descritiva são bem diferentes entre si, e sempre descritos de maneira totalmente independente, os da funcionalidade preditiva não são tão diferenciados pela literatura. Em geral, as técnicas preditivas são tratadas como sendo de um único problema, chamado de “Classificação e Previsão”, como por exemplo, em Han & Kamber (2006). Isto ocorre fundamentalmente por que os principais algoritmos de Previsão podem ser usados para Classificação, e vice-versa. As técnicas não estão descritas na Figura 3 pois, como já foi dito a estrutura ficaria confusa já que uma técnica poder estar relacionada a mais de um problema (COELHO,2006).

2.5.1 - Classificação

É usada para predição de uma classe ou categoria discreta a partir de atributos de entrada. Os algoritmos de classificação analisam os dados para os quais as categorias discretas são conhecidas. Os algoritmos de classificação podem ser utilizados em sistemas de detecção de intruso em redes de computadores ou levantamento de fraude em instituições financeiras.

2.5.2 - Associação

Descoberta de co-ocorrência entre elementos em grandes grupos de dados. Consiste em identificar quais atributos estão associados com outros em um dado ambiente. Um exemplo típico seria tentar descobrir quais produtos vendem bem, juntos, em um supermercado ou uma loja.

2.5.3 - Agrupamento

Descoberta de grupos ou *clusters* naturais nos quais dados que estão em um mesmo agrupamento são considerados semelhantes e dados em grupos distintos são considerados diferentes. A clusterização difere da classificação no fato de não repartir registros de dados em subgrupos predefinidos. Aglomeração divide uma população na base da auto-similaridade entre registros. Ela produz uma descrição de alto nível de uma população baseada somente numa medida de distância entre suas entidades. Nenhum modelo de classificação é construído. Um exemplo típico é o processamento de imagens de satélite para identificar queimadas.

2.5.4 - Previsão

A previsão consiste em examinar atributos de um conjunto de entidades e, baseado nos valores destes atributos, assinalar valores a atributos de uma nova entidade que se quer caracterizar. O termo previsão é usado quando a estimação é usada para prever o futuro valor de um atributo. Um exemplo bem interessante (complexo e sujeito a erros) é a previsão de indicadores econômicos financeiros.

3 - AGRUPAMENTO DE DADOS

3.1 - INTRODUÇÃO

O processo de agrupamento de objetos físicos ou abstratos em classes ou grupos de objetos similares é chamado de clusterização, que consiste na divisão dos dados em classes ou *clusters*, de maneira que objetos dentro de um mesmo *cluster* tenham alta similaridade, mas objetos pertencentes a *clusters* diferentes sejam muito distintos (alta dissimilaridade). Em geral, para que os procedimentos possam medir a similaridade ou a diferença entre os objetos que estão sendo avaliados, diversas medidas são usadas. Estas medidas normalmente são baseadas em métricas de distância, e consideram adaptações para dados especiais (variáveis binárias, nominais, ordinais etc.), (Han & Clamber, 2006).

Mais especificamente, pode-se definir o problema de segmentação da seguinte maneira:

Seja $X = \{ X_1, X_2, \dots, X_n \}$ um conjunto com n elementos. Cada elemento do conjunto X , X_i é um vetor de R^p , representando um objeto através de p medidas que o descreve. Estas medidas são chamadas de atributos. Cada elemento de X deve ser considerado pertinente a um dos grupos $C = \{ C_1, C_2, \dots, C_k \}$, aonde k é o número de *clusters*. Diz-se que C é a segmentação obtida para X , se podem ser observadas três características do conjunto C :

1. $C_1 \cup C_2 \dots \dots \dots \cup C_k = X$;
2. $C_i \neq \phi; \forall_i$;
3. $C_i \cap C_j = \phi; \forall_{i=j}$.

A análise de agrupamento de dados é uma ferramenta bastante útil para o estudo e compreensão do comportamento de dados nas mais diferentes situações. Um exemplo disto é o caso de dados coletados através de pesquisas, onde pode-se obter uma quantidade extremamente grande de informações que, observados sob um contexto geral, podem não apresentar nenhum sentido, porém quando classificados e separados em grupos passam a fornecer informações com respeito ao comportamento de cada um destes grupos (HAIR *et al*, 1998). Uma outra situação onde a análise de agrupamento de dados tem grande utilidade são os casos onde se deseja desenvolver hipóteses

concernentes à natureza de uma massa de dados, ou mesmo examinar-se a veracidade de hipóteses previamente concebidas.

Dependendo dos interesses e objetivos que se deseje atingir, a visualização de agrupamento de dados nos possibilita obter uma preciosa ajuda para um rápido entendimento e assimilação de informações por meio das quais se possa avaliar fatores tais como: o quanto os grupos foram bem definidos. Como se diferenciam uns dos outros; seus tamanhos; a pertinência total ou parcial de uma amostra.

Uma análise de dados, portanto, pode ter basicamente dois objetivos primordiais, primeiramente poderá tratar-se de uma análise exploratória onde se buscará obter, dos dados em questão, as informações relevantes que eles possam conter e que não estão claramente mostradas em uma simples observação. Um outro objetivo seria uma análise confirmatória, onde os dados são utilizados para confirmar-se supostas informações previamente esperadas e que se acredita estarem neles contidas, e que podem ser confirmadas após uma maior exploração dos dados.

Agrupamento de dados é uma tarefa que procura segmentar populações heterogêneas em subgrupos ou segmentos homogêneos. Os registros são agrupados conforme alguma similaridade em si. (JAIN *et al*, 1999)

A simplicidade de uma estrutura se reflete em função do número de grupos, ou seja, uma estrutura é tão mais simples quanto menor possível for o número de grupos. Entretanto há que se considerar que a diminuição do número de grupos acarreta necessariamente uma diminuição também na homogeneidade dentro dos mesmos; portanto é necessário que exista um balanceamento entre o número de grupos e a similaridade entre eles. (JAIN *et al*, 1999).

Além disso, como parte do procedimento da análise de agrupamento de dados, pode-se executar uma redução dos mesmos, de forma a obtermos uma quantidade menor, porém extremamente objetiva das informações sobre uma população inteira de dados. A redução de dados nos fornece amostragem com informações a respeito de subgrupos menores, podendo-se assim obter uma descrição mais concisa e mais compreensível das observações, com uma perda mínima das mesmas.

Uma tarefa que pode ser associada ao agrupamento de dados é a identificação de pontos fora dos padrões. Os grupos representam registros ou objetos similares, entretanto existem muitos objetos que não apresentam uma forte pertinência a nenhum dos grupos em questão. Estes são exemplos de pontos fora dos padrões, que podem ser vistos como anomalias e pontos não bem assentados. (HAN & KAMBER, 2006).

Dependendo do segmento de negócios representado pelo conjunto de dados em análise, os pontos fora dos padrões podem, por exemplo, representar transações fraudulentas ou um comportamento não usual de cliente ou ainda uma tendência.

Pontos fora dos padrões são, portanto, as observações cujas características os identificam distintamente dos demais, ou seja, o ponto resultante da combinação de suas características apresenta-se comparativamente diferente dos demais. Pontos fora dos padrões não podem ser categoricamente caracterizados como benéficos tampouco como problemáticos, mas observados dentro do contexto da análise e avaliados em função do tipo de informação que poderão fornecer (HAIR *et al*,1998).

Um ponto fora do padrão poderá ser indicativo de alguma característica da população que não tenha sido revelada durante o curso normal da análise. Neste caso, ainda que distante da maioria das demais observações, apresenta um importante benefício na análise. Por outro lado, poderá também ser contrário aos objetivos da análise, distorcendo seriamente os testes estatísticos e, neste caso, trata-se de um ponto problemático, não sendo, portanto representativo da população. (HAIR *et al*,1998, HAN & KAMBER,2006)

Devido a estes diferentes aspectos na interpretação de um ponto fora do padrão, é de suma importância que se verifique os dados em análise a fim de se determinar os tipos de influências que os mesmos poderão causar.

Na evolução de um modelo de agrupamento de dados, o interesse primordial está concentrado nas seguintes questões(HAIR *et al*,1998):

- Como são os grupos similares entre si?
- A pertinência dos registros aos grupos mais prováveis é forte ou fraca?
- Existem registros de pontos fora dos padrões?
- Quais são as características típicas dos registros pertencentes a cada grupo? (Perfil dos grupos)
- O que diferencia cada grupo dos demais?

3.2 - TIPOS DE ATRIBUTOS EM PROBLEMAS DE AGRUPAMENTO

Antes de aprofundar o estudo sobre agrupamento de dados propriamente dito, serão mostradas as diversas maneiras de como os dados podem estar representados.

3.2.1 - Atributos Numéricos

A forma mais comum de se representar objetos a serem agrupados é por meio de medidas numéricas de seus atributos como, por exemplo: distância, altura, largura, peso, quantidade, temperatura, etc. Essas medidas são naturalmente contínuas. Dessa forma cada objeto é representado por um vetor de dimensão igual ao número de atributos a serem considerados e cada elemento do vetor expressa numericamente a medida de cada atributo.

3.2.1.1 - Normalização

A escolha das unidades que expressam os atributos tem influência no peso que cada atributo terá no processo de agrupamento. Quanto menor for a unidade escolhida, maior será a faixa de valores e conseqüentemente maior a influência daquele atributo no agrupamento.

Quando se pretende que os atributos tenham influências equivalentes no agrupamento é preciso que as medidas sejam normalizadas. Isto pode ser feito da forma descrita na equação 3.1. (HAN & KAMBER,2006)

$$Z_i = \frac{(X_i - X \text{ min})}{(X \text{ max} - X \text{ min})} \quad (3.1)$$

Onde:

X_i : Valor não normalizado do atributo do elemento i ;

X_{min} : Valor mínimo do atributo dentre todos os elementos;

X_{max} : Valor máximo do atributo dentre todos os elementos;

Z_i : Valor normalizado do atributo do elemento i .

3.2.1.2 - Medidas de Distância

Para atributos numéricos, as medidas de distância mais comuns são feitas através da:

Distância Euclidiana:

$$d(i, j) = \sqrt{(X_{i1} - X_{j1})^2 + (X_{i2} - X_{j2})^2 + \dots + (X_{in} - X_{jn})^2} \quad (3.2)$$

Distância de Manhattan ou 'city-block':

$$d(i, j) = |X_{i1} - X_{j1}| + |X_{i2} - X_{j2}| + \dots + |X_{in} - X_{jn}| \quad (3.3)$$

A distância de Manhattan se aproxima da distância Euclidiana quando apenas um atributo diferencia dois objetos. Quando mais de um atributo diferencia dois objetos,

a distância de Manhattan tende a separar mais os objetos do que a distância Euclidiana separaria. (HAN & KAMBER,2006)

3.2.2 - Atributos Binários

A representação dos atributos de um conjunto de elementos a ser agrupado pode ser representado através de variáveis binárias. Por exemplo, a ficha médica de um grupo de indivíduos a serem agrupados em níveis de risco de problemas cardíovasculares poderia conter os seguintes atributos binários: sexo (m ou f) pratica exercícios físicos regularmente (sim ou não), sofre de pressão alta (sim ou não), dieta equilibrada (sim ou não), tem histórico de problemas cardíovasculares na família (sim ou não), sofre de obesidade (sim ou não), fumante (sim ou não), possui nível de colesterol acima de 240 (sim ou não), etc. Embora algumas características pudessem ter atributos mais flexíveis (sim, um pouco, não), os atributos binários apresentam a vantagem de serem mais fáceis e rápidos de serem coletados e muitas vezes é dessa forma que os dados estão disponíveis.

As variáveis binárias podem ser de dois tipos: simétricas ou assimétricas. As variáveis simétricas são aquelas cujos dois estados influenciam igualmente no agrupamento; já as variáveis assimétricas são variáveis cujos dois estados têm influência diferenciada no agrupamento. Por exemplo: Pratica exercícios físicos regularmente (sim ou não) por si só não coloca ou retira um indivíduo de um grupo de risco. Da mesma forma que o estado 'sim' afasta o indivíduo do grupo de risco, o estado 'não' o aproxima do grupo de risco. Ambos estados têm o mesmo peso no agrupamento. Já a variável: Possui nível de colesterol acima de 240, não pode ser considerada da mesma forma. O estado 'sim' tem uma influência muito mais forte na inclusão do indivíduo no grupo de risco do que o estado 'não' tem no afastamento do grupo de risco.

3.2.3 - Atributos Nominais

São aqueles cujos estados não se limitam a dois como nos atributos binários, mas podem assumir um determinado número de estados. Por exemplo, para um conjunto de automóveis, o atributo marca é um atributo nominal, pois pode assumir um entre vários estados, como por exemplo: Fiat, General Motors, Volkswagen, Ford, etc.

3.2.4 - Atributos Ordinais

Muitas vezes os objetos são representados através de atributos cujos estados guardam uma relação de ordenação entre si. Por exemplo, em um grupo de militares, o atributo patente (soldado, sargento, tenente, capitão, general) é um atributo ordinal.

3.2.5 - Atributos Mistos

Na realidade é muito comum que os objetos de um determinado conjunto a ser agrupado possuam atributos de vários tipos. Como ilustração, pode-se supor que os dados médicos coletados com intenção de se agrupar pacientes em grupos de risco de doenças cardíaco-vasculares sejam conforme a Tabela 1.

Tabela 1 - Ficha Médica dos Pacientes

Paciente	Sexo	Fumante	Nível de colesterol	Idade	Nível de atividade física
A	M	N	160	30	intenso
B	M	S	200	50	nenhuma
C	F	N	180	40	média

Os diferentes tipos de atributos utilizados no preenchimento na ficha médica são exemplificados na Tabela 2.

Tabela 2 - Tipos de Atributo

Sexo	binário simétrico
Fumante	binário assimétrico
Nível de colesterol	numérico
Idade	numérico
Nível de atividade física	ordinal

3.3 - CARACTERÍSTICAS DOS PROBLEMAS DE AGRUPAMENTO

As ferramentas de agrupamento de dados podem ser empregadas nas mais diversas áreas do conhecimento humano e, até mesmo por essa diversidade, de acordo com HAN & KAMBER (2006) devem atender aos seguintes requisitos específicos:

Escalabilidade: geralmente os algoritmos de segmentação trabalham bem em grupos de dados pequenos, cerca de 200 objetos. Entretanto, uma grande base de dados

pode conter milhões de objetos. Assim, tais algoritmos necessitam ser altamente escaláveis.

Habilidade em lidar com tipos diferentes de atributos: normalmente, os algoritmos de segmentação desenvolvidos para agrupar dados numéricos. Contudo, algumas aplicações podem requerer a segmentação de outros tipos de dados, tais como binários, categóricos (nominais), ordinais ou misturas destes tipos de dados.

Descoberta de grupos com formas arbitrárias: os algoritmos de segmentação são muitas vezes baseados em distâncias Euclidianas ou *Manhattan*. Tais algoritmos tendem a achar grupos “esféricos”, com tamanho e densidade similares. Contudo, um grupo pode assumir qualquer forma. É importante desenvolver algoritmos que possam descobrir grupos que possuam formas arbitrárias. (HAN & KAMBER 2006, HAIR *et al*,1998) .

Necessidade mínima de conhecimento do domínio para determinação dos parâmetros de entrada: em sua maioria, os algoritmos de segmentação requerem dos usuários certos parâmetros para análise de agrupamento, tais como o número desejado de grupos. Os algoritmos, assim, ficam muito sensíveis à influência de parâmetros que muitas vezes são difíceis de determinar, principalmente, em grupos de objetos com um elevado número de atributos.

Habilidade em lidar com dados “ruidosos”: muitos bancos de dados do mundo real contêm dados fora do padrão ou perdidos, desconhecidos ou errôneos. Alguns algoritmos de segmentação são sensíveis a tais dados e podem conduzir a agrupamentos de baixa qualidade.

Insensibilidade para a ordem de entrada dos registros: alguns algoritmos de segmentação podem gerar diferenças entre os grupos formados, dependendo da ordem que os dados lhes são apresentados.

Alta dimensionalidade: um banco de dados ou *data warehouse* pode conter várias dimensões-atributos. Muitos algoritmos de segmentação são bons para manuseio de objetos com baixa dimensionalidade, envolvendo somente duas a três dimensões. A capacidade humana permite julgar a qualidade de segmentos de até três dimensões. O desafio está em agrupar objetos de dados representados por um elevado número de atributos – alta dimensionalidade, principalmente considerando-se que tais dados podem ser muito esparsos e altamente assimétricos.

Segmentação baseada em restrições: aplicações do mundo real podem necessitar segmentações que obedeçam a algumas restrições. Um bom algoritmo de

segmentação deve ser capaz de descobrir os grupos ideais que atendam a essas restrições.

Interpretabilidade e usabilidade: deve ser esperado que o resultado da segmentação seja explicável, compreensível e utilizável.

A segmentação de dados, ou *clustering*, é a tarefa central para a qual muitos algoritmos têm sido propostos, (AZZAG *et al*,2004, AZZAG *et al*,2007, COELHO & EBECKEN,2001, HANDL & MEYER, 2007, LIU *et al*, 2007, NUANNUAN *et al*, 2005).

Regularmente, variantes novas de métodos mais velhos, ou novas abordagens, emergem ao mesmo tempo em que as atividades humanas demandam o estabelecimento de padrões para comparação de tais algoritmos (ANDRADE, 2004, MACHADO, 2002 MORAES, 2004, PUNTAR, 2003).

Existe uma grande diversidade de algoritmos de segmentação devido à grande variedade de princípios de indução e modelos. E os princípios indutivos existem em tão grande número porque “segmentação é em parte o olho do observador”, sendo formalizações matemáticas do que os pesquisadores acreditam ser uma definição de agrupamento. O conhecimento do domínio é que forma a crença de que existem subgrupos em meio a um grande volume de objetos e tal convicção é que molda as estruturas utilizadas para representar os grupos. Assim, diferenças nos princípios indutivos são de ordem “filosófica”, mas fundamentais. Contudo, de acordo com ESTIVILL-CASTRO, 2002 alguns mecanismos podem ser usados na comparação de princípios indutivos e, conseqüentemente, de algoritmos de segmentação, resumidos a seguir:

- os métodos de segmentação são oriundos da percepção do observador;
- diferentes pesquisadores podem determinar boas segmentações por diferentes fórmulas matemáticas;
- a segmentação traduz em otimização, problemas cuja complexidade computacional é tipicamente intratável e são solucionados por algoritmos de aproximação;
- o primeiro nível de comparação entre dois algoritmos de segmentação dá-se em termos da qualidade da solução, como o

- valor obtido pela mesma função objetivo e recursos de computação iguais (por exemplo, número de avaliações da função objetivo);
- existem objetivos funcionais (princípios indutivos) que apresentam custo menor que outros. Naturalmente, busca-se aperfeiçoar aqueles na esperança de apresentarem boas soluções com custo menor. Contudo, isto implica em uma investigação sobre o relacionamento entre os princípios indutivos e, num segundo estágio, entre os próprios algoritmos;
 - pesquisas e compilações de categorias de algoritmos de segmentação são naturalmente baseadas mais em modelos do que em princípios de indução. A mais forte distinção entre tais algoritmos está na adoção de modelos matemáticos (contínuos), comuns em inferências estatísticas, ou modelos de estruturas (discreto), comuns em aprendizado de máquina;
 - os pesquisadores que propõem modelos e princípio indutivo novos, ou algoritmos de segmentação melhorados, deveriam explicitar seus métodos matemáticos, o que facilitaria investigações e comparações prévias com métodos emergentes;
 - os índices de validade da segmentação são formulações matemáticas diretas de princípios de indução. A comparação entre os algoritmos pode fornecer algumas noções sobre o contexto no qual um algoritmo processa melhor que outro. Contudo, isto não implica que um algoritmo produz resultados mais válidos que outro. A validade depende da existência de uma estrutura do conjunto de dados;
 - a qualidade da segmentação pode ser demonstrada por critérios externos de validade e medidas associadas. Métodos de aprendizado supervisionado podem ser usados para avaliação de algoritmos, como medidas de discrepância entre os resultados da segmentação e os rótulos previamente conhecidos;
 - um algoritmo adaptado a um determinado universo de modelos não tem chance de bom desempenho se os grupos de dados têm uma estrutura que é, na verdade, representada por uma família

radicalmente diferente (por exemplo, “K-Means” não pode determinar grupos não-convexos).

São inúmeras as citações sobre algoritmos de segmentação que podem ser encontradas na literatura. A escolha do algoritmo mais adequado depende tanto do tipo dos dados disponíveis quanto do propósito específico da aplicação.

De maneira geral, a maioria dos métodos de agrupamento podem ser classificados em cinco grandes categorias (HAN & KAMBER, 2006, JAIN *et al*, 1999):

- métodos particionais;
- métodos hierárquicos;
- métodos baseados em densidade;
- métodos baseados em malhas; e
- métodos baseados em modelos.

3.3.1 - Métodos Particionais

Alguns autores como [HAN & KAMBER, 2006, JAIN *et al*, 1999] consideram que n é o número de objetos de uma base de dados e k é o número de grupos desejado. Os algoritmos particionais irão gerar k partições utilizando medidas de distância, de forma que uma determinada função objetivo seja otimizada, sendo que objetos comparados com os demais objetos dentro de sua própria partição guardem máxima similaridade e objetos comparados com os objetos de outras partições apresentem mínima similaridade.

Os métodos particionais clássicos mais conhecidos, e comumente utilizados, são “*K-Means*” e “*k-Medoids*”. O método “*K-Means*” é baseado no conceito estatístico do centróide de forma que a semelhança dos objetos dentro do mesmo grupo, “intragrupo”, seja alta e a semelhança entre objetos de grupos diferentes, “intergrupo”, seja baixa, sendo, porém, muito sensível a dados fora do padrão, ou seja, objetos que tenham um valor extremamente alto ou baixo, que podem causar substanciais distorções nos resultados.

O método “*K-Medoids*”, em contrapartida, busca diminuir a sensibilidade a objetos isolados, deixando de levar em conta o centro do grupo como ponto de referência para utilizar o medóide, que é o objeto localizado mais próximo ao centro do grupo. Assim, este método pode abrandar a influência de objetos isolados, baseando-se

no princípio de minimização da dissimilaridade entre os objetos de um grupo e o seu ponto de referência.

A estratégia básica do algoritmo de segmentação “*K-Medoids*” é achar k grupos em n objetos; primeiramente, arbitrando para cada grupo um objeto como seu representante (o medóide). Cada objeto remanescente é agrupado com o medóide ao qual possui a maior similaridade. Então, iterativamente, é substituído um medóide por um não-medóide, desde que a qualidade da segmentação resultante seja melhor. Esta qualidade é calculada usando uma função de custo que calcula a medida de dissimilaridade média entre um objeto e o medóide de seu grupo.

Numa comparação entre os dois métodos particionais clássicos, o método “*K-Medoids*” pode ser apontado como mais robusto que “*K-Means*” por trabalhar com mais eficiência na presença de dados isolados e outros ruídos, em decorrência do medóide sofrer menos influência de dados desse tipo. Contudo, seu processamento tem custo maior que o método “*K-Means*”. Mas, sempre vale ressaltar, ambos os métodos requerem a prévia especificação de k , o número de grupos.

3.3.2 - Métodos Hierárquicos

Um método hierárquico de segmentação trabalha agrupando objetos em uma árvore de grupos. A qualidade dos métodos hierárquicos puros pode ser prejudicada pela sua inabilidade em executar ajustes após uma fusão ou divisão terem sido executadas. Os métodos hierárquicos podem ser classificados de acordo com a forma com que a decomposição hierárquica é realizada: *bottom-up* ou *top-down*, e são conhecidos como:

- Segmentação hierárquica aglomerativa: esta estratégia *bottom-up* inicia por considerar cada objeto como sendo seu próprio grupo e, então, vai criando sucessivas fusões destes grupos “atômicos” em conjuntos maiores, até que todos os objetos estejam em um único grupo ou até que certas condições de conclusão sejam satisfeitas. Os métodos hierárquicos, em sua maioria, pertencem a esta categoria, diferindo entre si apenas pela definição da similaridade “intergrupo”.
- Segmentação hierárquica divisiva: esta estratégia *top-down* faz o contrário da segmentação hierárquica aglomerativa, começando com todos os objetos num único grupo. A partir daí os grupos são

subdivididos em conjuntos menores, até que cada objeto forme o próprio grupo; até que o algoritmo satisfaça determinadas condições de conclusão, tais como a obtenção de um número desejado de grupos; ou que a distância entre dois grupos próximos ultrapasse um certo limiar.

A Figura 4 apresenta, de maneira esquemática, os passos da segmentação aglomerativa e divisiva.

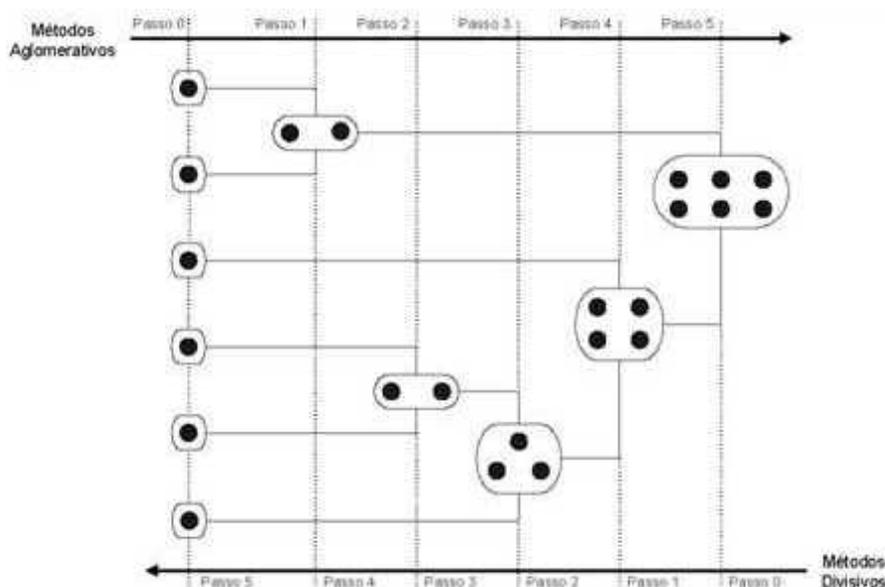


Figura 4 - Segmentação Aglomerativos e Divisivos(HAN & KAMBLER,2006)

Os métodos de segmentação hierárquicos, embora simples, freqüentemente encontram dificuldades no que diz respeito à seleção de pontos de fusão ou de divisão. Tal decisão é crítica, pois uma vez que um grupo de objetos é fundido ou dividido, o próximo passo será processado com base no grupo recém gerado. Considerando-se que tais algoritmos não desfazem o que já foi processado, tampouco trocam objetos entre grupos, se as fusões ou divisões não são bem feitas em algum passo, a qualidade do resultado final da segmentação pode ficar seriamente comprometida. Além disso, o método não possui critérios para examinar e avaliar se a fusão ou divisão redundante em um número ideal de objetos ou grupos. (HAN & KAMBER, 2006, JAIN *et al*, 1999)

3.3.3 - Métodos Baseados em Densidade

Devido à falta de habilidade que os algoritmos baseados em métodos particionais têm em descobrir grupos com formas arbitrárias, isto é, grupos que apresentam objetos dispersos em uma distribuição espacial bastante irregular, foram desenvolvidos métodos de segmentação baseados em densidade. Estes métodos

consideram tipicamente os grupos como regiões densas de objetos separados no espaço por regiões de baixa densidade – que, normalmente, representam algum tipo de ruído.

Alguns algoritmos de implementação dos métodos de segmentação baseados em densidade podem ser especificamente citados, como abaixo:

DBSCAN (*Density-Based Spatial Clustering of Application with Noise*): é um algoritmo baseado em conexões de regiões com suficientemente alta densidade que emprega a estratégia de estabelecer tais regiões como grupos. O DBSCAN tem a capacidade de descobrir grupos com formas arbitrárias em espaços de dados, inclusive com presença de ruídos, definindo os grupos como a combinação máxima de pontos densamente conectados.

Para realizar a segmentação, o DBSCAN verifica a vizinhança próxima de cada objeto dentro do banco de dados. Se a vizinhança de um objeto p contém ao menos o número mínimo de ocorrências, então p será considerado como o núcleo (ou centro) de um novo grupo a ser criado em seu entorno, chamado de vizinhança próxima. O raio de definição da vizinhança - ϵ , bem como o número mínimo de ocorrências, MinPts, são parâmetros prédefinidos. A partir daí o algoritmo irá agrupar, iterativamente, os objetos com “densidade de alcance” direta em relação ao núcleo dos grupos, ou seja, são agrupados os objetos mais próximos ao núcleo, de maneira que o grupo mantenha alta densidade. Este processo pode envolver a fusão de grupos com pouca densidade e termina quando nenhum ponto novo pode ser incluído em qualquer grupo. Todo objeto não contido em qualquer um dos grupos formados será considerado como ruído.

OPTICS (*Ordering Points To Identify the Clustering Structure*): método que emprega a identificação da estrutura dos grupos a partir da ordenação de seus objetos, OPTICS apresenta uma importante vantagem em relação ao método apresentado anteriormente. Como visto, o algoritmo DBSCAN depende da introdução de parâmetros, como o raio de definição da vizinhança e o número mínimo de ocorrências para realizar a segmentação de objetos, o que deixa com o usuário a responsabilidade de definir valores de entrada adequados para a formação de agrupamentos satisfatórios. Na verdade, este é um problema associado a muitos outros algoritmos de segmentação. Tais parâmetros são usualmente difíceis de ponderar e determinados de maneira empírica, principalmente para dados do mundo real com muitas dimensões. Naturalmente, muitos algoritmos são bastante sensíveis aos valores de parâmetros. Por exemplo, combinações de parâmetros de entrada, apenas levemente diferentes, podem conduzir à formação de agrupamentos muito discrepantes em relação aos dados analisados. Além disso,

coleções de dados reais com muitas dimensões têm frequentemente uma inclinação de distribuição induzida por sua estrutura intrínseca, que não pode ser caracterizada por parâmetros de densidade globais.

O algoritmo OPTICS foi desenvolvido justamente procurando superar a fragilidade causada pela dependência dos parâmetros de entrada. Ao invés de formar agrupamentos explícitos, este algoritmo faz o cômputo da “ordenação aumentada de agrupamentos” para análise interativa e automática. Esta ordenação representa a estrutura dos dados agrupados por densidade e contém informações que são equivalentes às obtidas em um largo espectro de parâmetros de entrada.

No algoritmo DBSCAN, a constante MinPts diz respeito ao grau de densidade dos grupos formados, e ϵ é um parâmetro de distância que representa o raio da vizinhança mais próxima ao núcleo. Portanto, a fim de criar uma ordenação aumentada de agrupamentos, pode-se estender o DBSCAN para processar um conjunto de parâmetros de distância ao mesmo tempo. Para construir diferentes segmentações simultaneamente, os objetos devem ser processados em uma ordem específica. Esta ordem seleciona os objetos dentro da densidade de alcance da vizinhança com menor raio, de forma que os grupos com maior densidade sejam completados primeiro. Com base nesta idéia, dois valores precisam ser armazenados para cada objeto:

- a distância de núcleo é o mínimo valor de ϵ que faz de p um objeto núcleo. Se p não é um objeto núcleo, a distância de núcleo fica indefinida;
- a distância de alcance de um objeto q em relação a outro objeto p é definida como o maior valor entre a distância de núcleo de p e a distância Euclidiana entre p e q . Se p não é um objeto núcleo, a distância de alcance fica indefinida.

Em seu processamento, o algoritmo OPTICS cria uma ordenação dos objetos, armazenando adicionalmente a distância de núcleo e a distância de alcance mais satisfatória para cada objeto, que serão posteriormente usadas para extrair os grupos baseados em densidade.

DENCLUE (*DENSity-based CLUstEring*): método baseado em um conjunto de funções de distribuição de densidade, constituído das seguintes idéias:

1. a influência de cada objeto de uma base de dados pode ser formalmente modelada usando uma função matemática, chamada função de influência, a

- qual descreve o impacto de um ponto de dados dentro de sua vizinhança;
2. a densidade global do espaço de dados pode ser modelada analiticamente como a soma das funções de influência de todos os objetos;
 3. os grupos podem ser determinados matematicamente identificando-se a densidade de atração, que é o máximo local da função de densidade global.

Em comparação com outros algoritmos de segmentação, existem algumas significativas vantagens apresentadas pelo DENCLUE: forte fundamentação matemática e generalização de outros métodos de segmentação; boa capacidade de segmentação em coleções de dados com elevada presença de ruídos; possibilidade de descrição matemática para formas arbitrárias de grupos de dados com muitas dimensões; utilização de malhas que contêm somente informações sobre a estrutura das células, sendo assim significativamente mais rápidos que alguns algoritmos mais influentes como o DBSCAN. (HAN & KAMBER, 2006, JAIN *et al*, 1999)

3.3.4 - Métodos Baseados em Malhas

A abordagem utilizada pelos métodos baseados em malhas utiliza-se de uma malha de estrutura de dados com múltipla resolução. Dessa forma, o espaço de estudo é quantificado em um número finito de células que formam uma malha sobre a qual todas as operações de segmentação serão realizadas. A vantagem principal desta abordagem é seu rápido processamento, pois não é diretamente dependente do número de objetos a serem analisados e sim do número de células em cada dimensão do espaço quantificado.

Alguns exemplos típicos dos métodos baseados em malhas são o STING, que explora informações estatísticas armazenadas nas células da malhas; WAVECLUSTER, que agrupa os objetos usando o método de transformada e CLIQUE, que apresenta uma combinação entre métodos baseados em malhas e baseados em densidade para segmentação de bases de dados com alto número de dimensões. Estes exemplos são melhor explicados a seguir. (HAN & KAMBER, 2006, JAIN *et al*, 1999)

STING (*Statistical Information Grid*): técnica baseada em malhas de múltipla resolução, na qual o espaço a ser analisado é dividido em células retangulares. Normalmente, existem diversos níveis de células, correspondendo a níveis diferentes de resolução que formam uma estrutura hierárquica: cada célula de um nível qualquer é particionada em um certo número de células que formam o nível imediatamente inferior. Informações estatísticas com respeito aos atributos em cada célula, como

média, máximo e mínimo, são computadas e armazenadas. (HAN & KAMBER, 2006, JAIN *et al*, 1999)

O *WAVECLUSTER* é um algoritmo baseado em malhas e em densidade, que atende muitos requerimentos de um bom algoritmo de segmentação: manipula grandes coleções de dados eficazmente; descobre grupos com formas arbitrárias; processa dados fora do padrão com sucesso; é insensível para a ordem de introdução dos dados e não exige a especificação de parâmetros como o número desejado de grupos ou raio de vizinhança. (HAN & KAMBER, 2006, JAIN *et al*, 1999)

CLIQUE (CLustering In QUEst): algoritmo de segmentação que integra métodos de malha e densidade. É de grande utilidade para segmentar dados de muitas dimensões em banco de dados extensos, baseado nas seguintes características:

- em uma grande base de dados multidimensionais, os objetos normalmente não se distribuem uniformemente pelo espaço. A segmentação de *CLIQUE* identifica as áreas esparsas e as áreas densas (unidades), determinando assim os padrões de distribuição globais da base de dados.
- uma unidade é considerada densa se a fração de objetos nela contidos excederem um parâmetro fornecido. Um grupo é definido como a combinação máxima de unidades densas conectadas.

3.3.5 - Métodos Baseados em Modelos

Os métodos baseados em modelos criam um modelo hipotético para cada grupo desejado e procuram ajustar os dados da melhor maneira ao modelo criado. Os algoritmos baseados neste método são capazes de descobrir os grupos por meio de ações de densidade que refletem a distribuição espacial dos objetos.

Como tentativa de otimização dos métodos de segmentação, com uma melhor adequação entre os dados apresentados e alguns modelos matemáticos, tais métodos são usualmente baseados na hipótese de que os dados são originados de acordo com uma probabilidade estatística de distribuição. (HAN & KAMBER, 2006).

Os métodos de modelos seguem três principais abordagens: a abordagem estatística, a de *machine learning* e a abordagem de redes neurais.

A abordagem estatística é baseada na hipótese de que os dados são originados de acordo com uma probabilidade estatística de distribuição e possibilitam modos de

determinar automaticamente o número de grupos baseados em padrões estatísticos, inclusive em presença de ruídos ou dados fora do padrão, o que proporciona o desenvolvimento de algoritmos bastante robustos.

A abordagem de *machine learning* é um tipo de segmentação que utiliza aprendizado de máquina, também conhecida como segmentação conceitual. Dada uma coleção de objetos sem rótulo, este método produz um esquema de classificação que vai além dos dados. Ao contrário de métodos de segmentação convencionais, os quais identificam basicamente grupos de objetos afins, a segmentação conceitual descobre também descrições das características de cada grupo, onde os grupos representam um conceito ou classe. Neste método, a avaliação da qualidade da segmentação não é somente uma função dos objetos individuais, ela passa por fatores como a generalidade e a simplicidade derivadas da descrição dos conceitos. (HAN & KAMBER, 2006).

Como exemplo desta abordagem, pode-se citar o COBWEB, que é um método popular e simples de segmentação conceitual. Nele, os objetos introduzidos são descritos por pares de atributos categóricos e a segmentação é hierárquica, em forma de uma árvore de classificação, como pode ser visto na Figura 5.

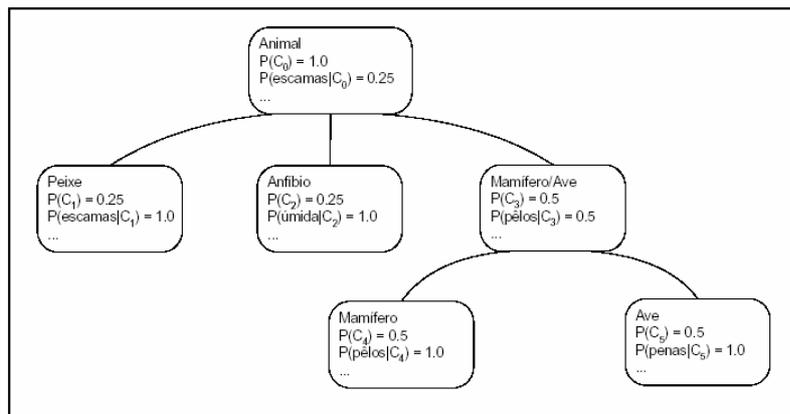


Figura 5 - Árvore de Classificação

Enquanto percorre o melhor caminho pela árvore em busca do melhor nó hospedeiro ao qual classificar o objeto, COBWEB aloca temporariamente o objeto em cada nó e calcula a “utilidade de categoria” da partição resultante. O arranjo que resultou na mais alta “utilidade de categoria” é indicado como hospedeiro ideal para o objeto. Se um objeto não pertencer a quaisquer dos conceitos já representados na árvore, uma nova classe é criada e automaticamente é ajustado o número de classes existentes em uma partição, dispensando que este seja um parâmetro a ser introduzido pelo usuário.

A abordagem de redes neurais busca representar cada grupo como um exemplar. Um exemplar atua como protótipo do grupo e não tem necessariamente que corresponder a um objeto específico. Novos objetos podem ser alocados ao grupo quando o seu exemplar apresentar a maior semelhança baseada em alguma medida de distância.

Dois dos mais proeminentes métodos de abordagem de redes neurais são o aprendizado competitivo e os mapas auto-organizáveis de características, ambos envolvendo competição entre as unidades neurais. (HAN & KAMBER, 2006)

3.3.5.1 - Algoritmo *Expectation Maximization*

O algoritmo *Expectation Maximization* (EM) segue a abordagem estatística dos Métodos de Agrupamento de dados Baseado em Modelos, seguindo a hipótese de que os dados são agrupados de acordo com uma distribuição estatística.

Fazendo uma comparação com o algoritmo K-means pode-se dizer que este constrói um *hard clustering* onde cada ponto pertence somente a um cluster e o algoritmo EM constrói um *soft clustering*, onde cada ponto tem uma probabilidade de pertencer a cada cluster.

O algoritmo EM supõe que os dados são gerados a partir de uma mistura de curvas gaussianas e tenta encontrar as curvas mais prováveis.

O caso mais simples são as gaussianas circulares com a mesma variância e o mais complexo são as gaussianas elípticas com matrizes de covariância diferentes. (BISHOP, 2006, ZADROZNY, 2006)

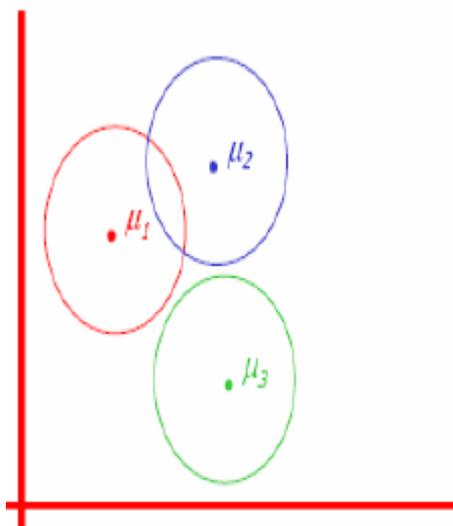


Figura 6 - Gaussianas Circulares (ZADROZNY, 2006)

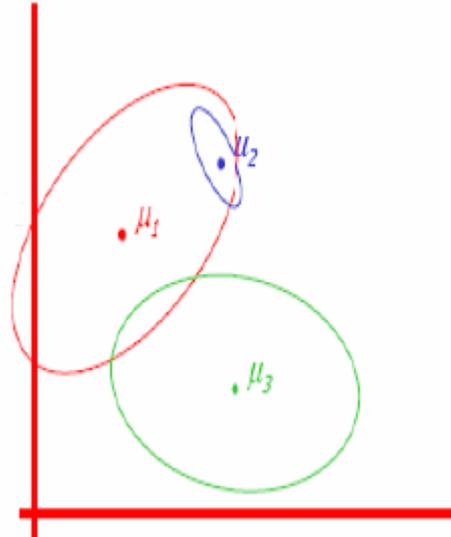


Figura 7 - Gaussianas Elípticas (ZADROZNY, 2006)

O algoritmo tem uma estrutura parecida com a do K-means:

- Escolher valores iniciais aleatórios para os vetores de média μ_i ;
- Passo Expectation: calcular a probabilidade de cada vetor de média μ_i dados os pontos; e
- Passo Maximization: encontrar os vetores de média μ_i mais prováveis.

3.4 - NOVAS TENDÊNCIAS

Os algoritmos apresentados são algoritmos que levam em conta que a clusterização é uma tarefa não supervisionada, o que sempre gera dificuldades para definir o número de clusters existentes nas estruturas dos dados.

Recentemente, o uso de informações privilegiadas ou restrições conhecidas, têm sido utilizadas durante a tarefa de clusterização. Ou seja, em alguns casos existem amostras que necessariamente pertencem a determinados clusters, e estas informações auxiliam ou interferem no processo de clusterização, denominando este processo de aprendizado semi-supervisionado. Isso é particularmente importante na área de mineração de textos (DA SILVA, 2006).

A Figura 8 apresenta um resumo dos métodos de agrupamento com os principais algoritmos, com destaque para o método semi-supervisionado acrescentado à direita em linha tracejada.

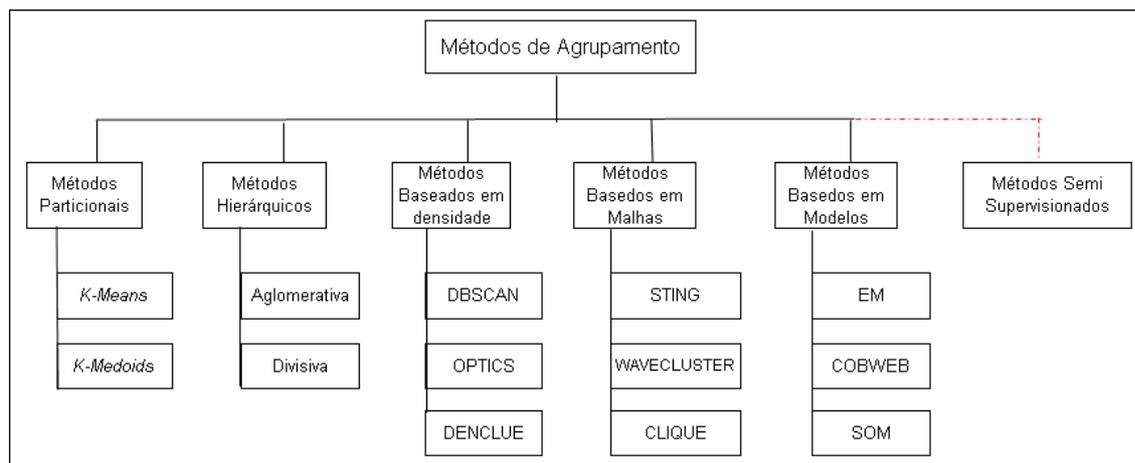


Figura 8 - Métodos de Agrupamentos

3.5 - O PROBLEMA DA DEFINIÇÃO DO NÚMERO DE GRUPOS

A maior desvantagem do método ou dos métodos clássicos “*K-Means*” certamente é a exigência do valor de k como parâmetro inicial. Para uma aplicação real, em grandes bases de dados, esta informação normalmente é desconhecida, podendo ser necessária uma prévia análise de um especialista para que houvesse possibilidade de percepção da quantidade de grupos em que a coleção poderia dividir-se de forma satisfatória, o que por si só já representa uma grande carga de trabalho que, até mesmo, diante de uma base muito grande poderia tornar a análise inviável ou desinteressante.

Existem vários critérios para a determinação do número de grupos e quase todos funcionam da seguinte maneira: realizar o agrupamento dos dados considerando 2 grupos e calcular o valor de uma função proposta que tenha o número de grupos como um de seus parâmetros, realizar o agrupamento dos dados considerando 3 grupos e calcular o valor da mesma função, repetir este procedimento até atingir um número máximo de grupos estabelecido.

Há diversos índices semi-empíricos que podemos usar, tais como, Calinski e Harabasz, Critério Condorcet, *Cubic Clustering Criterion* e PBM, para ver qual o melhor “ n ”, número de grupos. Este tipo de classificação não supervisionada encontra vasta literatura e tem sido tema de pesquisa ininterrupta. (ANDRADE, 2004, MACHADO, 2002, MORAES, 2004, PUNTAR, 2003).

O agrupamento que ocasionar o valor máximo (ou, em alguns casos, mínimo) da função, deve ser considerado como o melhor agrupamento possível para a base de dados.

O agrupamento de dados continua sendo uma tarefa desafiadora, pois ainda não se encontrou uma resposta definitiva para este problema. Está sempre atraindo os pesquisadores em busca de algoritmos inspirados em novas teorias, tais como sistema imunológico artificiais, colônia de formigas, *flock agents*, algoritmos genéticos, algoritmos *swarm* (AZZAG *et al*,2004, AZZAG *et al*,2007, COELHO& EBECKEN,2001, HANDL & MEYER, 2007, LIU *et al*, 2007 NUANNUAN *et al*, 2005).

A Tabela 3 apresenta a complexidade de alguns algoritmos (HANDL, 2003, NUANNUAN, 2005) e suas principais características (HAN &KAMBLER, 2006)

Tabela 3 - Características dos algoritmos

Algoritmos	Complexidade	Características
K-Means	$O(N)$	É sensível a ruídos e valores aberrantes, freqüentemente termina em ótimo local
Hierárquicos	$O(N^2)$	Não requer o n ^o de <i>k</i> grupos como entrada, não são escaláveis
EM	$O(N^2)$	Não trabalha muito bem com clusters de estruturas complexas, não requer o n de <i>k</i> grupos como entrada
Ant Colony	$O(N)$	Não necessita de informações a priori da base de dados, depende do número de ciclos e de formigas

4 - ALGORITMOS EVOLUTIVOS

4.1 - INTRODUÇÃO

A maioria dos problemas computacionais encontrados no mundo real não possui uma solução definida. Isto ocorre quando o problema é apenas parcialmente conhecido, quando não se possui informação suficiente sobre seu domínio para processá-lo; ou mesmo em situações em que não se detém tempo suficiente para resolvê-lo de maneira satisfatória.

Curiosamente, organismos vivos têm encarado estes tipos de problemas durante milhões de anos. Seus desafios são de alta complexidade e entre os vários, pode-se citar, por exemplo, os problemas de encontrar comida ou parceiros para a reprodução, ao mesmo tempo em que se evitam predadores e situações perigosas. A sobrevivência do nicho ecológico de uma determinada espécie depende, portanto, da habilidade dos organismos em solucionar problemas complexos. Dessa maneira, a permanência de uma classe de indivíduos se relaciona à capacidade de que os espécimes da classe atuem de acordo com padrões de comportamento préestabelecidos. Os organismos, desta forma, são capazes de preencher certos “buracos” de informação encontrados em problemas complexos. Eles podem “supor” que certas variáveis irão se comportar de acordo com um padrão, ou simplesmente fornecer a informação ausente. Mas que tipo de padrão de comportamento deve ser respeitado? Como os organismos vivos são capazes de reduzir um conjunto de centenas de ações a um pequeno subconjunto de ações práticas?

Wilson (1999) concebeu um modelo biológico de criação de um comportamento inato. Estes comportamentos seriam herdados geneticamente como resultado do processo de seleção natural. Estes atuam como um comportamento cognitivo para otimizar ou adequar possíveis ações a uma determinada situação. O autor definiu estes comportamentos inatos como regras epigenéticas.

Computação evolutiva é o nome dado à técnica de resolução de problemas computacionais em que utilizam-se modelos inspirados evolução natural. A maioria dos algoritmos evolutivos desenvolvidos seguem paradigmas biológicos, bem como conceitos de seleção natural, mutação e reprodução.

O homem em sua busca secular pela compreensão e estudo dos fenômenos da natureza, tem criado modelos matemáticos simplificados da realidade, procurando

reproduzir os fenômenos observados. Apesar dos modelos serem apenas uma simplificação da realidade, freqüentemente são utilizados como base para a resolução de problemas em várias áreas do conhecimento.

Nas últimas décadas tem-se observado uma inversão nesta busca do conhecimento. Em face da complexidade cada vez maior dos problemas apresentados e da dificuldade de se chegar a uma solução puramente matemática, o homem tem buscado modelar os problemas usando exemplos observados na natureza, tais como a evolução das espécies, o comportamento de “bando de pássaros” e as colônias de formigas. Esta nova área do conhecimento que está nascendo é chamada de Algoritmos Evolutivos ou Evolucionários (AE).

A modelagem matemática de problemas complexos pode levar a funções bastante complicadas: multidimensionais, não lineares, descontínuas, multivariáveis entre outros, cuja otimização pode conduzir às soluções em tempo não polinomial, o que inviabilizaria a obtenção de resultados em um tempo razoável.

Uma forma de contornar esta situação seria utilizando os Algoritmos Evolutivos tais como : Algoritmos Genéticos, Sistemas Imunológicos Artificiais, *Particle Swarm Optimization* (PSO) e *Ant Colony Optimization* (ACO), especialmente úteis às tarefas de otimização global e busca, onde algoritmos determinísticos podem levar a soluções de mínimos locais.

O ciclo básico em um AE pode ser sintetizado nos seguintes passos: (GOLDENBERG,1989)

- (i) operar em uma população de pontos;
- (ii) não requerer cálculos de derivadas e informação sobre o gradiente da função objetivo;
- (iii) trabalhar com a codificação de seu conjunto de parâmetros, não com os próprios parâmetros (representação binária);
- (iv) realizar transições probabilísticas, em vez de regras determinísticas;
- (v) necessitar apenas da informação sobre o valor da função objetivo para cada indivíduo da população;
- (vi) apresentar simplicidade conceitual; e
- (vii) ser pouco afetado, quanto à eficiência, quando descontinuidades e ruídos estão presentes nos dados do problema.

As características (iii) a (v) não são comuns a todos os AE, mas geralmente presentes nos algoritmos genéticos.

Estes algoritmos buscam uma solução ótima em um determinado espaço de busca irregular, complexo e multidimensional, a partir de um conjunto de soluções tentativas que são testadas e comparadas entre si. Em função dessa comparação, as soluções tentativas são submetidas a um processo de evolução, geralmente baseado em uma analogia com algum processo natural. Este processo é orientado a preservar as características das soluções melhores e a descartar as características das soluções piores, produzindo assim um novo conjunto de soluções tentativas que teoricamente estarão mais próximas da solução ótima procurada.

O processo evolui até que a melhor das soluções tentativas atinja um determinado patamar de desempenho ou simplesmente até que um determinado número de evoluções tenha ocorrido. Evidentemente, algoritmos evolutivos podem não encontrar a solução ótima, ficando então em uma solução subótima, principalmente se todas as soluções tentativas convergirem ao longo do processo para essa solução subótima impedindo assim ou dificultando que a busca continue a explorar o domínio.

Para minimizar a possibilidade de isto ocorrer, o processo evolutivo a que são submetidas as soluções tentativas devem possuir certo grau de aleatoriedade na geração das novas soluções tentativas de modo a permitir o surgimento em pequena escala de características não presentes nas soluções tentativas originais. O processo natural, ao qual o algoritmo evolutivo é análogo em geral, possui também esse pequeno grau de aleatoriedade em sua evolução.

Os AE apresentam vantagens e desvantagens em relação aos métodos tradicionais de busca e otimização.(COELHO, 2003)

Entre as vantagens dos AE tem-se:

- (i) não existe a necessidade de assumir-se características do espaço do problema;
- (ii) vastamente aplicável (algoritmos de propósito geral);
- (iii) baixo custo de desenvolvimento e aplicação;
- (iv) facilidade de incorporar outros métodos; e
- (v) pode ser executado interativamente e possibilita a acomodação de soluções propostas pelo usuário no procedimento de otimização.

Entre as desvantagens dos AE deve-se mencionar que:

- (i) não garantem uma solução ótima;
- (ii) podem necessitar de sintonia de alguns parâmetros inerentes à metodologia evolutiva adotada;e

- (iii) tratam-se de métodos estocásticos e seu desempenho varia de execução para execução (a menos que o mesmo gerador de números aleatórios com a mesma semente seja utilizado).

4.2 - COLÔNIA DE FORMIGAS

Insetos que vivem em colônias, tais como as formigas, abelhas, vespas e cupins, têm suas próprias tarefas independentes uns dos outros. Entretanto, quando estes insetos atuam em conjunto (como uma comunidade), eles são capazes de solucionar problemas complexos de seus cotidianos através de cooperação mútua. Problemas como selecionar e coletar materiais, encontrar e estocar alimento, que requerem planejamentos sofisticados, são resolvidos por colônias de insetos sem nenhum tipo de supervisor ou controlador.

Este comportamento coletivo que surge em um grupo de insetos sociais tem sido chamado de “Inteligência de Enxames — Swarm Intelligence” (BONABEAU *et al*,1999). Alguns comportamentos observados e estudados atualmente em insetos sociais são:

- a forma como se organizam para agrupar e ordenar larvas e corpos;
- a maneira como constroem seus ninhos e colméias;
- a maneira como se organizam para procurar alimentos;
- como é feita a divisão de trabalho no ambiente em que vivem; e
- como se organizam para transportar materiais.

Uma colônia de insetos é um sistema descentralizado que soluciona problemas de grandes proporções, como citado anteriormente, fazendo-se uso de entidades muito simples que interagem entre si. As principais características destes sistemas são a flexibilidade e a robustez com os quais solucionam os problemas: a flexibilidade permite a adaptação da colônia às mudanças do ambiente, enquanto que a robustez permite que a tarefa em andamento seja concluída mesmo com uma possível falha de alguns indivíduos.

Um dos mecanismos que dá suporte no entendimento da cooperação entre os insetos está determinado através das diferenças físicas entre os indivíduos. Por exemplo, em espécies polimórficas de formigas, a divisão de trabalho pode ocorrer de forma que dois ou mais tipos físicos diferentes de operários coexistam na mesma colônia. Em algumas espécies, operárias mais jovens são menores e morfologicamente diferentes dos operários adultos e realizam tarefas diferentes: enquanto que os adultos,

com mandíbulas muito mais desenvolvidas, cortam pedaços de folhas ou defendem o ninho, as operárias jovens alimentam as larvas ou limpam o ambiente. Na ausência ou insuficiência de operárias jovens, os adultos são estimulados a realizar as tarefas usualmente cumpridas pelas mais jovens, demonstrando um alto grau de plasticidade na divisão de trabalho dentro da colônia. Muitas das atividades coletivas realizadas por insetos sociais são auto-organizáveis não necessitando de diferenças morfológicas entre os indivíduos. Estas atividades se caracterizam por interações a nível microscópico (através de substâncias químicas por exemplo) em muitas espécies de formigas e que fazem surgir padrões macroscópicos como, por exemplo, a definição das rotas entre fontes de alimentos e o ninho (PARPINELLI, 2001).

Um sistema auto-organizável é um sistema que possui mecanismos que condicionam o surgimento de padrões a nível global por meio de interações entre os componentes que o constituem. Estes componentes interagem entre si tendo como base unicamente uma informação local, sem nenhuma referência ao padrão global a ser encontrado (solução do problema). Como, por exemplo, o comportamento global (melhor rota) das formigas que buscam por alimento e que emerge a partir de trilhas de feromônio (substância química utilizada como meio de comunicação entre indivíduos de uma mesma espécie). Sob auto-organização se definem três características básicas (BONABEAU *et al*, 1999):

- Realimentação positiva: que promove a criação de padrões/soluções. Um exemplo de realimentação positiva é o recrutamento para busca de alimento, que pode ser na forma de trilhas de feromônio em algumas espécies de formigas ou danças em colônias de abelhas;
- Realimentação negativa: a qual contrabalança a realimentação positiva, ajudando a estabilizar o padrão coletivo sendo buscado. A realimentação negativa pode ser na forma de saturação, exaustão ou competição. No exemplo da busca por alimento em colônias de formigas, a realimentação negativa pode se originar por meio da exaustão da fonte de alimento, competição entre fontes de alimentos ou evaporação das trilhas de feromônio; e
- Flutuações: as quais são de grande importância pois tornam o sistema mais hábil para descobrir novas soluções (melhor exploração do espaço de busca).

Para que uma colônia de insetos se auto-organize na realização de determinada tarefa, é necessário que haja troca de informação entre eles: tal troca de informação pode ser por via direta ou indireta. Trocas diretas ocorrem na forma de contatos por meio das antenas, troca de comida ou líquido, contato mandibular, contato visual, etc. As indiretas são mais sutis: dois indivíduos interagem indiretamente quando um deles modifica o ambiente e o outro responde às novas características do ambiente em um instante de tempo seguinte. Tais interações indiretas são chamadas de estigmergia (um comportamento individual modifica o ambiente, o qual modifica o comportamento de outros indivíduos) (BONABEAU *et al*, 1999, DORIGO *et al*, 1991, DORIGO *e tal* 1999, DORIGO & STÜTZLE, 2004, PARPINELLI, 2001).

A estigmergia está diretamente associada com o fator de flexibilidade da colônia. Isto porque, quando o ambiente muda devido a alguma perturbação externa, os insetos respondem naturalmente a esta perturbação, como se fosse uma modificação causada pela atividade da colônia. Ou seja, a colônia coletivamente responde a perturbações externas com indivíduos exibindo o mesmo comportamento.

Alguns exemplos resultantes da combinação de estigmergia com auto-organização podem ser vistos em BONABEAU *et al* (1999): no recrutamento de formigas, aonde as trilhas de feromônio deixadas pelas formigas são uma forma de modificar o ambiente para se comunicar com as outras formigas que seguem estas trilhas; modificando o ambiente na limpeza do ninho, onde a resposta dos outros indivíduos às novas características do ambiente pode ser, não no engajamento de novos operários para a limpeza, estando livres para desempenhar outras tarefas; e na engenharia que surge nas construções dos ninhos e colméias. Em todos estes casos o ambiente serve como meio de comunicação.

Dentre os comportamentos observados em insetos sociais, a busca por alimento é o que possui maior aplicabilidade em problemas do mundo real. Na seção seguinte será mostrado com maiores detalhes como surge este comportamento em colônias de formigas reais e na seqüência é detalhado como colônias de formigas artificiais fazem uso deste comportamento para aplicações computacionais.

4.3 - FORMIGAS REAIS

Formigas são capazes de encontrar a rota mais curta entre uma fonte de alimento e o seu ninho sem o uso de informações visuais, sendo capazes também de se adaptar a mudanças no meio (DORIGO *et al*, 1996, DORIGO & STÜTZLE, 2004).

Um dos principais problemas estudados pelos entomologistas é compreender como animais tão primitivos, como as formigas, conseguem encontrar o caminho mais curto entre sua colônia e uma fonte de alimento. Foi descoberto que, para trocar informação sobre qual caminho deve ser seguido, as formigas se comunicam umas com as outras por meio de trilhas de feromônio. O movimento das formigas deixa uma certa quantidade de feromônio no chão, marcando o caminho com uma trilha desta substância. O comportamento coletivo que emerge é uma forma de processo autocatalítico que, quanto mais formigas seguirem uma trilha, mais atrativa esta trilha se tornará para ser seguida por outros indivíduos. Este processo pode ser descrito como um laço de realimentação positiva, onde a probabilidade de uma formiga escolher um caminho, aumenta com o número de formigas que passaram por aquele caminho anteriormente (BONABEAU *et al*, 1999, DORIGO *et al*, 1996, DORIGO e GAMBARDELLA, 1997, DORIGO *et al*, 1999, DORIGO & STÜTZLE, 2004, STUTZLE & DORIGO, 1999).

A idéia básica deste processo é ilustrado na Figura 9. Na ilustração (a) as formigas se movem em linha reta do ninho para a fonte de alimento e vice-versa. A ilustração (b) mostra o que acontece logo após um obstáculo ser colocado no caminho entre a fonte de alimento e o ninho. Para contornar o obstáculo, cada formiga, aleatoriamente, tem de escolher entre virar para a esquerda ou para a direita (com uma distribuição de probabilidade de 50%). Todas as formigas se movem aproximadamente com a mesma velocidade e depositam feromônio na trilha, aproximadamente à mesma taxa. Entretanto, as formigas que, probabilisticamente, escolherem virar para a esquerda (direção ninho→ comida) irão percorrer a trilha de feromônio mais rapidamente, enquanto que as formigas que optaram por contornar o obstáculo pela direita seguirão um caminho mais longo, demorando mais para contornar o obstáculo. O mesmo ocorre com as formigas que estiverem fazendo o caminho (comida→ ninho), ilustração (c). Como resultado, o feromônio é acumulado mais rapidamente no caminho mais curto ao redor do obstáculo. Visto que as formigas preferem seguir trilhas com grandes quantidades de feromônio, eventualmente todas as formigas convergirão para o caminho mais curto ao redor do obstáculo, como mostrado na ilustração (d). Este processo no qual uma formiga é influenciada por outra formiga a seguir determinado caminho em direção a uma fonte de alimento ou por uma trilha de feromônio é uma forma de estigmergia conhecida como recrutamento.

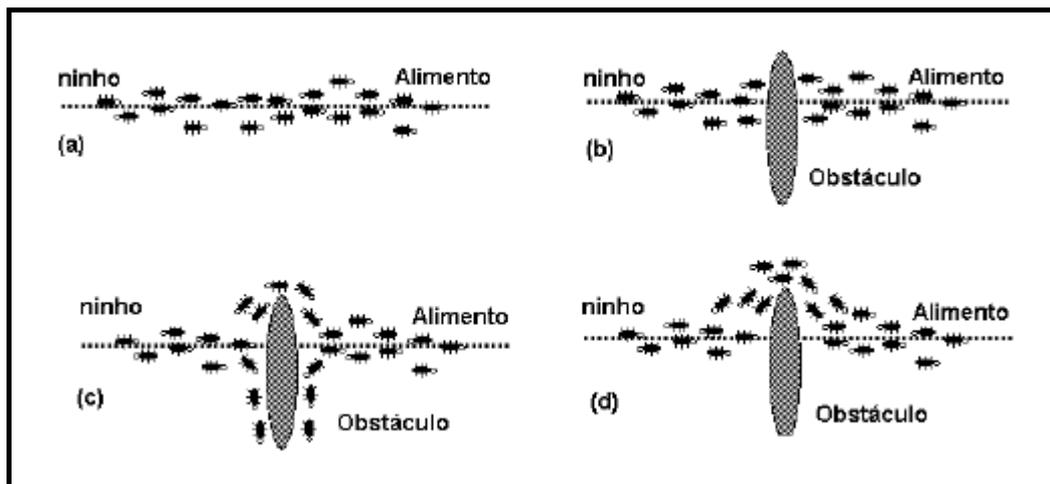


Figura 9 - Transposição de um Obstáculo (DORIGO, 2006)

4.4 - FORMIGAS ARTIFICIAIS

Um algoritmo de Colônias de Formigas (*Ant Colony Optimization* - ACO) é um sistema baseado em agentes que simula o comportamento natural das formigas na procura por alimentos desenvolvendo mecanismos de cooperação e aprendizado. A metodologia ACO foi proposta por DORIGO, *et al* (1996) como uma nova meta-heurística para resolver problemas de otimização combinatorial.

Existem diversos problemas de otimização combinatorial na área de Pesquisa Operacional que são da classe *NP*, isto é, não existem algoritmos limitados polinomialmente que resolvam estes problemas. Em situações práticas, estes problemas são muito difíceis de serem resolvidos por algoritmos determinísticos, pois seu espaço de busca é muito grande. Uma alternativa para a solução destes problemas é a utilização de algoritmos meta-heurísticos, os quais abandonam a certeza da busca em favor de heurísticas que aumentem a probabilidade de efetuar um movimento rápido em direção à meta final. Dentre estes métodos encontra-se o ACO.

A metodologia ACO tem sido aplicada a uma variedade de problemas de otimização combinatorial tais como: o problema do caixeiro viajante (DORIGO *et al*, 1997), associação quadrática (MANIEZZO & COLORNI, 1999), roteamento de veículos (BULLNHEIMER *et al* 1997), escalonamento *job shop* (COLORNI *et al.*, 1994), coloração de mapas (COSTA & HERTZ, 1997), dentre outras.

Além dos problemas de busca e otimização, recentemente temos visto na literatura o uso dos algoritmos PSO e ACO em *Data Mining*, com destaque para agrupamentos de dados: (AZZAG *et al*, 2003, FREITAS, 2001, HANDL, 2003, HANDL *et al*, 2003, HANDL & KNOWLES, 2004_a, 2004_b, HANDL *et al*, 2005,

HANDL & MEYER,2007, MONMARCHÉ, 1999, PARPINELLI *et al*, 2001, 2002, PULIDO & COELLO, 2004, VEENHUIS & KÖPPEN, 2004).

4.4.1 - Comparação entre as Formigas Artificiais e as Formigas Reais

Os algoritmos ACO originaram-se do estudo do comportamento da busca por alimento em formigas reais e fazem uso de (DORIGO *et al*, 1996, DORIGO *et al* 1999, DORIGO & STÜTZLE, 2004):

- uma colônia de indivíduos que cooperam entre si;
- uma trilha de feromônio artificial para comunicação local;
- uma sequência de movimentos locais para formar os caminhos; e
- uma política de decisão probabilística que faz uso somente da informação local.

Estes itens são detalhados a seguir (PARPINELLI, 2001):

- Cooperação entre os indivíduos da colônia. Como nas colônias reais, os algoritmos ACO são compostos por uma população, ou colônia, de entidades (formigas) concorrentes e assíncronas, que cooperam globalmente entre si para encontrar uma boa solução para o problema em questão. Mesmo que a complexidade do problema seja tal que prejudique a busca da formiga por soluções de alta qualidade, o resultado da cooperação entre os indivíduos da colônia como um todo normalmente produz soluções de alta qualidade. Como em uma colônia real, normalmente as formigas conseguem encontrar um caminho entre seu ninho e uma fonte de alimento. Esta cooperação entre os indivíduos da colônia surge por meio da informação que eles concorrentemente lêem/escrevem nos ‘estados do problema’ que visitam;
- Trilhas de feromônio. As formigas artificiais modificam alguns aspectos no ambiente em que ‘viverem’ assim como as formigas reais também o fazem. Enquanto as formigas reais depositam uma substância química, feromônio, as formigas artificiais modificam algumas informações numéricas armazenadas localmente no estado do problema que estão visitando. Por analogia, esta informação numérica é chamada de trilha de feromônio artificial, ou somente trilha de feromônio. Assim como as formigas reais, as formigas artificiais têm preferência probabilística por caminhos com maior quantidade de feromônio. Desta

forma, caminhos mais curtos tendem a ter altas taxas de crescimento em suas quantidades de feromônio. Em algoritmos ACO as trilhas de feromônio são os únicos canais de comunicação entre as formigas. Usualmente, um mecanismo de evaporação, similar à evaporação real do feromônio, modifica a informação da quantidade de feromônio o tempo todo. A evaporação do feromônio permite que a colônia de formigas vagorosamente esqueça seu passado, podendo direcionar sua busca para novas soluções sem a influência das decisões tomadas tempos atrás;

- Busca por caminhos curtos e uso de movimentos locais. Tanto as formigas reais quanto as formigas artificiais compartilham uma mesma tarefa: encontrar um caminho mais curto (de custo mínimo) que junte a origem (ninho) com um destino (fontes de comida). As formigas reais e artificiais não ‘pulam’ de um estado para outro, elas andam através de estados adjacentes no espaço de busca. A exata definição de estado e adjacência são específicas para cada problema ao qual se esteja aplicando o algoritmo;
- Regra de transição de estado probabilística. As formigas artificiais, assim como as reais, constroem soluções aplicando uma regra de transição probabilística para se mover através dos estados adjacentes. Como para as formigas reais, as formigas artificiais fazem uso somente da informação local, não podendo ‘olhar para frente’ para prever possíveis estados. Sendo assim, esta regra de transição é completamente local, no espaço e no tempo. A regra de transição é uma função composta pela informação previamente conhecida do problema, representada pelas especificações do mesmo (equivalente à estrutura do terreno para formigas reais), e pelas modificações locais no ambiente (trilhas de feromônio) provocadas pelas formigas no passado.

As formigas artificiais têm também algumas características que não são encontradas nas colônias de formigas reais (BONABEAU et al, 1999, DORIGO et al, 1996, DORIGO & STÜTZLE, 2004):

- Formigas artificiais vivem em um mundo discreto e seus movimentos consistem de transições de estados discretos para estados discretos;

- Formigas artificiais têm um estado interno. Este estado particular contém a memória das ações passadas de determinada formiga;
- Formigas artificiais depositam quantidades de feromônio em função da qualidade da solução encontrada. Somente algumas poucas espécies de formigas reais depositam quantidades de feromônio conforme a qualidade das fontes de alimentos encontradas; e
- O momento em que as formigas artificiais atualizam/depositam o feromônio nas trilhas depende do problema e frequentemente não reflete o comportamento das colônias reais. Por exemplo, em muitos casos as formigas artificiais atualizam as trilhas de feromônio somente depois de terem gerado uma solução.

Resumindo, a idéia básica é que, quando uma dada formiga tiver de escolher entre dois ou mais caminhos, o caminho que tiver sido mais frequentemente escolhido por outras formigas no passado terá uma maior probabilidade de ser escolhido por esta formiga para ser seguido. Sendo assim, trilhas com grandes quantidades de feromônio se tornam sinônimos de caminhos mais curtos (PARPINELLI, 2001).

4.4.2 - Desenvolvimento de um Algoritmo ACO

Em essência, um algoritmo ACO (*Ant Colony Optimization*) executa iterativamente um loop contendo dois procedimentos básicos, que são:

- Um procedimento especificando como as formigas constroem ou modificam as soluções do problema a ser resolvido; e
- Um procedimento para a atualização das trilhas de feromônio.

A construção ou modificação de uma solução é feita de maneira probabilística. A probabilidade de se adicionar um novo item à solução parcial corrente é dada por uma função heurística dependente do problema (η) e da quantidade de feromônio (τ) depositada pelas formigas no passado. A atualização das trilhas de feromônio é implementada como uma função que depende de uma taxa de evaporação do feromônio e da qualidade da solução produzida. Para implementar um ACO, primeiro deve-se definir (BONABEAU *et al*, 1999, DORIGO & STÜTZLE, 2004):

- Uma representação apropriada do problema, o qual deve permitir às formigas incrementalmente construir ou modificar soluções, por meio de uso de uma regra probabilística de transição baseada na quantidade de feromônio na trilha e em uma heurística local;

- Um método para forçar a construção de soluções válidas (por exemplo, soluções que são válidas em uma determinada situação do mundo real) correspondentes à definição do problema;
- Uma função heurística (η) que mede a qualidade dos itens que podem ser adicionados à solução parcial atual;
- Uma regra para atualização do feromônio, a qual especifica como modificar a quantidade de feromônio na trilha (τ); e
- Uma regra probabilística de transição baseada no valor da função heurística (η) e no conteúdo da trilha de feromônio (τ).

4.5 - ALGORITMO DE COLÔNIA DE FORMIGAS PARA AGRUPAMENTO DE DADOS

A clusterização inspirada em Colônia de Formigas foi proposta inicialmente, por DENEUBOURG *et al*, 1991.

A principal vantagem deste algoritmo é o fato de que não é necessária nenhuma informação inicial a respeito da massa de dados que iremos particionar, e os parâmetros necessários para sua execução são o número de “formigas“ que serão usadas e o número de ciclos que serão executados. Além das características descritas anteriormente, de diversos relatos na literatura do seu uso com sucesso para clusterização. : (AZZAG *et al*, 2003, FREITAS, 2001, HANDL, 2003, HANDL *et al*, 2003, HANDL & KNOWLES,2004_a, 2004_b, HANDL *et al*, 2005, HANDL & MEYER,2007, MONMARCHÉ, 1999, PULIDO & COELLO, 2004).

A implementação de um algoritmo ACO para Agrupamento de Dados, segue as seguintes etapas propostas por (MONMARCHÉ, 1999):

- (i) cada objeto é representado por um vetor de dimensão n , onde n representa seus atributos;
- (ii) os objetos são distribuídos aleatoriamente em um “tabuleiro”, dimensionado de acordo com o tamanho da nossa população;
- (iii) durante a execução do algoritmo, os objetos podem ser “empilhados” na mesma célula, constituindo “pilhas”; deste modo, as pilhas representam um *cluster*;
- (iv) a distância entre dois objetos é calculada como sendo a distância euclidiana entre dois vetores em R^n ;
- (v) o centro do *cluster* é determinado pelo centro de massa dos objetos que o compõem (centróide);

- (vi) a distância de dois *clusters* é dada pela distância dos seus centróides;
- (vii) um número pré-determinado de “formigas”, move-se no tabuleiro a cada interação, podendo tomar diferentes ações: largar ou pegar um objeto de acordo com seu estado:
 - se ela não carrega nenhum objeto ela pode: pegar um objeto na célula vizinha ou pegar o objeto com maior dissimilaridade (objeto que apresenta a maior distância ao centróide da pilha) da “pilha” vizinha;
 - se ela carrega um objeto ela pode: soltar o objeto em uma célula vizinha vazia, soltar o objeto em uma célula ocupada por um objeto e formar uma “pilha” ou soltar o objeto em uma célula ocupada por uma pilha, onde a colocação do objeto não interfira no deslocamento do centróide.
- (viii) O procedimento vii é repetido até atingirmos um número de interações pré-estabelecidas.

4.6 - NÚCLEO DO ALGORITMO

O algoritmo de Colônia de Formigas para Agrupamento de Dados, foi dividido da seguinte maneira conforme a propostas de (MONMARCHÉ, 1999):

Início do Algoritmo:

1. Distribua aleatoriamente as formigas (*ant*) no tabuleiro,
2. Para cada ant_i faça
 - a. Mova ant_i ,
 - b. Se ant_i não carrega nenhum objeto, Então olhe as oito células em volta e veja a possibilidade de carregar um objeto. (veja algoritmo “for picking up”),
 - c. Senão (ant_i está carregando um objeto O) olhe as oito células em volta e a possibilidade de soltar O . (veja algorithm for dropping),
3. Até atingir o critério de parada.

Explore as células c ao redor de ant_i de forma aleatória:

8	1	5
3	ant_i	2
6	4	7

Algoritmo for Picking up:

1. Classifique as oito células ao redor de ant_i como inexploradas,
2. Repita
 - a. Se c está ocupada então realize uma das seguintes ações:
 - i. Caso c contenha um objeto O , Então carregue O com a probabilidade, P_{load}
 - ii. Caso c contenha uma pilha de dois objetos, então remova um dos dois com a probabilidade $P_{destroy}$, senão.
 - iii. Caso c contenha uma pilha H com mais de dois objetos, então remova o objeto com maior dissimilaridade $O_{dissim}(H)$
 - b. Classifique c como explorada,
3. Até todas as oito células terem sido exploradas ou um objeto tenha sido carregado.

Algoritmo for Dropping

1. Classifique as oito células ao redor de ant_i como inexploradas,
2. Repita
 - i. Caso c esteja vazia, então solte O com a probabilidade, P_{drop}
 - ii. Caso c contém um objeto O' então solte O para criar uma pilha H , mas verifique a máxima dissimilaridade permitida para criar uma pilha de dois objetos (T_{create})
 - iii. Caso c contenha uma pilha H então solte O em H se e somente se não ocorra alteração do centróide de H
 - iv. Senão Classifique c como explorada
3. Até todas as oito células terem sido exploradas ou o objeto carregado tenha sido solto.

A parada do algoritmo ocorrerá após n ciclos definidos pelo usuário.

A Tabela 4 apresenta os parâmetros do algoritmo, sugeridos e testados em MONMARCHÉ, 1999.

Tabela 4 - Parâmetros do algoritmo

Parâmetros		Valores
P_{load}	Prob . de <i>pick up</i> um objeto	[0.4, 0.8]
$P_{destroy}$	Prob de destruir uma pilha de 2 objetos	[0, 0.6]
T_{create}	Max. Dissimilaridade permitida para criar uma pilha de dois objetos	[0.05, 0.2]

4.7 - IMPLEMENTAÇÃO DO ALGORITMO DE COLÔNIA DE FORMIGAS PARA AGRUPAMENTO DE DADOS

A implementação do Algoritmo de Colônia de Formigas para Agrupamento de Dados denominado de Formigas, seguindo os preceitos estabelecidos em 4.5 e 4.6, foi feita no software WEKA.

A implementação foi feita utilizando a linguagem JAVA, aproveitando as classes e os objetos do WEKA e usando o software ECLIPSE, versão 3.1 e posteriormente a versão EUROPA, como plataforma de desenvolvimento.

A Figura10 mostra o diagrama de classe dos três algoritmos implementados: Formigas, Formiga e Tabuleiro, que formam o Algoritmo de Colônia de Formigas para Agrupamento de Dados. Eles foram implementados na classe *clusterers* do WEKA.

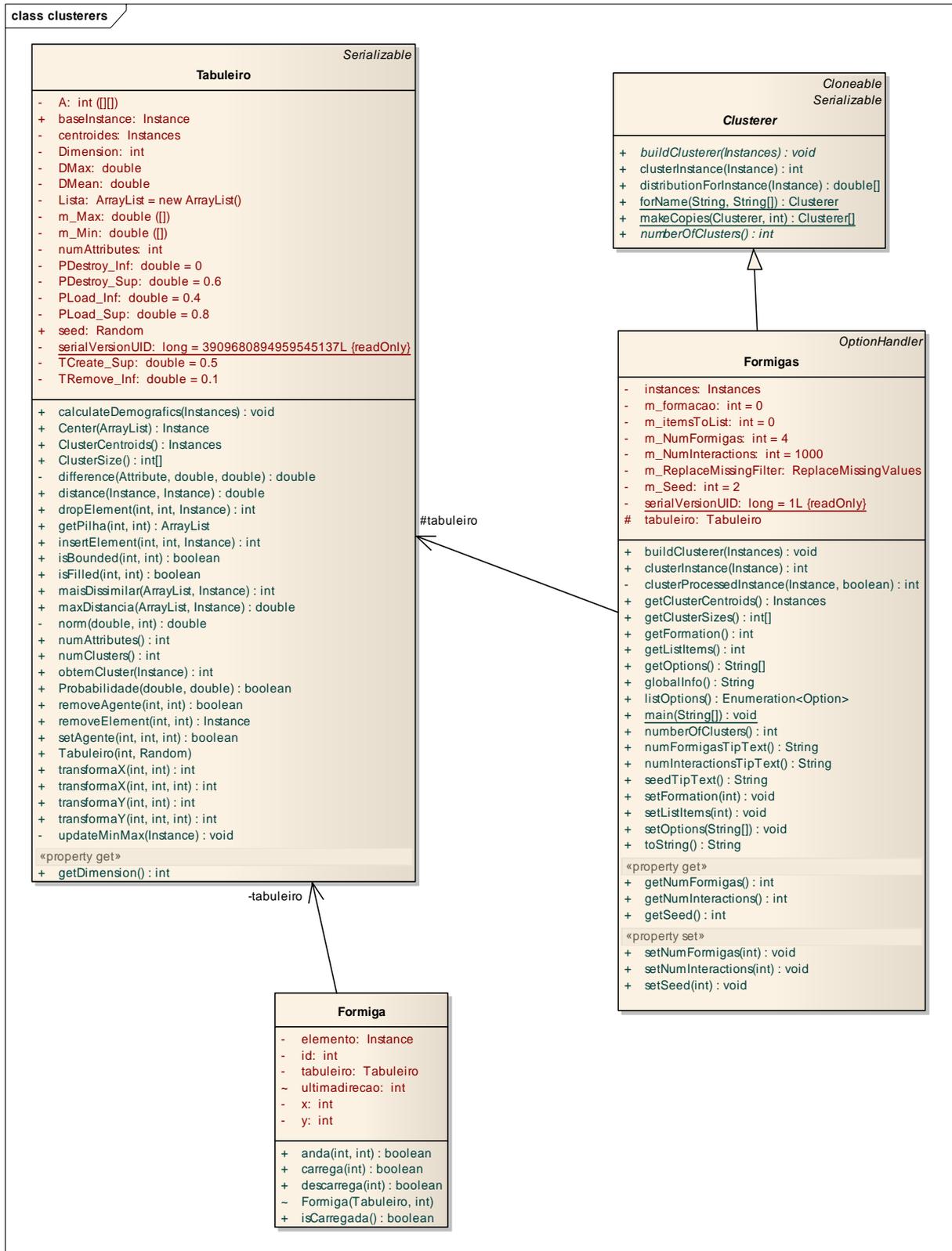


Figura 10 - Diagrama de Classe

4.8 - O SOFTWARE WEKA

O pacote Weka (Waikato Environment for Knowledge Analysis) é formado por um conjunto de implementações de algoritmos de diversas técnicas de Mineração de Dados.

O WEKA está implementado na linguagem Java, que tem como principal característica ser portátil. Desta forma, pode rodar nas mais variadas plataformas, aproveitando os benefícios de uma linguagem orientada a objetos como modularidade, polimorfismo, encapsulamento, reutilização de código dentre outros. Além disso é um software de domínio público estando disponível em (Weka Software, 2001).

O WEKA possui um formato próprio de arquivo de dados, o ARFF, o qual descreve o domínio do atributo, pois o mesmo não pode ser obtido automaticamente pelo seu valor.

Antes de aplicar os dados a qualquer algoritmo do pacote WEKA, estes devem ser convertidos para o formato ARFF que consiste basicamente de duas partes. A primeira contém uma lista de todos os atributos, onde se define o tipo do atributo ou os valores que ele pode representar, quando se utiliza valores estes devem estar entre “{ }” separados por vírgulas. A segunda parte consiste das instâncias, ou seja, os registros a serem minerados com o valor dos atributos para cada instância separado por vírgula, a ausência de um item em um registro deve ser atribuída pelo símbolo “?”.(WITTEN &FRANK, 2005)

Podem-se usar programas de planilhas eletrônicas e banco de dados os quais permitem exportar os dados em um arquivo onde as vírgulas são os separadores.

Uma vez feito isso, é necessário apenas carregar o arquivo em um editor de texto e adicionar o nome do conjunto de dados usando @relation nome_do_conjunto_de_dados, para cada atributo usa @attribute o nome do atributo e o tipo do atributo (real, nominal, categórico, binário etc) e após colocar uma linha com @data e logo em seguida os dados em si, salvando o arquivo como texto puro com extensão ARFF.

A Figura 11 apresenta o exemplo de um arquivo no formato ARFF reconhecido pelo WEKA.

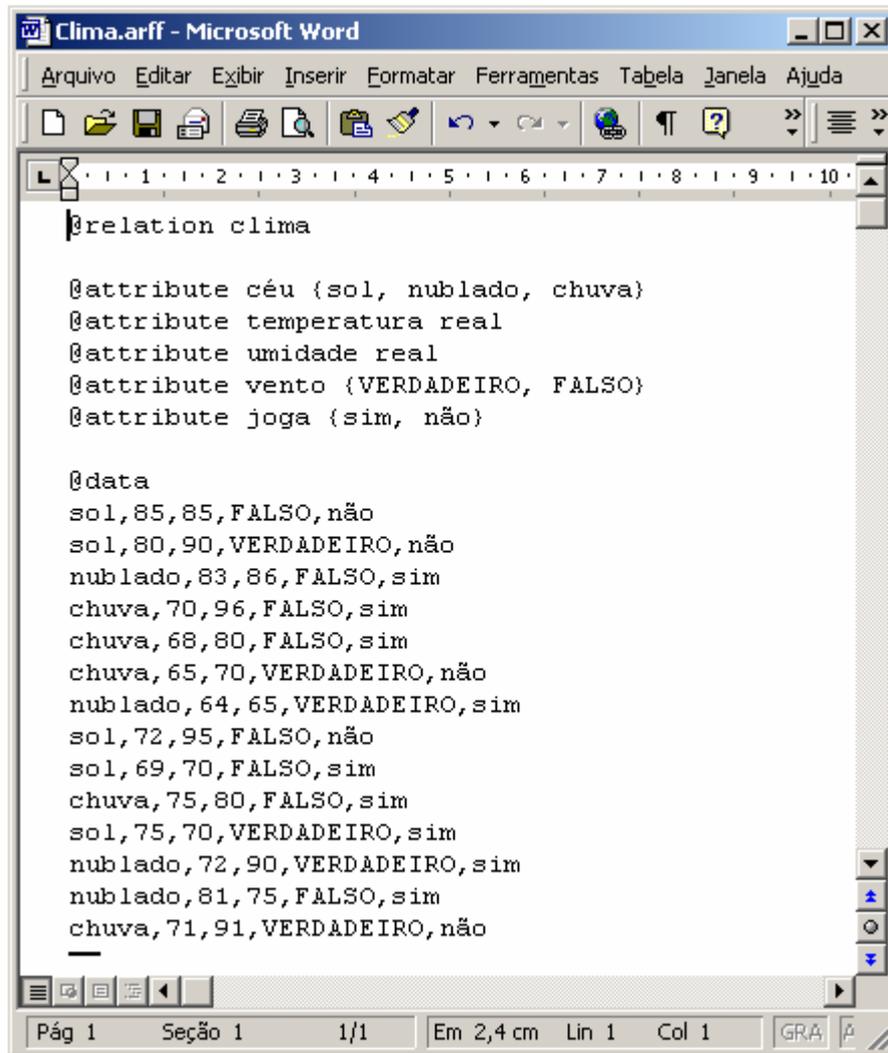


Figura 11 - Arquivo no Formato ARFF

5 - METODOLOGIA PARA SE DETERMINAR OS PARÂMETROS INTERNOS DO ALGORITMO DE COLÔNIA DE FORMIGAS PARA AGRUPAMENTO DE DADOS

A fim de validar os parâmetros internos contidos na Tabela 4, foram usadas bases contendo apenas duas variáveis, construídas para realizar testes com o algoritmo: a Clusterteste e a Clusterideal. As plotagens das bases são mostradas na Figuras 12 e Figura 13.

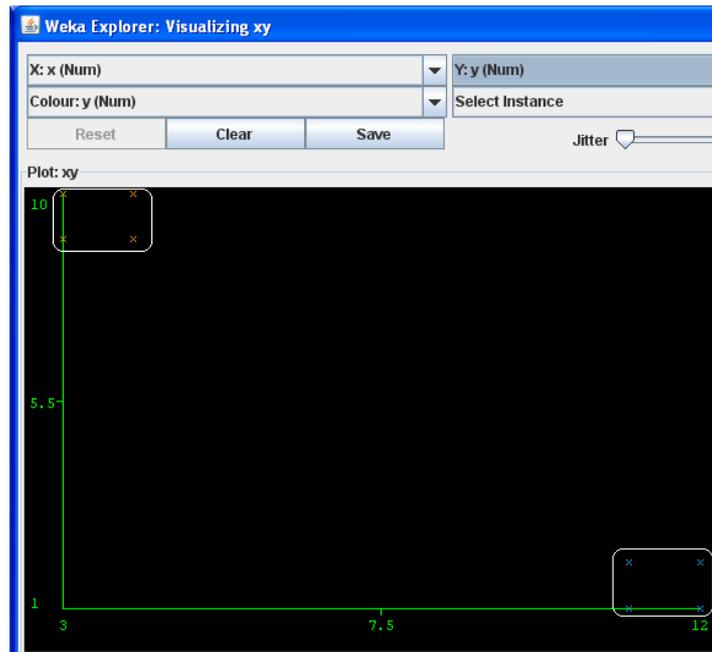


Figura 12 - ClusterTeste

A base Clusterteste é formada por dois grupos, bem separados, com quatro elementos cada.

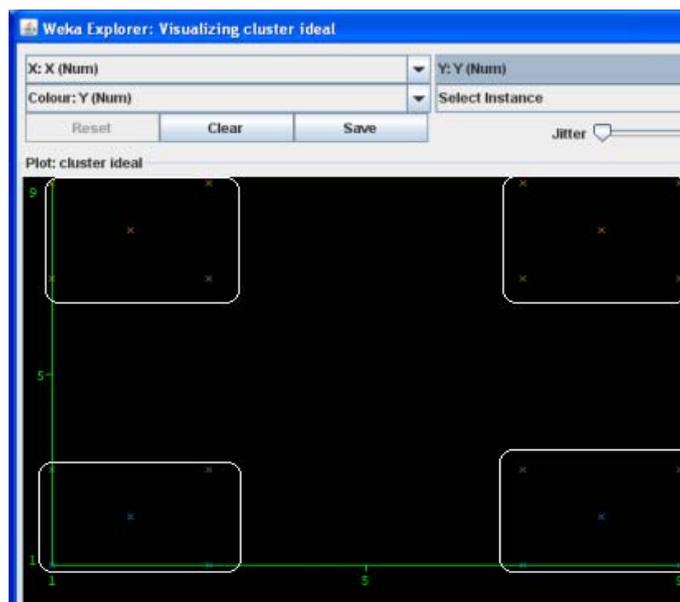


Figura 13 - ClusterIdeal

A base ClusterIdeal é formada por quatro grupos com cinco elementos cada.

Para o processo de validação foram estabelecidas duas métricas: a primeira métrica observada foi o acerto do número de *clusters*; em seguida mediu-se a acurácia, ou seja, a capacidade do algoritmo em acertar os elementos dos *clusters*.

Durante o processo de validação os parâmetros da Tabela 4 foram variados um a um e observadas as mudanças nas respostas do algoritmo.

Uma vez cumprida esta etapa, utilizando as bases construídas para este fim, com o algoritmo chegando a 100% de acerto nas duas métricas, observamos que os melhores parâmetros eram aqueles propostos inicialmente por MONMARCHÉ, 1999, a saber:

Tabela 5 - Parâmetros Validados

Parâmetros		Valores
P_{load}	Prob. de <i>pick up</i> um objeto	[0.4, 0.8]
P_{drop}	Prob de <i>drop up</i> um objeto	[0, 0.6]
T_{create}	Max. Dissimilaridade permitida para criar uma pilha de dois objetos	[0.05, 0.2]

Dando continuidade à validação dos parâmetros internos, novos testes foram realizados com os parâmetros da Tabela 5, com bases de dados mais complexas disponíveis em SANTOS, 2007, entretanto ainda bidimensionais.

As bases foram plotadas no *software* WEKA e são mostradas abaixo:

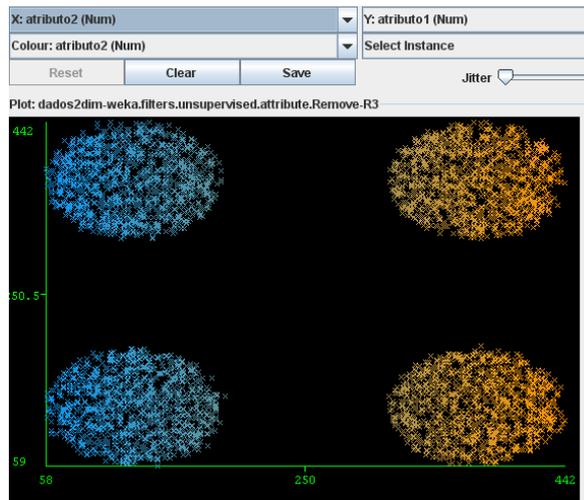


Figura 14 - p02.5_4 Elipses

A base plotada na Figura 14 é formada por quatro figuras em forma de elipse, bem separadas no espaço, apresentando um pequeno grau de dificuldade para o agrupamento.

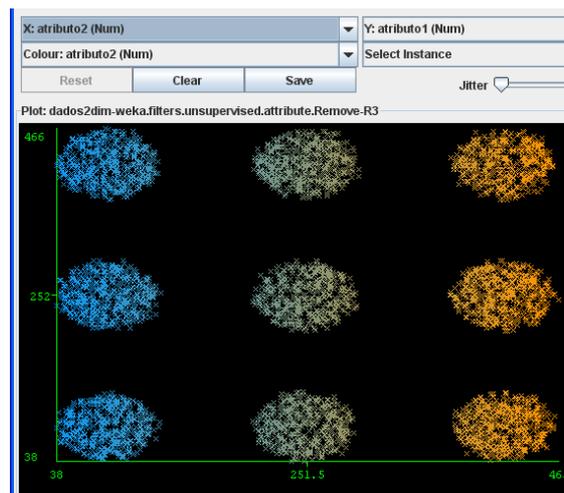


Figura 15 - p04.5_9Elipses

A base plotada na Figura 15 é formada por nove figuras em forma de elipse, bem separadas no espaço, apresentando um grau de dificuldade maior que o agrupamento anterior.

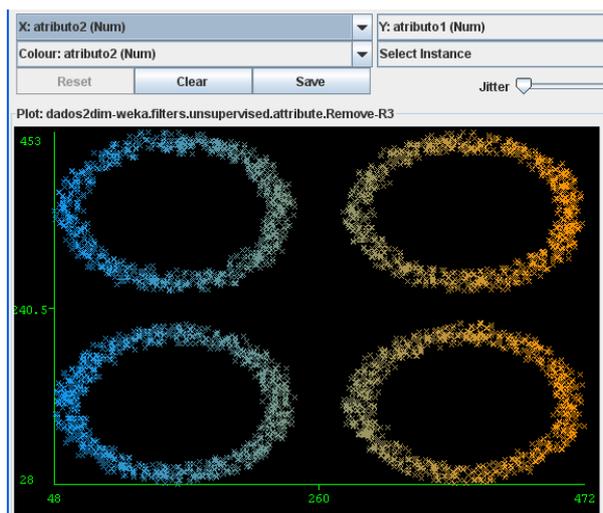


Figura 16 - p37.5_4Arcos

A base plotada na Figura 16 é formada por quatro arcos que não estão bem separados no espaço, sendo esta base mais complexa que as anteriores.

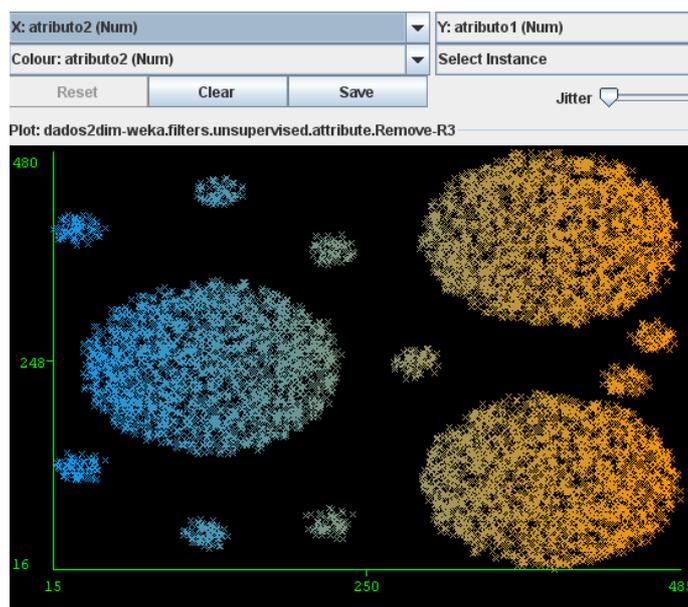


Figura 17 - p15.5_3 Elipses

A base plotada na Figura 17 é formada por 3 figuras em forma de elipse e por outras nove figuras bem menores que também de aproximam de uma elipse. Em alguns pontos estas figuras estão bem próximas, dificultando a tarefa de agrupamento.

Após efetuar o agrupamento das quatro bases anteriores foi observado que os resultados apresentados foram os mesmos observados anteriormente com as bases bi-dimensionais, com 100% de acertos no número de clusters e em acurácia. Desta forma podem-se considerar válidos os parâmetros internos do algoritmo contidos na Tabela 5 para bases bi dimensionais.

Antes de dar prosseguimento ao estudo com bases multidimensionais, será mostrada a maneira de se configurar alguns parâmetros do algoritmo. Cabe ressaltar que dos parâmetros configuráveis somente dois são necessários o usuário alterar: o número de formigas e o número de ciclos a serem executados. Os outros parâmetros são *default*, sendo opcional a sua configuração.

A seguir serão mostrados, na interface gráfica do WEKA, os parâmetros que são configuráveis pelo usuário, após a abertura da tela de inicialização do WEKA e da escolha da função a ser utilizada Figura 18, faz-se a escolha do algoritmo de clusterização, Figura19.

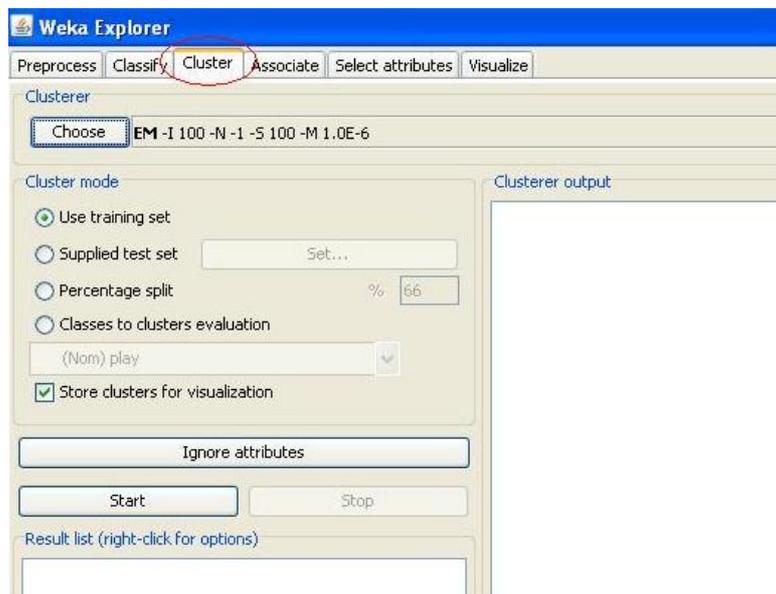


Figura 18 - Tela de Inicialização do WEKA



Figura 19 - Seleção do Algoritmo de Clusterização

Os algoritmos de clusterização que estão disponibilizados no WEKA estão mostrados na Figura 19, sendo que o algoritmo Formigas é o Algoritmo de Colônia de Formigas para Agrupamento de Dados que foi implementado no WEKA.

A Figura 20 apresenta os parâmetros configuráveis pelo usuário do Algoritmo de Colônia de Formigas para Agrupamento de Dados.

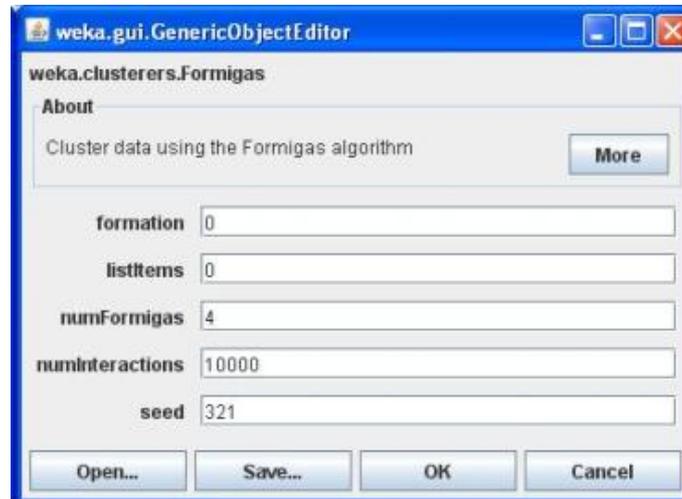


Figura 20 - Parâmetros Configuráveis

5.1 - DETERMINAÇÃO DO NÚMERO DE FORMIGAS

A determinação do número de formigas F (numFormigas) foi feita por meio dos experimentos realizados em 5.1, e baseado nos melhores resultados, chegou-se à conclusão que a quantidade de formigas a ser utilizada está em torno de $1/3$ do número de dados de que dispomos.

5.2 - NÚMERO DE INTERAÇÕES

O número de interações N (numInteractions) é selecionada no WEKA, conforme Figura 17 e, por *default*, está configurada para 10000 ciclos, sendo este considerado um bom número tendo em vista os resultados apresentados em 5.1, podendo ser alterado .

5.3 - FORMAÇÃO DO *CLUSTER*

Na descrição do algoritmo feita nas seções 4.5 e 4.6 de acordo com o estabelecido por MONMARCHÉ, 1999, o término do algoritmo ocorre após a execução dos n ciclos; entretanto, após a parada do algoritmo teremos pilhas formadas e elementos “soltos” que não formaram pilhas.

Para contornar o problema descrito acima foi criada uma maneira para se formar os *clusters* após a parada do algoritmo. A formação dos *clusters* depende do valor do parâmetro x “formation”.

Este parâmetro é informado na janela da Figura 17, podendo assumir os valores de 0, 1 ou 2.

- Para $x = 0$, o *cluster* é montado pela distância entre os centróides e os elementos, independentemente da posição dos elementos nas pilhas ou se o elemento faz parte ou não de alguma pilha.
- Para $x = 1$, o *cluster* só é formado com os objetos que compõem cada pilha. Desta maneira teremos vários objetos que não irão fazer parte de nenhum *cluster*; e
- Para $x = 2$, o *cluster* é formado com os objetos que compõem cada pilha, e os elementos que não fazem de nenhuma pilha se juntam à pilha cujo centróide esteja mais próximo.

O valor de $x = 2$ foi o que apresentou melhor resultado nos experimentos, e isto já era esperado uma vez que para $x = 0$ existe muita liberdade na formação dos *clusters* e $x = 1$ o critério de formação era muito rígido.

5.4 - OUTROS PARÂMETROS

O parâmetro L “listeItens”, apresenta por extenso a listagem dos itens nos *clusters* em que foram alocados. Quando igual a n (número de elementos da massa de dados) mostra, no console (na tela *cluster output*), em que cluster ficou cada objeto. Quando no console algum objeto aparece com valor -1, isto indica que aquele objeto não foi incluído em nenhum *cluster*. Por *default* o valor de L é 0, não apresentando nenhuma relação de objeto e *cluster*.

A configuração deste parâmetro é importante quando se desejar verificar em que *cluster* foi alocado cada elemento.

O último parâmetro S é a semente do gerador de números pseudo-aleatórios, “seed”. Apesar de existir uma rotina que determine esta semente, também existe a possibilidade dela ser escolhida ou replicá-la, caso necessite repetir algum experimento alterando alguns dos parâmetros acima, sem que se tenha a influência do gerador de números pseudo-aleatórios.

Como referência para estudos futuros, a realização dos experimentos e dos estudos de caso foi utilizado um PC com processador Intel Pentium D 2,8 Ghz com 1,5 GB de RAM e sistema operacional Microsoft Windows XP Home Edition.

5.5 - BASES MULTIDIMENSIONAIS

Para a experimentação com bases de mais de duas dimensões iniciou-se com a utilização do algoritmo dentro da melhor configuração estabelecida para as bases de 2 dimensões :

- $x = 2$;
- $F = 1/3$ do número de dados;
- $N = 10000$ ciclos; e
- S dado pelo sistema.

Ao fim dos ciclos, teremos uma quantidade de *clusters* m formada.

Repetimos a fase inicial com os *clusters* (m) encontrados, sendo neste caso, seus atributos, as coordenadas dos centróides dos m *clusters*.

Faz-se a conversão das coordenadas dos centróides dos m clusters para o formato de arquivo ARFF, conforme descrito na seção 4.8.

Esta fase é repetida até que z clusters seja o resultado em todas as execuções do algoritmo. Isto indica que o algoritmo estabilizou e o número de clusters procurados é o z .

Para a validação do método acima descrito usamos a base Fisher's Iris (Fisher,1936), sendo que esta é considerada clássica na literatura para *benchmark* de trabalhos correlatos. Ela é composta de 150 amostras, 4 atributos e dividida em 3 grupos.

A tabela abaixo resume os resultados obtidos em 3 dos experimentos realizados:

Tabela 6 - Resultados dos Experimentos

Execução do Algoritmo	1ºExperimento	2ºExperimento	3ºExperimento
1ª execução	48 <i>clusters</i>	51 <i>clusters</i>	37 <i>clusters</i>
2ª execução	13 <i>clusters</i>	27 <i>clusters</i>	18 <i>clusters</i>
3ª execução	3 <i>clusters</i>	9 <i>clusters</i>	7 <i>clusters</i>
4ª execução	3 <i>clusters</i>	3 <i>clusters</i>	3 <i>clusters</i>
5ª execução	3 <i>clusters</i>	3 <i>clusters</i>	3 <i>clusters</i>
6ª execução	---	3 <i>clusters</i>	3 <i>clusters</i>

A diferença entre os experimentos é a semente S do gerador de números pseudo-aleatórios, que para cada experimento o algoritmo atribuiu uma semente distinta.

No 1º experimento a partir da 3ª execução pode-se observar que o número de clusters se estabiliza em 3 e nos demais experimentos esta estabilidade começa a partir da quarta execução.

Com os resultados apresentados acima podemos dizer que o método de determinação do número de *clusters* apresentado é viável de ser utilizado.

6 - ESTUDO DE CASO

Com o intuito de, mais uma vez, validar e comprovar a viabilidade da metodologia desenvolvida para se determinar o número de *clusters* em bases multidimensionais, descrita no Capítulo 5, foram realizados oito estudos de casos empregando-se o algoritmo de Colônia de Formigas para Agrupamento de Dados implementado no *software* WEKA e feita a comparação com o algoritmo *Expectation Maximization* (EM), cujas características básicas foram abordadas no Capítulo 3, além de ser um algoritmo residente do WEKA e fornecer o número final de *clusters* sem a necessidade de se precisar fornecer o número inicial.

Foram avaliados dois pontos: o acerto do número de *clusters* e a acurácia .

As bases de teste usadas foram: íris, lentes de contato, soybean, bank, wine, mushroom, pima diabetes, breast câncer wisconsin obtidas no *software* WEKA e no repositório de dados da UCI (Universidade de Califórnia Irvine), disponível em UCI, 2007.

A Tabela 7 mostra as características das bases : números de atributos e número de exemplos das bases de dados que foram usadas.

Tabela 7 - Características das bases de dados

Base	Atributos	Exemplos
Iris	4	150
Lentes de Contato	5	24
Soybean	34	683
Bank	16	600
Wine	13	178
Mushroom	22	8124
Pima Diabetes	8	768
Breast Cancer Wisconsin	9	699

6.1 - RESULTADOS APRESENTADOS PELO ALGORITMO COLÔNIA DE FORMIGA PARA AGRUPAMENTO DE DADOS

As oito bases de dados descritas na Tabela 7 foram “rodadas” no Algoritmo de Colônia de Formigas para Agrupamento de Dados.

Para o estudo de caso a configuração do algoritmo foi a mesma utilizada na seção 5.6:

- $x = 2$;
- $F = 1/3$ do número de dados;
- $N = 10000$ ciclos; e
- S , dado pelo sistema.

Como na seção 5.6, esta configuração mostrou-se satisfatória, com o algoritmo atingindo o resultado esperado em todos os experimentos.

A maior dificuldade observada durante os experimentos foi que o método para se determinar o número de *clusters* necessita que, para cada nova interação se faça a conversão das coordenadas dos centróides dos *clusters* encontrados para o formato de arquivo ARFF, conforme descrito na seção 4.8.

A Tabela 8 apresenta os resultados obtidos com o Algoritmo de Colônia de Formigas para Agrupamento de Dados.

Tabela 8 - Resultados do Algoritmo de Colônia de Formigas para Agrupamento de Dados

Base	NºCluster	NºCluster Alg. Implementado	AcuráciaAlg. Implementado	Iterações Alg. Implementado
Íris	3	3	90%	5
Lentes de Contato	3	3	100%	3
Soybean	19	19	90%	6
Bank	6	6	95%	5
Wine	3	3	90%	3
Mushroom	2	2	90%	4
Pima Diabetes	2	2	90%	5
Breast Cancer Wisconsin	2	2	95%	4

6.2 - RESULTADOS APRESENTADOS PELO ALGORITMO *EXPECTATION MAXIMIZATION*

As oito bases de dados descritas na Tabela 7 foram “rodadas” no Algoritmo *Expectation Maximization* nativo do software WEKA. A Figura 21 apresenta os parâmetros do algoritmo EM.

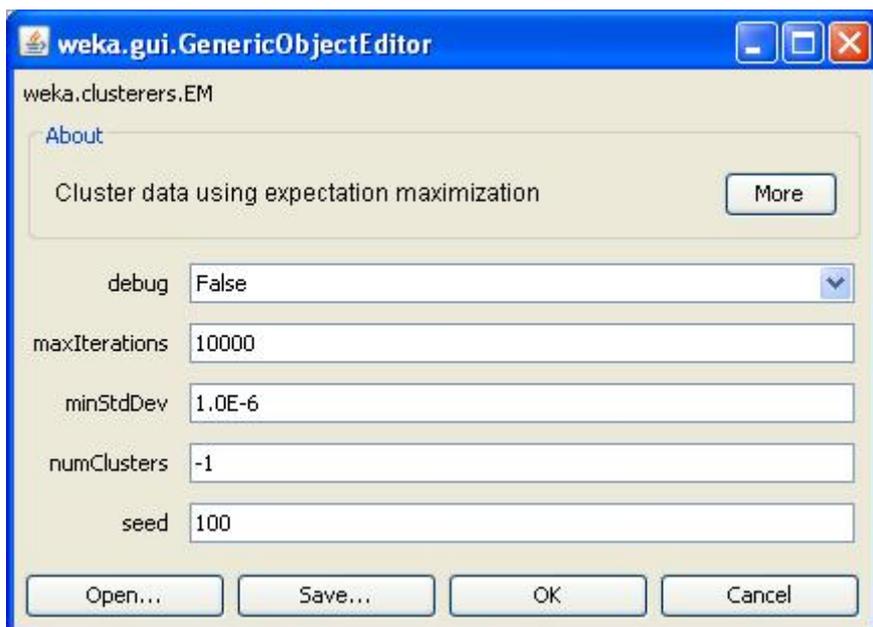


Figura 21 - Parâmetros do Algoritmo EM

Os parâmetros do algoritmo não foram alterados, seguindo a configuração *default* do WEKA. Com todas as bases foram feitos vários experimentos, sendo que o único parâmetro variado era a semente do gerador de números pseudo-aleatórios. Os resultados computados foram aqueles que mais se aproximaram do número real de *clusters*.

A Tabela 9 sintetiza os resultados encontrados com o algoritmo EM. Neste algoritmo não foi avaliada a acurácia, pois sua taxa de acerto do número de clusters foi nula; desta forma não faz sentido computar a acurácia.

Tabela 9 - Resultados do Algoritmo *Expectation Maximization*

Base	NºCluster	NºCluster EM
Íris	3	5
Lentes de Contato	3	2
Soybean	19	4
Bank	6	8
Wine	3	2
Mushroom	2	13
Pima Diabetes	2	6
Breast Cancer Wisconsin	2	4

6.3 - RESULTADOS FINAIS

Nos casos analisados, o Algoritmo de Colônia de Formigas para Agrupamento de Dados implementado apresentou a melhor taxa de acerto que o algoritmo EM.

Quanto à acurácia, o algoritmo EM não foi avaliado, pois a taxa de acerto do número de clusters foi nula, não fazendo sentido a comparação com o outro algoritmo que sempre apresentou um resultado muito bom neste quesito.

A Tabela 10 apresenta o quadro resumo dos resultados obtidos.

Tabela 10 - Quadro resumo dos resultados

Base	NºCluster	NºCluster EM	NºCluster Alg. Implementado	AcuráciaAlg. Implementado	Iterações Alg. Implementado
Íris	3	5	3	90%	5
Lentes de Contato	3	2	3	100%	3
Soybean	19	4	19	90%	6
Bank	6	8	6	95%	5
Wine	3	2	3	90%	3
Mushroom	2	13	2	90%	4
Pima Diabetes	2	6	2	90%	5
Breast Cancer Wisconsin	2	4	2	95%	4

7 - CONCLUSÃO E SUGESTÕES DE TRABALHOS FUTUROS

Este trabalho apresentou como objetivo uma proposta de ferramenta que seja de fácil utilização para um usuário não especialista segmentar uma massa de dados em um número desconhecido de grupos.

A maioria das tarefas de Mineração de Dados sofre fortes restrições para serem realizadas por um usuário comum, ou seja, aquele que é especialista na base de dados que irá ser analisada, mas não tem domínio das ferramentas existentes, pois as mesmas, muitas vezes, requerem um nível de conhecimento técnico em diversas áreas.

Quando trabalhamos com segmentação de dados, a tarefa se torna um pouco mais árdua, pois além dos diversos parâmetros que temos que ajustar, o que requer conhecimento específico do algoritmo que está sendo utilizado, alguns algoritmos necessitam do número (n) de conjuntos (*clusters*) nos quais queremos particionar a nossa massa de dados, além de serem sensíveis à partição que é feita inicialmente.

Para tornar a tarefa de agrupamento de dados o menos árdua possível buscou-se um algoritmo que realizasse a tarefa de segmentar os dados sem que fosse necessário selecionar a priori um número inicial de conjuntos e que não necessitasse de parâmetros complexos para se ajustar.

Dentre muitos algoritmos sugeridos na literatura, o paradigma escolhido foi o inspirado na teoria de Colônia de Formigas pois, além de diversos relatos na literatura do seu uso para clusterização com sucesso, os únicos parâmetros a serem selecionados são os números de formigas que serão usadas e a quantidade de ciclos a serem executados, não necessitando de nenhum conhecimento prévio da massa de dados.

Este algoritmo foi implementado no software WEKA, desenvolvido por uma equipe de pesquisadores da Universidade de Waikato (Nova Zelândia); é feito em Java, com o código fonte disponível o que, teoricamente, permite que o software seja utilizado em qualquer plataforma computacional.

A escolha se deu por ser um software livre, pela simplicidade de uso, para aproveitar as facilidades de entrada e saída de dados e interfaces gráficas residentes além de ter incluído filtros para a fase de pré-processamento e uma grande quantidade de técnicas e algoritmos usados em Mineração de dados.

Foram feitos experimentos com bases de dados bidimensionais criadas para ajustar os parâmetros internos da ferramenta proposta.

Variando os parâmetros internos do algoritmo, foram feitas medidas de acerto do número de *clusters* e acurácia, com o índice de acerto chegando a 100% nas duas medidas.

Foram feitos novos testes, ainda com bases de duas variáveis, entretanto mais complexas, e os índices de acerto foram de 100% nas duas medidas.

Além dos parâmetros internos do algoritmo, existe a necessidade de se fazer a seleção de alguns outros parâmetros. Entretanto, para o usuário comum, basta selecionar o número de formigas, que é em torno de 1/3 do número de amostras, e o número de ciclos que por *default* é 10000. Assim, atendeu-se ao objetivo que era proporcionar uma ferramenta de fácil utilização.

Na descrição do algoritmo feita nas seções 4.5 e 4.6, de acordo com o estabelecido por MONMARCHÉ, 1999, o término do algoritmo ocorre após a execução dos n ciclos; entretanto, após a parada do algoritmo teremos pilhas formadas e elementos “soltos” que não formaram pilhas.

Para contornar o problema descrito acima foi desenvolvido um método que consiste inicialmente em executar o algoritmo 1 vez, e verificar o número de clusters encontrados. Executar uma segunda vez, sendo os atributos dos dados de entrada os centróides dos *clusters* formados anteriormente. O algoritmo irá sendo executado até que o número de *clusters* comece a se repetir, indicando que o mesmo se estabilizou.

O problema observado neste método é que, para cada nova interação existe a necessidade de se fazer a conversão das coordenadas dos centróides dos m clusters para o formato de arquivo ARFF, conforme descrito na seção 4.8.

Este método foi empregado para o agrupamento de bases multidimensionais. Para a validação deste método, foi usada a base de dados flores íris em três experimentos, e observamos que a partir da segunda e terceira execução o algoritmo se estabilizava. Desta maneira, com os resultados apresentados, podemos dizer que o método apresentado é viável de ser utilizado.

Com o intuito de, mais uma vez, validar e comprovar a viabilidade do método desenvolvido para a determinação do número de *clusters*, do Algoritmo de Colônia de Formigas para Agrupamento de Dados implementado no WEKA, e do método desenvolvido para a formação dos *clusters*, foram realizados oito estudos de caso empregando-se o Algoritmo de Colônia de Formigas para Agrupamento de Dados, implementado no *software* WEKA e feita a comparação com o algoritmo *Expectation*

Maximization (EM). Além de fornecer o número final de clusters sem a necessidade de se precisar fornecer o número inicial de *cluster* ele é um algoritmo residente do WEKA.

As bases de teste usadas foram: íris, lentes de contato, soybean, bank, wine, mushroom, pima diabetes, breast cancer wisconsin obtidas no software WEKA e no repositório de dados da UCI (Universidade de Califórnia Irvine), disponível em UCI, 2007.

Nos casos analisados, o Algoritmo de Colônia de Formigas para Agrupamento de Dados apresentou melhor taxa de acerto que o algoritmo EM.

Quanto à acurácia, o algoritmo EM não foi avaliado, pois a taxa de acerto do número de clusters foi nula, não fazendo sentido a comparação com o Algoritmo de Colônia de Formigas para Agrupamento de Dados que sempre apresentou um resultado muito bom neste quesito.

Desta forma, pode-se concluir que para o especialista na base de dados que será analisada, o Algoritmo de Colônia de Formigas para Agrupamento de Dados implementado no software WEKA pode atender ao objetivo de se determinar o número de *clusters* de uma massa de dados em um número desconhecido de grupos, somente inseridos dois parâmetros: o número de formigas e o número de ciclos a serem executados.

Deve-se ressaltar ainda que, o Algoritmo de Colônia de Formigas para Agrupamento de Dados proposto, não é um algoritmo híbrido. Na literatura encontra-se propostas como o algoritmo inspirado no reconhecimento químico das colônias de formigas (LABROCHE, 2002), que aborda um novo conceito, onde haveria a comunicação entre as formigas, não apenas a formação de pilhas .

Como trabalho futuro, além de processar exaustivamente bases de dados para avaliar o desempenho do algoritmo implementado, deve-se examinar comparativamente os novos algoritmos propostos com novos paradigmas.

Para tornar mais rápida a execução do trabalho de agrupamento fica a sugestão de se elaborar uma rotina para se fazer a conversão automática para arquivo tipo ARFF das coordenadas dos centróides dos *m clusters* no método da determinação dos *clusters* descrito na seção 5. 6.

REFERÊNCIAS BIBLIOGRÁFICAS

- ANDRADE, L. P., **Procedimento Interativo de Agrupamento de Dados**. Tese M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2004.
- AZZAG, H., MONMARCHÉ, N., SLIMANE, M., GUINOT, C., VENTURINI, G.. **A clustering algorithm based on the ants self-assembly behaviour**. **Advances in Artificial Life** - Proceedings of the 7th European Conference on Artificial Life (ECAL), vol. 2801, p. 564-571, Dortmund, Germany, 2003.
- AZZAG, H., GUINOT, C., VENTURINI, G., **How to use ants for hierarchical clustering**. Fourth international workshop on Ant Colony Optimization and Swarm Intelligence, p.350-357, LNCS 3172, Brussels, Belgium, 2004.
- AZZAG H., VENTURIN G. I, OLIVER A., GUINOT C., **A hierarchical ant based clustering algorithm and its use in three real-world applications**. Special Issue on Applications of Metaheuristics, European Journal of Operational Research. Springer Editors, Volume 179, Issue 3, page 906-922, 2007
- AZZAG H., GUINOT C., VENTURINI G., PICAROUGNE F., **On data clustering with a flock of artificial agents**. Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 04), P.777-778, Boca Raton, Florida, USA, 2004.
- BERRY, M., LINOIFF, G. **Data mining techniques for marketing, sales and customer support**. NEW YORK (NY). John Wiley & Sons, 1997.
- BERSON, A; SMITH, S; THEARLING, K., **Building Data Mining Applications for CRM**. NOVA YORK, EUA: MCGRAW-HILL, 1999.
- BISHOP C. M. , **Pattern Recognition and Machine Learning**, Springer, 2006
- BONABEAU E.; DORIGO, M.; THERAULAZ, T., **Swarm Intelligence: From Natural to Artificial Systems**, New York: Oxford University Press. 1999

- BONISSONE, P. P., CHEN, Y., GOEBEL, T. K. & KHEDKAR, P. S., **Hybrid soft computing systems: industrial and comercial applications**, Proceedings of the IEEE,87(9), 1641-1667, 1999.
- BULLNHEIMER, B., HARTI, R. F., STRAUSS, C.. **Applying the Ant System to the Vehicle Routing Problem**. In 2nd International Conference on Metaheuristics, pages 699–719, Sophia-Antipolis, France.1997
- CALINSKI, T. e HARABASZ, J. **A dendrite method for cluster analysis**. Communications in statistics, 3(1): 1-27, 1974.
- COELHO, L.S., **Fundamentos, Potencialidades e Aplicações de Algoritmos Evolutivos** SBMAC - São Carlos, SP :, 2003.
- COELHO, P.S. S., EBECKEN, N. F. F., **Segmentação de Dados em um Número Desconhecido de Grupos Usando Algoritmos Genéticos**. Anais do XXXIII Simpósio Brasileiro de Pesquisa Operacional, Campos do Jordão, SP, 2001.
- COELHO, P.S. S., **Um Sistema par Indução de Modelos de Predição Baseados em Árvores**. Tese D. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2005.
- COELHO, P.S. S., **Data Mining: Algumas Questões Epistemológicas**.Anais do IX Simpósio de Pesquisa Operacional e Logística da Marinha, Rio de Janeiro, RJ,2006.
- COLE, R. M., **Clustering with Genetic Algorithms**. MSc thesis, University of Western Australia,1998.
- COLORNI A., DORIGO, M., MANIEZZO, V., M. TRUBIAN, **Ant System for Job Shop Scheduling**, Belgian Journal of Operations Research, Estatistic, and Computer Science 34, 39-53,1994.
- COUTINHO, F. V. Datamining; www.dwbrasil.com.br 2003. Acessado em 15dez 2006.

- CUNHA, A. G. G. ; ESPENCHITT, D. G. ; LACHTERMACHER, G. . **Pruning Techniques em Redes Neurais: Cuidados Preventivos** . In: SIO2001 Simposio Argentino em investigacion Operativa, 2001, Buenos Aires. Anales JAIIO, 2001.
- DA SILVA, E. B., **Agrupamento Semi-supervisionado de Documentos XML**, Tese D. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2006.
- DENEUBOURG, J.L.,GOSS, S., Franks N., SENDOVA, F. A., Detrain, C., CHR'ETIEN L.,**The Dynamics of Collective Sorting Robot-Like Ants and Ant-LikeRobots**. From Animals to Animats: Proc. of the 1st Int. Conf. on Simulationof Adaptive Behaviour. 1990.
- DORIGO, M ., MANIEZZO,V., COLORNI, A., **Positive feedback as a search strategy**, Technical Report 91-016, Dipartimento di Elettronica e Informazione,Politecnico di Milano, Italy, 1991.
- Dorigo M., Di Caro G., Gambardella L.M., **Ant Algorithms for Discrete Optimization**, Artificial Life, Vol. 5, N. 2, 1999
- DORIGO, M., STÜTZLE, T., **Ant Colony Optimization**, The Massachusetts Institute of Technology Press, Massachusetts, USA, 2004.
- DORIGO,M., **Behavior of real ants.**,
<http://iridia.ulb.ac.be/~mdorigo/ACO/RealAnts.html>. Acessado em ago/2006.
- EBERHART,R. C., KENNEDY, J., **A new optimizer using particle swarm theory**. in “Proceedings of the Sixth International Symposium on Micro Machine and Human Science”, Nagoya, Japan, Piscataway, NJ: IEEE Service Center, pp.39-43, 1995 .
- ESPENCHITT, D. G.,**Uma Nova Visão no Uso de Redes Neurais na Previsão de Falência de Empresas**. Tese M. Sc., UFF, Niterói, RJ, Brasil, 2000.

- ESPENCHITT, D. G., LACHTERMACHER, G. & GOMES, L. F. A. M, **O uso de Redes Neurais em Previsão com a Priorização das Variáveis, Utilizando as Ferramentas do Apoio Multicritério à Decisão**, Trabalho aprovado para a XII Escuela de Perfeccionamiento en Investigacion Operativa, Córdoba, Argentina. 2000
- ESTIVILL-CASTRO, V., **Why so many clustering algorithms – A position Paper**, ACM SIGKDD Explorations Newsletter, v. 04, Issue 1, pp 65-75, 2002.
- FAYYAD, U., PIATETSKY-SHAPIO, G.; SMYTH, P., **From Data Mining to Knowledge Discovery : An Overview. In : Advances in Knowledge Discovery and Data Mining**. Mit Press 1a. Ed. 1996
- FISHER R. A., **The use of multiple measurements in taxonomic problems**. Annals of Genics, 7:179 188. 1936
- FREITAS, A. A., **A survey of evolutionary algorithms for data mining and knowledge discovery**. Advances in Evolutionary Computation, A. Ghosh & S. Tsutsui (eds.), Springer-Verlag, 2001.
- GOLDBERG, D.E., **Genetic algorithms in search, optimization, and machine learning**. Addison Wesley: Reading, MA, USA, 1989.
- HAIR, J.F. JR, A., R. F., TATHAM, R. L., **Multivariate Data Analyses**. 5 ed. Prentice Hall, Inc, USA, 1998.
- HAIR, J.F. Jr, BLACK. J.F. , **Cluster analysis**. In: Laurence G. Grimm & Paul R. Yarnold (Eds.), 2000
- HAN, J., KAMBER, M., **Data Mining Concepts and Techniques**. Morgan Kaufmann Publishers, San Francisco, USA, 2006.

- HANDL, J., **Ant-based methods for tasks of clustering and topographic mapping: extensions, analysis and comparison with alternative techniques.** Masters Thesis, Universität, Erlangen-Nürnberg, Erlangen, Germany, 2003 .
- HANDL, J., KNOWLES, J., DORIGO, M., **On the performance of ant-based clustering. Design and application of hybrid intelligent systems.** Frontiers in Artificial intelligence and Applications 104. Pages 204-213. 2003.
- HANDL, J., KNOWLES, J., **Multiobjective clustering with automatic determination of the number of clusters.** Technical Report TR-COMPSYSBIO-2004-02. UMIST, Manchester, UK, 2004_a.
- HANDL, J., KNOWLES, J., **Evolutionary multiobjective clustering.** Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature (PPSN VIII). Pages 1081-1091. LNCS 3242. Springer-Verlag, 2004_b.
- HANDL, J., KNOWLES, J., **Exploiting the trade-off -- the benefits of multiple objectives in data clustering.** Third International Conference on Evolutionary Multi-Criterion Optimization, 2005.
- HANDL, J., KNOWLES, J., DORIGO, M., **Ant-based clustering and topographic mapping.** Artificial Life 11.2., 2005.
- HANDL, J., MEYER, B., **Ant-based and swarm-based clustering.** Proceedings of the IEEE Swarm Intelligence Symposium, SIS 2007. 2007.
- HAYKIN, S., **Neural Networks - A comprehensive Foundation** - Macmillan College Publishing Company, 2nd ed, 1999.
- HOLLAND, J., **Adaptation in Natural and Artificial Systems.** University of Michigan Press, 1975.
- JAIN, A.K., MURTY, M.N., FLYNN, P.J., **Data Clustering: A Review,** ACM Computing Surveys, Vol. 31, No. 3, September 1999.

KAUFMAN, L., ROUSSEEUW, P. J. **Finding Groups in Data: An Introduction to Cluster Analysis**. New York: John Wiley & Sons, 1990.

KENNEDY, J., EBERHART, R. C., **Particle swarm optimization**. in “Proceedings of the IEEE International Conference on Neural Networks”, Piscataway, NJ:IEEE Service Center, pp. 1942-1948, 1995.

KENNEDY, J., EBERHART, R. C., **Swarm Intelligence**, Morgan Kaufmann Publishers, ISBN: 1-55860-595-9, 2001

KRINK, T., URSEM, R. K., THOMSEN R., **Ant systems and particle swarm optimization**, EVALife Course, Faal 2002, Topics of Evolutionary Computation, ALife Group, Dept. of Computer Science, University of Aarhus EV, Denmark,2002.

LABROCHE, N., MONMARCHÉ, N., VENTURINI G., **A new clustering algorithm based on the chemical recognition system of ants**. Proceedings of the European Conference on Artificial Intelligence, 2002.

LACHTERMACHER, G. ; ESPENCHITT, D. G. . **Previsão de Falência de Empresas: Estudo de Generalização de Redes Neurais** . In: XXV ENANPAD, 2001, Campinas. XXV ENANPAD, 2001.

LIU, Z., JIN, X., BIE, R., GAO,X.,**FAISC: a Fuzzy Artificial Immune System Clustering Algorithm**. Third International Conference on Natural Computation (ICNC 2007) IEEE, 2007.

LIANG, J. J., QIN, A. K., SUGANTHAN, P. N., BASKAR, S., **Particle Swarm Optimization Algorithms with Novel Learning Strategies**, Int. Conf. on Systems, Man and Cybernetics (SMC2004), The Hague, The Netherlands, Oct. 2004.

- LUMER E.D., FAIETA B., **Diversity and adaptation in populations of clustering ants**. In Proceedings of the Third International Conference on Simulation of Adaptive Behaviour, pages 501–508, 1994.
- MACHADO FILHO, O. M., **Exploração e Análise de Agrupamento de Dados**. Tese M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2002.
- MANIEZZO, V., CARBONARO A., **Ant Colony Optimization: an Overview**. In Ribeiro, C., ed.: Essays and Surveys in Metaheuristics, Kluwer Academic Publishers, 2001.
- MANIEZZO, V., COLORNI, A., **The ant system applied to the quadratic assignment problem**, IEEE Transactions on Knowledge and Data Engineering, 11(5), 769-778, 1999.
- MONMARCHÉ, N., **On data clustering with artificial ants**. A.A. Freitas, editor, AAAI-99 & GECCO-99 Workshop on Data Mining with Evolutionary Algorithms: Research Directions, pages 23-26, Orlando, Florida, July 18, 1999.
- MORAES, D. R. S. **Inteligência Computacional na Classificação Litológica**. Tese M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2004.
- NOGUEIRA, C. F., **Metodologia de Valorização de Clientes Utilizando Mineração de Dados** - Tese D. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2004.
- NOVAES, U. R., **Agrupamento de Dados através de algoritmos Swarm**. Tese M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2002.
- NUANNUAN, Z., GUI, F., ADJOUADI, M., **A New Clustering Algorithm of Large Datasets with $O(N)$ Computational Complexity**. Proceedings of the 2005 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), IEEE .2005

- PARPINELLI, R. S **Algoritmo Baseado em Colônias de Formigas para Classificação em Data Mining**. Dissertação M. Sc., CEFET/PR, Curitiba, PR, Brasil, 2001.
- PARPINELLI, R. S., LOPES, H. S., FREITAS, A. A., **An ant colony based system for data mining: applications to medical data**. Proc. Genetic and Evolutionary Computation Conf. (GECCO-2001), pp. 791-798. Morgan Kaufmann, 2001.
- PARPINELLI, R. S., LOPES, H. S., FREITAS, A. A., **Data Mining with an Ant Colony Optimization Algorithm**. To appear in IEEE Trans. on Evolutionary Computation, special issue on Ant Colony algorithms. Volume 6, Number 4 August 2002.
- PAKHIRA, M. K., BANDYOPADHYAY, S., MAULIK, U., **Validity Index For Crisp and Fuzzy Clusters**, Pattern Recognition, vol. 37, nº3, pp 487-501, 2004.
- PULIDO, G. T., COELLO, C. C., **Using Clustering Techniques to Improve the Performance of a Multi-Objective Particle Swarm Optimizer**. The Genetic and Evolutionary Computation Conference (GECCO-2004)., 2004.
- PUNTAR, S. G., **Métodos e Visualização de Agrupamento de Dados**, Tese M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2003.
- PYLE, D., **Data Preparation for Data Mining**. SCIENCE & TECHNOLOGY BOOKS, 1999.
- SALIM, M., YAO, X., **Evolving SQL queries for data mining**. in “Proceedings of the 3rd International Conference on Intelligent Data Engineering and Automated Learning”, IDEAL’02, pp.62-67, 2002.
- SANTOS R., Disponível em <http://www.lac.inpe.br/~rafael.santos/dmdata.jsp>>. Acessado em: 23/dez/2007
- SANTOS R., Disponível em <http://www.lac.inpe.br/~rafael.santos/Docs/ELAC/dm-elac-2x2.pdf>>. Acessado em: 10/jan/2008

SHI,Y., EBERHART,R. C., **A modified particle swarm optimizer**. in “Proceedings of the IEEE International Conference on Evolutionary Computation”, Piscataway, NJ: IEEE Press, pp. 69-73, 1998.

SHI,Y., EBERHART,R. C., **Empirical study of particle swarm optimization**.in “Proceedings of the Congress on Evolutionary Computation”, Piscataway,NJ: IEEE Service Center, pp. 1945-1950, 1999.

STÜTZLE, T., DORIGO, M., **ACO algorithm for the quadratic assignment problem**.in “New ideas in optimization”, (D. Corne, M. Dorigo & F. Glover (eds.),McGraw-Hill, 1999.

UCI Machine Learning Repository-<<http://ww.ics.uci.edu/~mlearn/>>Acessado em: 20 dez 2007

VEENHUIS, C., KÖPPEN, M., **Data Swarm Clustering**. Proceedings of 4th Interncional Conference on Intelligent System Desing and Applications (ISDA), Budapest, Hungary, August, 2004.

ZADROZNY, B., **Tópicos Avançados em Otimização e IA II Aprendizado Automático**, <http://www.ic.uff.br/~bianca/topicos>. Acessado em dez 2006.

WEISS, S.M., INDURKHYA, N., **Predictive Data Mining**: a practical guide. San Francisco (EUA): Morgan Kaufmann Publishers, 1998.

WEKA Machine Learning Project .Disponível em: <http://www.cs.waikato.ac.nz/~ml>
Acesso em: jun/2006.

WEKA Software. Weka 3 - Data Mining with Open Source Machine Learning Software in Java. Disponível em : <http://www.cs.waikato.ac.nz/ml/weka/>.
Acesso em: jun/ 2006.

WITTEN, I.H., FRANK, E., **Data Mining: Practical machine learning tools and techniques with Java implementations**. San Francisco: Morgan Kaufmann Publishers. 2nd Edition, 2005.