

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e Computação
Departamento de Engenharia de Computação e Automação Industrial



**Aprendizagem Participativa em Agrupamento Nebuloso de
Dados**

Leila Roling Scariot da Silva

Orientador: Prof. Dr. Fernando Antonio Campos Gomide

Dissertação apresentada à Faculdade de
Engenharia Elétrica e de Computação da
Universidade Estadual de Campinas como
requisito parcial exigido para a obtenção do
título de Mestre em Engenharia Elétrica

Banca Examinadora: Prof. Dr. Fernando Antonio Campos Gomide
(DCA/FEEC/ UNICAMP)

Prof. Dr. Evandro de Oliveira Araújo
(DELTA-EEUFMG)

Prof. Dr. Ivan Luiz Marques Ricarte
(DCA/FEEC/ UNICAMP)

Prof. Dr. Ricardo Ribeiro Gudwin
(DCA/FEEC/ UNICAMP)

Dissertação de Mestrado
Campinas – SP – Brasil
Março de 2003

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Si38a Silva, Leila Roling Scariot da
Aprendizagem participativa em agrupamento
nebuloso de dados / Leila Roling Scariot da Silva.--
Campinas, SP: [s.n.], 2003.

Orientador: Fernando Antonio Campos Gomide.
Dissertação (mestrado) - Universidade Estadual
de Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Inteligência artificial – Processamento de
dados. 2. Sistema de auto - organização. 3.
Conjuntos difusos. I. Gomide, Fernando Antonio
Campos. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Elétrica e de Computação.
III. Título.

Dedico ao Gilberto e à nossa filha Ana com amor.

Agradecimentos

Agradeço ao professor Fernando Gomide, pela orientação sempre oportuna e por ter acreditado neste trabalho.

Agradeço ao CNPq pelo suporte financeiro.

Agradeço ao professor Fernando Von Zuben pelo incentivo e apoio oferecido no início do mestrado.

Agradeço ao meu marido Gilberto por ter me fornecido apoio e força em tantas ocasiões.

Aos meus pais Guilherme e Inês pelo amor e pelo carinho infinito.

Aos meus amigos do LCA, DENSIS e DT. Em especial a Ivette, Giselle, Marina, Ivana, Edgar, Vinicius, Márcia, Simone, Cris, Rosangela, ..., pelo carinho e pelas valiosas discussões e sugestões para a realização deste trabalho.

E finalmente a Deus, pela vida e pelas oportunidades que ela me ofereceu.

Resumo

Este trabalho apresenta um algoritmo para agrupamento nebuloso de dados, derivado do paradigma de aprendizagem participativa introduzido por Yager na década de noventa. O algoritmo de aprendizagem participativa utiliza a distância de Mahalanobis como medida de similaridade e parâmetros de controle para o agrupamento não supervisionado de dados. O desempenho do algoritmo é verificado em duas circunstâncias. A primeira considera diferentes conjuntos de dados sintéticos e a segunda utiliza o conjunto de dados real, *Iris*. Em ambos os casos, o algoritmo determina o número de classes e os respectivos centros com sucesso. O algoritmo também se mostra competitivo quando comparado com cinco outros importantes algoritmos de agrupamento nebuloso: algoritmo C-Means Nebuloso (FCM), Gustafson-Kessel (GK), Aprendizagem Competitiva Nebulosa (FCL), Aprendizagem Competitiva Elíptica Nebulosa Sensível a Frequência (FFSECL) e K-Means Nebuloso Modificado (MFKM).

Abstract

This work presents a new unsupervised fuzzy clustering algorithm based on the concept of participatory learning suggested by Yager in the nineties. The original participatory learning procedure is translated into an efficient algorithm to handle data clustering. The current version uses the Mahalanobis distance as a similarity measure and extra parameters to control learning behavior. The performance of the algorithm is verified in two circumstances. The first uses several representative synthetic data sets whereas the second uses the well-known Iris data. In both circumstances the participatory learning algorithm determines the expected number of clusters and the corresponding cluster centers successfully. Comparisons with Fuzzy C-Means (FCM), Gustafson-Kessel (GK), Fuzzy Competitive Learning (FCL), Fuzzy Frequency Sensitive Elliptical Competitive Learning (FFSECL) and Modified Fuzzy K-Means fuzzy (MFKM), are included to show the effectiveness of the participatory approach in data clustering.

Conteúdo

Agradecimentos	iii
Resumo	v
Abstract	v
Conteúdo	vii
Lista de Figuras	ix
Lista de Tabelas	xi
Notação	xiii
Lista de Abreviaturas	xv
1.Introdução	1
1.1 Motivação e Relevância	1
1.2 Reconhecimento de Padrões	1
1.3 Objetivos e Organização	2
2.Agrupamento de Dados	5
2.1 Introdução	5
2.2 Normas e Critérios de Distância	10
2.3 Categorias de Algoritmos de Agrupamento	16
2.4 Métodos de Agrupamento por Partição	20
2.5 Algoritmos Nebulosos para Agrupamento de Dados	23
2.6 Métodos de Validação de Grupos	35
2.7 Resumo	40
3.Aprendizagem Participativa	41
3.1 Introdução	41
3.2 Fundamentos da Aprendizagem Participativa	42
3.3 Sistema de Aprendizagem Participativa	45
3.4 Resumo	49
4.Aprendizagem Participativa em Agrupamento	51
4.1 Introdução	51
4.2 Algoritmo A - Aprendizagem Participativa	51
4.3 Algoritmo B - Aprendizagem Participativa com Matriz de Covariância	55
4.4 Discussão sobre os Algoritmos	59
4.5 Resumo	62
5.Resultados e Discussões	63
5.1 Introdução	63
5.2 Requisitos e Critérios de Desempenho	63

5.3	Características dos Conjuntos de Dados	66
5.4	Resultados Experimentais	72
5.5	Discussões	87
5.6	Discussões sobre os Resultados das Funções de Validação	95
5.7	Análise do Algoritmo AP-B	96
5.8	Resumo	97
6. Conclusões		99
Referências Bibliográficas		101
Índice Remissivo por Autores		105
Apêndice I		107
Apêndice II		109
Apêndice III		115

Lista de Figuras

Figura 1.1: Reconhecimento de padrões versus agrupamento de dados.....	2
Figura 2.1: Árvore de tipos de classificação.....	7
Figura 2.2: Distância Euclidiana versus Mahalanobis.....	13
Figura 2.3: Possíveis efeitos de uma normalização.....	15
Figura 2.4: Exemplo de um dendograma.....	18
Figura 2.5: Exemplo de grade em um conjunto de dados bidimensional.....	19
Figura 2.6: Partições.....	21
Figura 2.7: AO versus ACE.....	22
Figura 2.8: (a) Conjunto de dados original e (b) a grade gerada com 3×3 de células.....	32
Figura 3.1: Estrutura da aprendizagem participativa.....	41
Figura 3.2: Aprendizagem participativa com índice de alerta.....	46
Figura 3.3: Sistema de aprendizagem participativa.....	48
Figura 4.1: Exemplo de centros redundantes.....	55
Figura 5.1: Conjunto de dados originais de a, b, c.....	67
Figura 5.2: Conjunto de dados originais de d, e, f.....	68
Figura 5.3: Conjunto de dados originais de g, h, i.....	69
Figura 5.4: Conjunto de dados originais de j, l, m.....	69
Figura 5.5: Conjunto de dados originais de n, o, p.....	70
Figura 5.6: Agrupamento para o conjunto de dados da Figura 5.1.a.....	73
Figura 5.7: Agrupamento para o conjunto de dados da Figura 5.1.b.....	74
Figura 5.8: Agrupamento para o conjunto de dados da Figura 5.1.c.....	74
Figura 5.9: Agrupamento para o conjunto de dados da Figura 5.2.d.....	75
Figura 5.10: Agrupamento para o conjunto de dados da Figura 5.2.e.....	75
Figura 5.11: Agrupamento para o conjunto de dados da Figura 5.2.f.....	76
Figura 5.12: Agrupamento para o conjunto de dados da Figura 5.3.g.....	76
Figura 5.13: Agrupamento para o conjunto de dados da Figura 5.3.h.....	77
Figura 5.14: Agrupamento para o conjunto de dados da Figura 5.3.i.....	77
Figura 5.15: Agrupamento para o conjunto de dados da Figura 5.4.j.....	78
Figura 5.16: Agrupamento para o conjunto de dados da Figura 5.4.l.....	78
Figura 5.17: Agrupamento para o conjunto de dados da Figura 5.4.m.....	79
Figura 5.18: Agrupamento para o conjunto de dados da Figura 5.5.n.....	79
Figura 5.19: Agrupamento para o conjunto de dados da Figura 5.5.o.....	80
Figura 5.20: Agrupamento para o conjunto de dados da Figura 5.5.p.....	80
Figura 5.21: (a) Conjunto de dados original, com 50 pontos. (b) Conjunto de dados original, com 1000 pontos.....	82
Figura 5.22: Tempo de processamento dos algoritmos em relação ao número de pontos.....	83
Figura 5.23: (a) Conjunto de dados original, com 2 grupos. (b) Conjunto de dados original, com 9 grupos.....	84

Figura 5.24: Tempo de processamento dos algoritmos em relação ao número de grupos.	85
Figura 5.25: (a) Conjunto de dados original, com 2 dimensões. (b) Conjunto de dados original, com 3 dimensões.	85
Figura 5.26: Tempo de processamento dos algoritmos em relação à dimensão.	86
Figura 5.27: Algoritmos com sensibilidade à ordem de apresentação dos dados.	90
Figura 5.28: Algoritmos com sensibilidade ao número de atributos.	91
Figura 5.29: Algoritmos com sensibilidade à inicialização dos centros dos grupos.	92
Figura 5.30: Algoritmos com sensibilidade à densidade dos dados.	93
Figura 5.31: Algoritmos com sensibilidade à variação da densidade entre os grupos.	94

Lista de Tabelas

Tabela 2.1: Métricas.....	12
Tabela 2.2: Normalização.....	15
Tabela 2.3: Principais características dos algoritmos de agrupamento.....	34
Tabela 2.4: Funções de validação.....	39
Tabela 4.1: Principais características dos algoritmos A e B.....	58
Tabela 4.2: Características funcionais de cada algoritmo.....	59
Tabela 5.1: Descrição dos conjuntos de dados.....	71
Tabela 5.2: Tempo médio de processamento dos algoritmos.....	81
Tabela 5.3: Tempo de processamento dos algoritmos em relação ao número de pontos.....	83
Tabela 5.4: Tempo de processamento dos algoritmos em relação ao número de grupos.....	84
Tabela 5.5: Tempo de processamento dos algoritmos em relação à dimensão.....	86
Tabela 5.6: Requisitos e critérios de análise de desempenho dos algoritmos.....	87

Notação

Variáveis

- X – conjunto finito de dados $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, $\mathbf{x}_k \in \mathfrak{R}^p$, $k = 1, \dots, n$
- \mathbf{x}_k – k -ésimo elemento de X , $k = 1, \dots, n$
- p – número de componentes de \mathbf{x}_k
- n – número de elementos de X
- c – número de grupos
- \mathbf{x}_{kj} – j -ésimo componente de \mathbf{x}_k , $j = 1, \dots, p$
- U – matriz de pertinência ($n \times c$)
- u_{ki} – elemento da matriz U , $k = 1, \dots, n$, $i = 1, \dots, c$
- V – conjunto dos centros dos c grupos $\{\mathbf{v}_1, \dots, \mathbf{v}_c\}$, $\mathbf{v}_i \in \mathfrak{R}^p$
- v_{ij} – elemento da matriz V , $i = 1, \dots, c$, $j = 1, \dots, p$

Lista de Abreviaturas

- FCM : Algoritmo Fuzzy C-Means
- GK : Algoritmo Gustafson Kessel
- FCL : Algoritmo de Aprendizagem Competitiva Nebulosa
- FFSECL : Algoritmo de Aprendizagem Competitiva Elíptica
Nebulosa Sensível à Frequência
- MFKM : Algoritmo K-Means Nebuloso Modificado
- AP : Algoritmo de Agrupamento com Aprendizagem
Participativa
- AO : Otimização Alternada
- ACE : Estimação Alternada de Grupos

1. Introdução

1.1 Motivação e Relevância

O problema de agrupamento de dados tem sido encontrado em muitos contextos por pesquisadores das mais diversas áreas, refletindo a importância de se explorar e analisar os dados. As aplicações para as técnicas de agrupamento de dados são as mais variadas, incluindo a comunicação homem-máquina, reconhecimento automático de objetos, medicina, transporte, geoprocessamento, estudo de recursos naturais (agricultura, florestas, geologia, meio ambiente), indústrias (teste, inspeção e controle de qualidade em montagem, modelagem e controle de processos) (Bezdek and Pal, 1992), psiquiatria (por exemplo agrupar clientes com quadros de depressão), arqueologia (Cole, 1998), recuperação de informação (*information retrieval*) (Jain *et al.*, 1999) e mineração de dados (Karypis *et al.*, 1999), e muitas outras.

Contudo, os métodos de agrupamento existentes possuem limitações e restrições de aplicação, como é observado por Araújo (2001). Os principais algoritmos tais como o FCM (Bezdek, 1981), GK (Gustafson and Kessel, 1979), FCL (Backer and Scheunders, 1999a) e o FFSECL (Backer and Scheunders, 1999b) necessitam conhecer o número de grupos contidos no conjunto de dados. O algoritmo proposto neste trabalho é capaz de encontrar o número de grupos em um conjunto de dados, e de capturar formas de grupos esféricas e elipsoidais.

1.2 Reconhecimento de Padrões

Reconhecimento de padrões é a área que tem o propósito de procurar, detectar e explicitar estruturas associadas às regularidades ou propriedades presentes em um conjunto de dados. Por ser uma atividade muito complexa, o reconhecimento de padrões pode ser decomposto em vários subproblemas, aqui chamados de etapas, tornando-o deste modo mais compreensível. Uma de suas etapas é a análise de classes, a qual requer

técnicas de agrupamento de dados, sendo esta a área de estudo abordada neste trabalho. Podemos, assim resumir o relacionamento entre reconhecimento de padrão e o agrupamento de dados adotado neste trabalho mostrado na Figura 1.1.

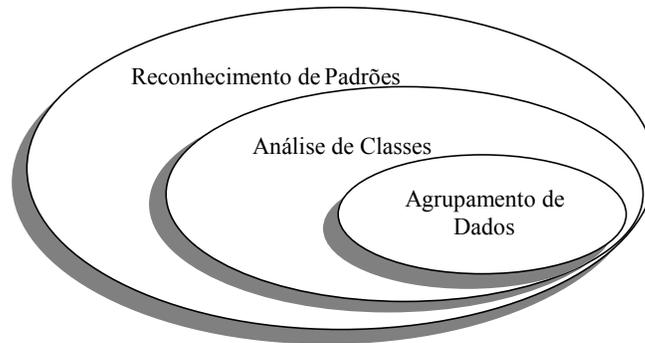


Figura 1.1: Reconhecimento de padrões versus agrupamento de dados.

Para realizar o reconhecimento de padrões é necessário elaborar a análise de classes a partir de agrupamento de dados como mostra a Figura 1.1.

Os algoritmos de agrupamentos são procedimentos de exploração que procuram estruturas “naturais” em um conjunto de dados. O agrupamento envolve classificação de dados, ou pontos, em grupos, ou classes, de modo que os pontos de um mesmo grupo são mais semelhantes entre si que com aqueles pontos de grupos distintos (Cole, 1998).

Em geral, após a partição de um conjunto de dados em c grupos, associa-se à cada um destes grupos uma classe. Por esta razão, freqüentemente usa-se o termo *classe* como sinônimo de *grupo*. Claramente, o agrupamento de dados constitui uma das principais e mais complexas etapas do processo de reconhecimento de padrões.

1.3 Objetivos e Organização

O objetivo deste trabalho é apresentar um novo algoritmo nebuloso de agrupamento de dados inspirado no conceito de aprendizagem participativa. O principal propósito é obter um algoritmo para o agrupamento não supervisionado de dados, onde a determinação do melhor ou de um número aceitável de classes e dos centros que as representam é parte do próprio algoritmo.

Este trabalho também discute o conceito de aprendizagem participativa. Este conceito de aprendizagem, introduzido por Yager (1990), permite a um sistema aprender sobre um ambiente externo com o auxílio, isto é, com a participação das crenças do próprio sistema. Traduzindo este modelo para o contexto de agrupamento de dados, isto significa que o algoritmo deve utilizar sua estrutura de conhecimento juntamente com a organização implícita nos dados para aprender sobre a própria estrutura dos dados.

O trabalho é dividido em seis capítulos. Após esta introdução, o Capítulo 2 aborda o estado da arte sobre agrupamento, enfatizando os principais algoritmos propostos na literatura. O Capítulo 3 descreve os conceitos fundamentais de aprendizagem participativa. A seguir, o Capítulo 4 apresenta os algoritmos de agrupamento aplicando os conceitos da aprendizagem participativa. No Capítulo 5 discutem-se os resultados e comparações com os principais algoritmos disponíveis na literatura. O Capítulo 6 conclui o trabalho resumindo suas principais contribuições e tópicos que ainda necessitam de desenvolvimentos futuros.

2. Agrupamento de Dados

2.1 Introdução

Este capítulo aborda os conceitos de agrupamento de dados, as suas categorias, e detalha os principais algoritmos de agrupamento nebuloso de dados existentes na literatura.

2.1.1 Definições

Os termos e notações a seguir são utilizados durante o decorrer deste trabalho.

1) Dado, padrão, objeto, ponto ou observação: denotado por x , é representado por um vetor com p componentes.

2) Dados rotulados: são aqueles que assumem valores em um mesmo espaço vetorial multidimensional, acompanhados da classe a que cada dado pertence, podendo haver múltiplas classes, com variâncias e números de dados distintos ou não, para cada classe (Von Zuben, 2002).

3) Dados não rotulados: são aqueles que assumem valores em um mesmo espaço vetorial multidimensional e que não se conhece, *a priori*, as classes a que pertencem, embora cada um pertença a uma classe específica. O número de classes pode ser conhecido *a priori* ou não. A variância e o número de dados de cada classe pode diferir ou não (Von Zuben, 2002).

4) Atributo ou característica: valor de um componente x_i de um vetor x .

5) Conjunto de dados: denotado por $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathfrak{R}^p$, sendo \mathfrak{R} o conjunto dos números reais. O k -ésimo elemento de X é representado por $\mathbf{x}_k = (x_1, \dots, x_p)^T$, onde T denota transposto.

6) Grupo: um grupo é um subconjunto homogêneo de pontos de um conjunto X de pontos. Os pontos que pertencem a um mesmo grupo devem ser

similares, ou próximos, e os pontos de diferentes grupos o mais dissimilares, ou distantes, possíveis. Uma outra definição proposta por Jain e Dubes (1988) é: "Grupos são descritos como regiões de um espaço multidimensional contendo uma alta densidade de pontos, separados entre si por regiões contendo uma baixa densidade de pontos".

7) Agrupamento e classificação:

Na literatura, a diferença entre agrupamento e classificação se dá quanto à capacidade dos algoritmos de agrupamento e classificação de responderem as seguintes perguntas:

- 1) Quantos grupos existem em um determinado conjunto de dados?
- 2) Quais são estes grupos?
- 3) A qual grupo pertence um ponto x do conjunto de dados?

Para Jain e Dubes (1988) o agrupamento é um caso especial de classificação. A Figura 2.1 sugere uma árvore com os tipos de classificação, onde cada nó da árvore define um modo de resolver o problema de classificação. As definições dos nós da Figura 2.1 são as seguintes:

- Classificação exclusiva versus não exclusiva. Uma classificação exclusiva é uma partição de um conjunto X , onde cada ponto pertence exclusivamente a um único grupo. Uma classificação não exclusiva, ou sobreposta, pode assumir que um ponto pertença a vários grupos. Um agrupamento nebuloso é um tipo de classificação não exclusiva, onde o ponto é associado a cada um dos grupos com um respectivo grau de pertinência.
- Classificação supervisionada versus não supervisionada. Uma classificação supervisionada aplica somente uma medida de similaridade entre os pontos para se determinar a classe de um ponto. Esta classificação utiliza categorias já rotuladas, isto é, conhece-se o número de grupos e quais são estes grupos. Uma classificação não supervisionada possui categorias ainda não rotuladas, isto é, os algoritmos não supervisionados

precisam responder a primeira e a segunda perguntas citadas acima. Em Jain *et al.* (1999) os algoritmos de agrupamento são algoritmos de classificação não supervisionados e necessariamente respondem a primeira questão, ou seja, os algoritmos determinam o número de classes nos conjuntos de dados fornecidos.

Os agrupamentos hierárquicos e por partição serão discutidos com detalhes na Seção 2.3.3.



Figura 2.1: Árvore de tipos de classificação.

Para outros (Gath and Geva, 1989; Geva, 1989; Geva, 1999; Zahid *et al.*, 1999) os algoritmos de agrupamento são divididos em supervisionados e não supervisionados. Os algoritmos de agrupamento supervisionados não respondem a primeira questão, pois precisam conhecer a priori o número de grupos. Os algoritmos não supervisionados, ao contrário, têm o propósito de encontrar o número de grupos. Os algoritmos de classificação respondem apenas a terceira questão. Portanto, ao contrário de Jain, a classificação torna-se um caso especial de agrupamento.

Este trabalho considera os algoritmos de agrupamento como supervisionados ou não supervisionados (Gath and Geva, 1989; Geva, 1989; Geva, 1999; Zahid *et al.*, 1999). Portanto, uma vez definido o número de grupos, os algoritmos que distribuem os elementos de X nestes grupos de forma aceitável

são algoritmos do tipo supervisionado. Quando o número de grupos não é fornecido a priori, o algoritmo é não supervisionado ou auto-organizado. Atualmente, a maioria dos algoritmos não supervisionados utilizam os algoritmos supervisionados como parte de um procedimento iterativo para a determinação do número de grupos. Objetivamente, o número de grupos depende do critério de validação adotado e das características dos atributos dos próprios elementos de X . Entretanto, exceto em casos triviais, a escolha do número de grupos é subjetiva, pois depende tanto da percepção do usuário, quanto das características e dos requisitos da aplicação. Portanto, pode-se observar a dificuldade e a importância de se obter o número de grupos.

8) Centro ou protótipo: é uma referência para o algoritmo determinar se um ponto pertence ou não a um grupo. O centro pode ser visto como o "centro de gravidade" de um grupo. Um centro denotado por $v_i \in \mathcal{R}^p$ representa o i -ésimo grupo. Cada ponto de X é associado a um grupo de acordo com uma medida de similaridade em relação aos centros dos grupos. $V = \{v_1, \dots, v_c\} \subset \mathcal{R}^p$ é o conjunto dos centros dos c grupos.

9) Atualização de centros por batelada: os centros são avaliados e/ou atualizados em bloco, ou seja após a avaliação de todo um conjunto de pontos. Os métodos por batelada são melhores quando se tem interesse apenas no resultado final, isto é, o melhor resultado é obtido no final da execução do algoritmo utilizado. Porém, esta técnica torna-se numericamente ou computacionalmente inviável quando o conjunto de dados é muito grande e a solução do problema é obtida através da minimização de uma função objetivo (Baraldi and Blonda, 1998).

10) Atualização de centros ponto-a-ponto: a atualização dos centros é feita junto com a avaliação de cada ponto. Neste caso, é possível atualizar ou o centro mais similar ao ponto analisado, ou todos os centros de acordo com os respectivos graus de pertinência do ponto. Os métodos ponto-a-ponto são necessários quando: a) os resultados precisam ser conhecidos em tempo real; b) os conjuntos de dados são grandes e os métodos por batelada se tornam

inviáveis em termos de memória ou tempo de processamento; e, c) os dados são processados de forma contínua o que impossibilita a aplicação do método por batelada (Baraldi and Blonda, 1998). Os resultados da atualização ponto-a-ponto tipicamente dependem da ordem de apresentação dos dados de entrada. Além disso, estes métodos são mais sensíveis à presença de ruídos (Baraldi and Blonda, 1998).

11) Ruído: segundo Han e Kamber (2001), os pontos que são completamente diferentes perante todos os grupos existentes em um conjunto X podem ser chamados de ruídos, de forma que é provável que estes pontos foram gerados por medidas ou execuções erradas.

12) Matriz de pertinência: matriz onde o valor de cada elemento é o grau de pertinência de cada ponto aos grupos existentes, sendo representada por uma matriz $U (n \times c)$, onde c é o número de grupos e n representa o número de dados em X . Portanto, cada elemento u_{ki} de U , $i=1, \dots, c$, $k=1, \dots, n$, fornece o grau de pertinência de $x_k \in X$ ao i -ésimo grupo. Esta pertinência poderá ser rígida, com valores em $\{0,1\}$ ou flexíveis, com valores no intervalo $[0,1]$. Estas restrições induzem tipos de partição de X , como será visto na Seção 2.3.4.

13) Medida de similaridade versus dissimilaridade: um método de agrupamento requer um índice de proximidade (semelhança, afinidade), ou uma associação estabelecida entre os pontos. Índices de proximidade podem ser obtidos via medidas de similaridade ou de dissimilaridade. Quanto maior a medida de similaridade, maior será a relação existente entre os pontos; quanto maior a medida de dissimilaridade, menor a semelhança entre pontos. A distância Euclidiana é um exemplo de medida de dissimilaridade, enquanto que o coeficiente de correlação é um exemplo de similaridade (Jain and Dubes, 1988).

14) Densidade: é a quantidade de dados em relação a alguma unidade do espaço. Por exemplo, 200 árvores por hectares. Para o agrupamento, os grupos são considerados como regiões do espaço onde a densidade de dados é maior se comparadas com as regiões vizinhas.

2.2 Normas e Critérios de Distância

Um aspecto importante a ser definido, para as técnicas de agrupamento, é o critério de similaridade ou dissimilaridade adotado. As métricas proporcionam meios para se definir uma medida de similaridade ou dissimilaridade entre os pontos. Ou seja, as métricas quantificam o quanto os pontos são próximos ou semelhantes. Esta seção tem por objetivo rever a noção de distância e descrever as principais métricas.

2.2.1 Definição de espaço métrico

Definição: (Domingues, 1982) Dado um conjunto $M \neq \emptyset$ e $d: M \times M \rightarrow \mathfrak{R}_+$, sendo \mathfrak{R}_+ o conjunto dos números reais positivos, indica-se por $d(\mathbf{x}, \mathbf{y})$ a imagem de um par genérico (\mathbf{x}, \mathbf{y}) através da função d . Diz-se que d é uma métrica sobre M se as seguintes condições são verificadas para quaisquer $\mathbf{x}, \mathbf{y}, \mathbf{z} \in M$:

$$(M_1) \quad d(\mathbf{x}, \mathbf{y}) = 0 \quad (\mathbf{x} = \mathbf{y}),$$

$$(M_2) \quad d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}),$$

$$(M_3) \quad d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y}).$$

Nestas condições, cada imagem $d(\mathbf{x}, \mathbf{y})$ recebe o nome de distância de \mathbf{x} a \mathbf{y} e o par (M, d) , onde d é uma métrica sobre M , de espaço métrico. Cada objeto de um espaço métrico será sempre referido como ponto desse espaço, seja ele um ponto em si mesmo, ou um número, ou ainda uma função ou um vetor. A propriedade (M_3) é conhecida como desigualdade triangular.

2.2.2 Exemplos de Métricas

1. Métrica *discreta* ou Métrica *zero-um* (Domingues, 1982). Dado $M \neq \emptyset$, define-se $d: M \times M \rightarrow \mathfrak{R}_+$ do seguinte modo:

$$d(\mathbf{x}, \mathbf{x}) = 0 \quad \text{e} \quad d(\mathbf{x}, \mathbf{y}) = 1, \quad \forall \mathbf{x}, \mathbf{y} \mid \mathbf{x} \neq \mathbf{y}. \quad (2.1)$$

É fácil provar que a função d , assim definida, é uma métrica. A verificação de (M_1) e (M_2) é direta. Para verificar o axioma (M_3) um dos casos é aquele em que $\mathbf{x} \neq \mathbf{y}$ e $\mathbf{y} = \mathbf{z}$. Nestes caso $d(\mathbf{x}, \mathbf{y}) = 1$, $d(\mathbf{x}, \mathbf{z}) = 1$ e $d(\mathbf{z}, \mathbf{y}) = 0$. Logo,

$$d(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y}).$$

2. Seja o conjunto de todas as p -uplas (seqüências finitas) (x_1, x_2, \dots, x_p) , onde $x_i \in \mathfrak{R}$. Sendo $\mathbf{x}=(x_1, \dots, x_p)$ e $\mathbf{y}=(y_1, \dots, y_p)$ pontos arbitrários, definem-se as três métricas mais conhecidas e aplicadas:

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= \sqrt{(x_1 - y_1)^2 + \dots + (x_p - y_p)^2}, \\ d_1(\mathbf{x}, \mathbf{y}) &= |x_1 - y_1| + \dots + |x_p - y_p|, \\ d_2(\mathbf{x}, \mathbf{y}) &= \max\{|x_1 - y_1|, \dots, |x_p - y_p|\}. \end{aligned} \quad (2.2)$$

De certa forma estas três métricas são equivalentes. As métricas d , d_1 e d_2 , apresentam as seguintes relações (Domingues, 1982):

$$d_2(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{y}) \leq d_1(\mathbf{x}, \mathbf{y}) \leq p d_2(\mathbf{x}, \mathbf{y}),$$

para quaisquer $\mathbf{x}, \mathbf{y} \in \mathfrak{R}^p$. De fato:

$$\begin{aligned} d_2(\mathbf{x}, \mathbf{y}) &= |x_i - y_i| \text{ para um certo, } i \ 1 \leq i \leq p. \text{ Então,} \\ d_2(\mathbf{x}, \mathbf{y}) &= |x_i - y_i| = \sqrt{(x_i - y_i)^2} \leq d(\mathbf{x}, \mathbf{y}) \\ d(\mathbf{x}, \mathbf{y}) &= \sqrt{(x_1 - y_1)^2 + \dots + (x_p - y_p)^2} \leq \\ &\leq \sqrt{|x_1 - y_1|^2 + \dots + |x_p - y_p|^2 + 2|x_1 - y_1||x_2 - y_2| + \dots + 2|x_{p-1} - y_{p-1}||x_p - y_p|} = \\ &= |x_1 - y_1| + \dots + |x_p - y_p| = d_1(\mathbf{x}, \mathbf{y}). \end{aligned} \quad (2.3)$$

Supondo $|x_i - y_i| = \max\{|x_1 - y_1|, \dots, |x_p - y_p|\}$, logo

$$|x_1 - y_1| \leq |x_i - y_i|, \dots, |x_p - y_p| \leq |x_i - y_i|.$$

Portanto,

$$d_1(\mathbf{x}, \mathbf{y}) = |x_1 - y_1| + \dots + |x_p - y_p| \leq p|x_i - y_i| = p d_2(\mathbf{x}, \mathbf{y}).$$

Existem outras métricas que podem ser utilizadas como uma medida de distância entre dois pontos, além das já citadas anteriormente. Sendo $\mathbf{x}=(x_1, \dots, x_p)$ e $\mathbf{y}=(y_1, \dots, y_p)$ pontos arbitrários do \mathfrak{R}^p , a Tabela 2.1 abaixo lista as mais importantes.

Tabela 2.1: Métricas.

Distância	Equação
Euclidiana	$d = \sqrt{\sum_{i=1}^p (x_i - y_i)^2}$ (2.4)
Euclidiana Média	$d = \sqrt{\frac{\sum_{i=1}^p (x_i - y_i)^2}{p}}$ (2.5)
Euclidiana Padronizada	$d = \sqrt{\sum_{i=1}^p \left(\frac{x_i - y_i}{s_i} \right)^2}$ (2.6)
Euclidiana Ponderada (Mahalanobis)	$d = \sqrt{(\mathbf{x} - \mathbf{y})^T A (\mathbf{x} - \mathbf{y})}$ (2.7)
Manhatan ou Hamming	$d = \sum_{i=1}^p w_i x_i - y_i $ (2.8)
Minkowsky	$d = \left[\sum_{i=1}^p w_i x_i - y_i ^k \right]^{1/k} \quad k \geq 1, x_i \neq y_i$ (2.9)
Gower	$d = -\log_{10} \left[1 - \frac{1}{p} \sum_{i=1}^p \frac{ x_i - y_i }{x_{\max_i} - x_{\min_i}} \right]$ (2.10)
Cattel	$d = \frac{2 \left(p - \frac{2}{3} \right) - d_e^2}{2 \left(p - \frac{2}{3} \right) + d_e^2}$ (2.11)
Tchebyshev	$d(x, y) = \max\{ x_1 - y_1 , \dots, x_p - y_p \}$ (2.12)

Na tabela 2.1 define-se:

- s_i : desvio-padrão da i -ésima coordenada, em (2.6).
- A : matriz de ponderação em (2.7). Os casos particulares mais importantes de A são:
 - $A=I$, onde I é a matriz identidade, tornando-se a distância Euclidiana.
 - $A= C^{-1}$, onde C é uma matriz de covariância, tornando-se a distância de Mahalanobis.
- w_i representa as ponderações para as variáveis em (2.8) e (2.9): os valores mais usados são os da equiponderação $w_i=1$ ou a média $w_i=1/p$.

- Casos especiais de k em (2.9):
 - i. Distância Euclidiana: $k=2, w_i = 1, i=1, \dots, p$;
 - ii. Distância de Manhattan : $k=1$;
 - iii. Distância Tchebyshev: $k=\infty, w_i = 1, i=1, \dots, p$.
- d_e^2 é a distância Euclidiana com variáveis padronizadas em (2.11), e p é o número de elementos de x .

A distância Euclidiana é a mais utilizada destas métricas. Contudo, a distância Euclidiana assume que os valores dos atributos do ponto não são correlacionados uns com os outros (Jain and Dubes, 1988). Em muitas aplicações esta característica não é desejável, como mostram os conjuntos do lado direito da Figura 2.2. A distância de Mahalanobis (Jain and Dubes, 1988) pode ser utilizada como alternativa. Neste caso, a distância entre os pontos x e y é:

$$d(x, y) = \sqrt{(x - y)^T C^{-1} (x - y)}. \quad (2.13)$$

O uso da distância de Mahalanobis remove algumas limitações da distância Euclidiana (Duda, 1997), por não depender das escalas dos atributos e correlaciona atributos com escalas diferentes. Por exemplo um atributo em cm com outro em km. Além disso, pode fornecer resultados com limites de decisão tanto curvos como lineares como por exemplo, detectar as duas formas da Figura 2.2 mostrada abaixo.

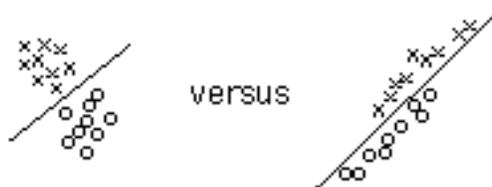


Figura 2.2: Distância Euclidiana versus Mahalanobis.

Contudo, a matriz de covariância pode ser difícil de ser calculada, pois a memória e o tempo de processamento crescem quadraticamente com o número de atributos, isto é, a dimensão dos dados. Este problema é evidentemente irrelevante somente no caso de poucos atributos.

2.2.3 Normas

Os algoritmos de agrupamento podem ser sensíveis à escala dos dados e à normalização. Portanto, estes fatores são abordados neste trabalho pois podem interferir nos resultados dos algoritmos de agrupamento.

A necessidade de adaptação de escalas com grandezas muito diferentes pode ser suprida através da normalização dos valores dos atributos. Uma *norma* sobre um conjunto E é uma função que associa a cada $\mathbf{u} \in E$ um número real não negativo, indicado por $\|\mathbf{u}\|$, e chamado *norma* de \mathbf{u} , de maneira que:

- (n₁) $\|\mathbf{u}\| = 0 \Leftrightarrow \mathbf{u} = 0$,
- (n₂) $\|\alpha \mathbf{u}\| = |\alpha| \|\mathbf{u}\|, \forall \alpha \in \mathfrak{R} \text{ e } \forall \mathbf{u} \in E$,
- (n₃) $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|, \forall \mathbf{u}, \mathbf{v} \in E$.

Se $\|\cdot\|$ é uma norma sobre E , então $d: E \times E \rightarrow \mathfrak{R}$, definida por $d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|$ é uma métrica sobre E , pois:

- $d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| = 0 \Leftrightarrow \mathbf{u} - \mathbf{v} = 0 \Leftrightarrow \mathbf{u} = \mathbf{v}$,
- $d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| = \|(-1)(\mathbf{v} - \mathbf{u})\| = |-1| \|\mathbf{v} - \mathbf{u}\| = \|\mathbf{v} - \mathbf{u}\| = d(\mathbf{v}, \mathbf{u})$,
- $d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| = \|\mathbf{u} - \mathbf{w} + \mathbf{w} - \mathbf{v}\| \leq \|\mathbf{u} - \mathbf{w}\| + \|\mathbf{w} - \mathbf{v}\| = d(\mathbf{u}, \mathbf{w}) + d(\mathbf{w}, \mathbf{v})$.

A métrica d assim obtida, chama-se *métrica induzida pela norma* dada sobre E .

Um exemplo importante é a norma Euclidiana:

$$\mathbf{x} = (x_1, \dots, x_p) \mapsto \|\mathbf{x}\| = \sqrt{x_1^2 + \dots + x_p^2}. \quad (2.14)$$

2.2.4 Normalização

A Tabela 2.2 apresenta as funções de normalização mais utilizadas. As funções seguintes referem-se a um atributo i , para um ponto x_{ki} .

Tabela 2.2: Normalização.

Função	Valores normalizados dos extremos da escala	Observações
$f(x_{ki}) = \frac{x_{ki} - x_{min_k}}{x_{max_k} - x_{min_k}}$	Máximo=1 Mínimo=0	Atributos de maximização
$f(x_{ki}) = \frac{x_{max_k} - x_{ki}}{x_{max_k} - x_{min_k}}$	Máximo=1 Mínimo=0	Atributos de minimização
$f(x_{ki}) = \frac{x_{ki}}{x_{max_k}}$	Máximo=1 Mínimo= x_{min_k} / x_{max_k}	Atributos de maximização

Contudo, uma normalização poderá ser perigosa para o processo de agrupamento. Por exemplo, se um padrão está presente em um determinado grupo, uma determinada normalização poderá alterar a relação das distâncias entre os pontos e alterar a separação existente entre dois grupos, como mostra na Figura 2.3 (Jain and Dubes, 1988).

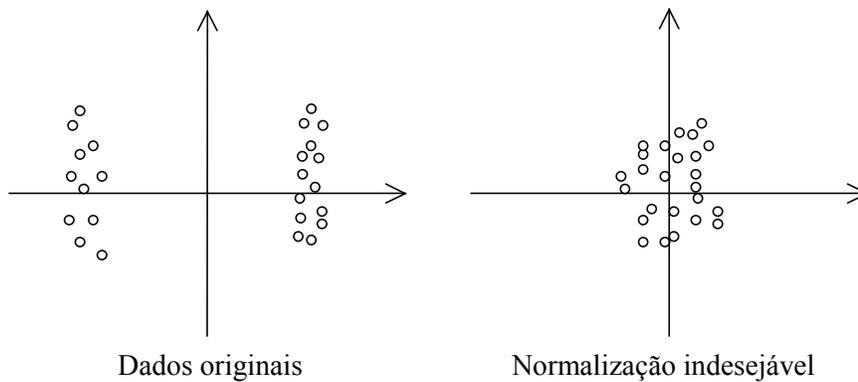


Figura 2.3: Possíveis efeitos de uma normalização.

É possível observar na Figura 2.3 que, o critério de similaridade e a norma adotada por um determinado algoritmo pode influenciar dramaticamente nos resultados de um agrupamento. Portanto, é necessário estudar estes temas, para não obter resultados indesejáveis.

Na próxima seção são apresentadas as categorias dos algoritmos de agrupamento.

2.3 Categorias de Algoritmos de Agrupamento

Uma multiplicidade de algoritmos de agrupamento é encontrada na literatura. Estes algoritmos podem ser classificados de acordo com várias características, sendo os mais importantes:

- Tipo de dados;
- Parâmetros do algoritmo;
- Critério de similaridade que determina um grupo;
- Conceitos e fundamentos em que se baseiam os algoritmos de agrupamento (e.g. conjuntos nebulosos, estatística, redes neurais, etc...).

2.3.1 Tipo de Dados

A característica *tipo de dados* se refere à capacidade do algoritmo agrupar ou não dados de diferentes naturezas: numérico, caracteres, palavra, texto, imagens, objetos, etc.

2.3.2 Parâmetros do algoritmo

Os conhecimentos *a priori* sobre as características do conjunto X , tais como o número de grupos, as classes desejadas, a matriz de pertinência, ou a falta deles, normalmente são fatores cruciais na escolha de um método de agrupamento.

Segundo Gath e Geva (1989), as três maiores dificuldades encontradas durante o agrupamento nebuloso de dados são:

1. Número de grupos, o que nem sempre pode ser definido *a priori*;
2. Características e localização dos centros dos grupos, o que não são necessariamente conhecidas *a priori*;
3. Grande variabilidade de formas, densidade, e número de dados em cada grupo.

Vê-se, portanto, que um dos principais problemas na tarefa de agrupamento é a estimação do número de grupos. Existem inúmeras propostas na literatura que tratam deste assunto, mas até o presente momento não há um procedimento que,

independentemente das características do conjunto de dados, detecte o número esperado de classes (Bezdek and Pal, 1992; Duda and Hart, 1973). Sabe-se que o problema de determinar o número ótimo de classes em um conjunto finito de dados é um problema NP-completo (Duda and Hart, 1973). Devido a esta característica, a maioria dos algoritmos, sejam eles nebulosos ou não, assume que o número c de classes em um conjunto finito X é conhecido, de modo que, cabe para as técnicas de validação, Seção 2.6, caracterizar o melhor agrupamento.

Como definido na Seção 2.1.1, este trabalho adota a definição Gath e Geva (1989) para agrupamento supervisionado e não supervisionado.

2.3.3 Critérios de similaridade ou dissimilaridade que determinam um grupo.

De acordo com a categoria de critério de similaridade ou dissimilaridade, os algoritmos podem ser classificados como (Jain *et al.*, 1999):

- *Agrupamento por partição*: visa obter diretamente uma partição de X . Mais especificamente, estes algoritmos determinam a partição utilizando a otimização de uma função objetivo que enfatiza a estrutura local e global dos dados. Para se encontrar uma solução ótima global, um agrupamento por partição requer uma busca exaustiva de todas as partições possíveis (Han and Kamber, 2001).
- *Agrupamento hierárquico*: classifica elementos de X em grupos em diferentes etapas, de modo hierárquico. O resultado destes algoritmos é uma árvore de grupos, chamada de dendrograma, como mostra a Figura 2.4. As técnicas hierárquicas podem ainda ser subdivididas em dois tipos: a primeira chamada de *aglomerativa*, ou aproximação *bottom-up*, onde fusões sucessivas dos $n, n-1, n-2$ pontos vão sendo obtidas, até reunir todos os pontos em um único grupo. A segunda técnica conhecida como *divisiva*, onde partindo de um único grupo, e por divisões sucessivas vão sendo obtidos 2,3, ... grupos menores, até cada ponto ser um grupo ou até obter uma outra condição de parada determinada. As técnicas divisivas também são conhecidas como aproximação *top-down*. O que caracteriza estes processos é a união de dois grupos em uma certa etapa, produzindo um dos agrupamentos da etapa superior caracterizando assim, um processo hierárquico. Os

dendogramas são impraticáveis para grandes conjuntos de dados (Jain and Dubes, 1988).

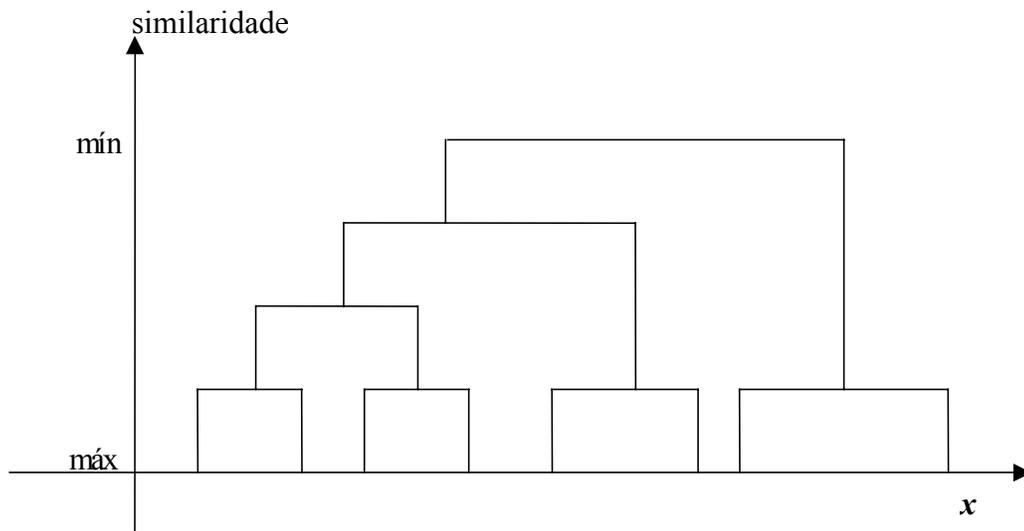


Figura 2.4: Exemplo de um dendograma.

- *Agrupamento baseado em densidade:* a idéia desta técnica é agrupar pontos vizinhos de um conjunto de dados baseado nas condições de densidade. Os grupos são as regiões densas de pontos e separados por regiões de baixa densidade de pontos.
- *Agrupamento baseado em grade:* este tipo de algoritmo é típico em *data mining*. Sua principal característica é subdividir o espaço em partições menores denominadas de células. Um exemplo de grade bidimensional é mostrado na Figura 2.5. Os algoritmos quantificam o espaço dentro de um número finito de células e realizam todas as operações de quantificação neste espaço, como por exemplo, calculam o número de pontos em cada célula e verificam onde há uma região de células "vazias" em torno de células "cheias", determinando assim um possível grupo.

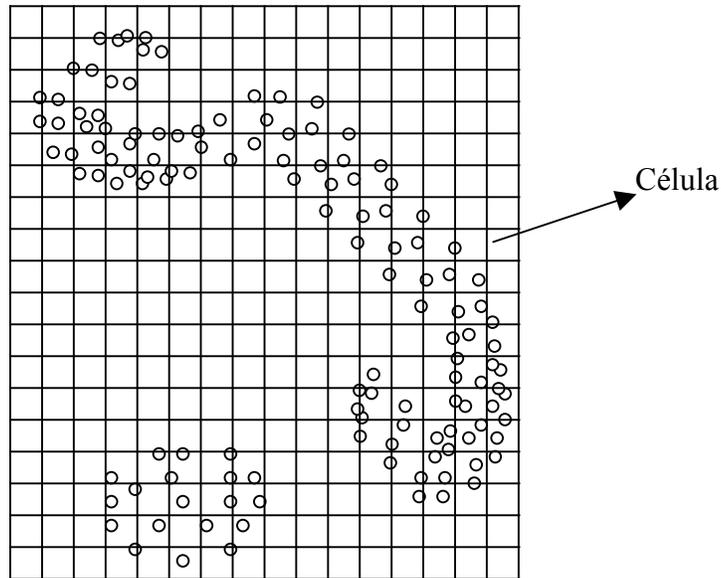


Figura 2.5: Exemplo de grade em um conjunto de dados bidimensional.

2.3.4 Conceitos e fundamentos

De acordo com a categoria dos *conceitos e fundamentos*, os algoritmos podem ser caracterizados como:

- *Nebulosos*: utilizam conjuntos nebulosos para classificar dados e consideram que um ponto pode ser classificado em mais de um grupo, mas com diferentes graus de associação. Estes tipos de algoritmos levam para um esquema de agrupamento que são compatíveis com experiências da vida cotidiana, pois tratam as incertezas dos dados reais. O mais representativo algoritmo de agrupamento nebuloso é o *Fuzzy C-Means* (FCM) (Bezdek, 1981).
- *Rígidos*: não consideram a sobreposição de grupos aos quais um ponto de X pertence. O algoritmo deverá resultar em um agrupamento com valores na matriz de pertinência restritos ao conjunto $\{0, 1\}$.
- *Neurais*: adotam abordagens conexionistas para o agrupamento. Em geral utilizam redes neurais artificiais com aprendizagem não supervisionada. Um exemplo típico é o método dos mapas organizáveis de Kohonen (1995).
- *Estatísticos*: Os Conceitos lógico-matemáticos empregados na análise estatística para agrupamento de dados são principalmente probabilísticos. Os

métodos estatísticos são conhecidos também como métodos de máxima verossimilhança (*maximum-likelihood*), sendo que a aproximação mais aplicada para o cálculo é a abordagem *Bayesiana*.

2.4 Métodos de Agrupamento por Partição

Este trabalho aborda algoritmos que adotam métodos de agrupamento por partição. Esta seção tem por objetivo caracterizar estes métodos.

O método de agrupamento por partição é formalmente definido da seguinte maneira: Tendo um número n de pontos de p dimensões, determinar uma partição M dos pontos em c grupos, ou classes ($c \leq n$), de modo que os pontos que pertencem a um mesmo grupo sejam similares, e os pontos de diferentes grupos sejam dissimilares em termos de seus atributos. O valor de c pode ou não ser conhecido *a priori*. (Han and Kamber, 2001).

Os algoritmos de agrupamento por partição dividem-se entre quatro categorias, de acordo com o tipo de partição (Runkler and Bezdek, 1999):

1. Partição rígida:

$$M_{hcn} = \left\{ U \in \{0,1\}^{nc} \mid \sum_{i=1}^c u_{ki} = 1, k = 1, \dots, n, \sum_{k=1}^n u_{ki} > 0, i = 1, \dots, c \right\}, \quad (2.15)$$

2. Partição nebulosa:

$$M_{fcn} = \left\{ U \in [0,1]^{nc} \mid \sum_{i=1}^c u_{ki} = 1, k = 1, \dots, n, \sum_{k=1}^n u_{ki} > 0, i = 1, \dots, c \right\}, \quad (2.16)$$

3. Partição nebulosa com ruído:

$$M_{ncn} = \left\{ U \in [0,1]^{nc} \mid 0 < \sum_{i=1}^c u_{ki} \leq 1, k = 1, \dots, n, \sum_{k=1}^n u_{ki} > 0, i = 1, \dots, c \right\}, \quad (2.17)$$

4. Partição possibilística:

$$M_{pcn} = \left\{ U \in [0,1]^{nc} \mid \sum_{i=1}^c u_{ki} > 0, k = 1, \dots, n \right\}. \quad (2.18)$$

Onde U é a matriz de pertinência. Observe na Figura 2.6, que $M_{hcn} \subset M_{fcn} \subset M_{ncn} \subset M_{pcn} \subset [0, 1]^{nc}$.

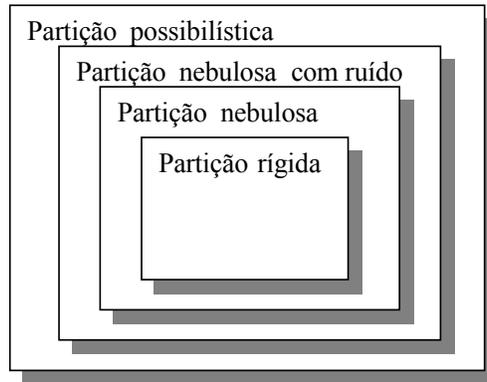


Figura 2.6: Partições.

Independente da matriz de pertinência, a maioria dos algoritmos de agrupamentos por partição possui em comum a obtenção dos grupos pela otimização de uma função objetivo. Estes algoritmos utilizam um mecanismo de otimização iterativa chamado de *otimização alternada* (*alternating optimization* - AO) (Runkler and Bezdek, 1999).

2.4.1 Estimação alternada de grupos

As funções de pertinência, obtidas através da minimização da função objetivo do modelo de agrupamento formulada como um problema de otimização, são restritas a uma forma particular para a atualização de U^l e V^l , onde U é a matriz de pertinência, V é a matriz dos centros dos grupos e l é a iteração atual. Contudo, há a necessidade de se estudar formas de função de pertinência que são consideradas mais adequadas para uma determinada aplicação. Neste caso o modelo de otimização para agrupamento pode ser substituído por um modelo mais genérico onde se especifica o mecanismo de atualização de U e V . Quando se utiliza mecanismos ou expressões para a atualização dos centros V e U que não sejam decorrentes da solução de um modelo de otimização, os algoritmos são denominados *algoritmos de estimação alternada de grupos* (*alternating cluster estimation* - ACE) (Runkler and Bezdek, 1999).

Para os modelos de agrupamento definidos como um problema de otimização de uma função objetivo J , é necessário distinguir entre o modelo (FCM) e o algoritmo para solucioná-lo (FCM-AO), pois a função objetivo pode, usualmente, ser otimizada com

uma variedade de diferentes algoritmos, tais como, gradientes, algoritmos genéticos, e0 esquemas híbridos, etc.. ACE, por outro lado, mantém uma estrutura mais genérica e não necessariamente aplica um modelo de agrupamento baseado em otimização. Desta forma, ACE pode ser chamado de um modelo ou uma arquitetura ou um algoritmo.

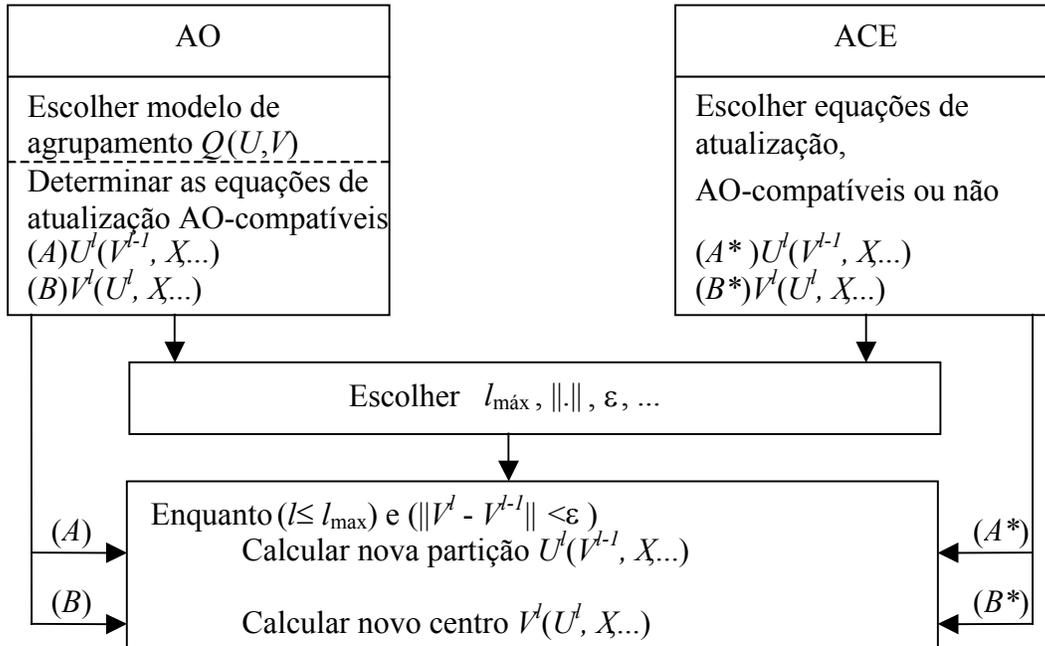


Figura 2.7: AO versus ACE.

O lado esquerdo da Figura 2.7 ilustra a relação entre o usuário AO, o modelo e o algoritmo. O ponto mais importante está no usuário AO escolher o modelo $Q(U, V)$, determina-se a forma de atualização da equação (A) e (B) que aparecem no bloco do *Enquanto* da Figura 2.7. No lado direito da Figura 2.7, a abordagem ACE abandona o modelo $Q(U, V)$, escolhe-se uma equação para atualizar (A*) e (B*), e segue-se para o bloco *Enquanto*. Um par (A, B) é AO-compatível se e somente se (A,B) são condições necessárias para o modelo de agrupamento $Q(U, V)$. Como na abordagem ACE pode-se escolher (A*) = (A) e (B*) = (B), é evidente que a abordagem AO é um caso particular de ACE. Na Figura 2.7 l_{\max} representa o número máximo de iterações e $||.||$ é uma norma.

A seguir os principais algoritmos de agrupamento de dados de interesse neste trabalho são apresentados.

2.5 Algoritmos Nebulosos para Agrupamento de Dados

Esta seção apresenta os principais algoritmos de agrupamento nebulosos propostos na literatura, resumindo as suas características, suas vantagens e desvantagens em relação a outros algoritmos. O algoritmo *Fuzzy C-Means* (FCM) é o principal algoritmo por partição. O algoritmo Gustafson Kessel (GK) é o algoritmo FCM com a distância de Mahalanobis como medida de similaridade. O algoritmo com aprendizagem competitiva nebulosa (FCL) e o algoritmo com Aprendizagem Competitiva Elíptica Nebulosa Sensível a Frequência (FFSECL) contêm em suas estruturas os conceitos de *aprendizagem competitiva* (Kohonen, 1995). Observe também que o algoritmo *Modificado Fuzzy K-Means* (MFKM) além de ser um método de agrupamento por partição, possui em sua estrutura os conceitos de hierarquia, grade e densidade, sendo portando relevante para este trabalho.

Os algoritmos conexionistas, tais como quantização vetorial e mapeamentos auto-organizáveis, têm propósitos semelhantes para o agrupamento. Contudo, estes algoritmos não serão abordados aqui por estarem fora do escopo deste trabalho.

2.5.1 Algoritmo Fuzzy C-Means (FCM)

O primeiro algoritmo de agrupamento nebuloso, desenvolvido em 1969 por Ruspini, é uma extensão do algoritmo *C-Means Rígido* (HCM), chamado de ISODATA proposto por Ball e Hall (1965). O HCM é um dos mais populares métodos de agrupamento. Ruspini (1969) introduziu a partição nebulosa para descrever estruturas de grupos de um conjunto de dados e sugeriu um algoritmo computacional que otimiza esta partição nebulosa. Dunn (1973) generalizou o procedimento de agrupamento de variância mínima para a técnica de agrupamento nebuloso do HCM. Bezdek (1981) generalizou a aproximação de Dunn criando assim o algoritmo *Fuzzy C-Means* (FCM).

Para descrever o algoritmo FCM é necessário definir alguns parâmetros e conceitos utilizados pelo algoritmo. Recordando que $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathfrak{R}^p$ é um conjunto finito de pontos, uma c -partição nebulosa para X é dado por.

$$M_{fcn} = \left\{ U \in [0,1]^{nc} \mid u_{ki} \in [0,1], \forall i, k; \sum_{i=1}^c u_{ki} = 1, \forall k; 0 < \sum_{k=1}^n u_{ki} < n, \forall i \right\}, \quad (2.19)$$

onde c é o número de grupos e já é conhecido *a priori*, com $2 \leq c \leq n$.

Os passos básicos do algoritmo FCM, fornecidos por Bezdek em 1981, são os seguintes:

1. Escolher o número de grupos c , $2 \leq c \leq n$, o parâmetro $m > 1$, o critério de parada $\varepsilon > 0$, o número máximo l_{\max} de iterações. Inicializar U^0 aleatoriamente e o contador $l=1$.
2. Calcular os c centros das classes \mathbf{v}_i^l $i=1, \dots, c$ com U^l e a fórmula:

$$\mathbf{v}_i^l = \frac{\sum_{k=1}^n [u_{ki}^{l-1}]^m \mathbf{x}_k}{\sum_{k=1}^n [u_{ki}^{l-1}]^m}, \quad i=1, \dots, c. \quad (2.20)$$

3. Atualizar a matriz de pertinência:

3.1 Para $1 \leq i \leq c$, $1 \leq k \leq n$,

3.1.1 Se $d_{ki} > 0$ então

$$u_{ki}^l = \left[\sum_{j=1}^c (d_{ki}/d_{kj})^{1/(m-1)} \right]^{-1} \quad (2.21)$$

3.1.2 Senão

Se $d_{ki} = 0$ para $i \in I \leq c$, então

Definir u_{ki}^l para $i \in I$ com um número real positivo que satisfaça a condição: $\sum u_{ki}^l = 1$ deste modo:

$$u_{ki}^l = 1 - \sum u_{ki}^l$$

Definir $u_{ki}^l = 0$ para $i \in c - I$.

4. Calcular $\Delta = \|U^l - U^{l-1}\| = \max_{ji} |u_{ji}^l - u_{ji}^{l-1}|$, $j=1, \dots, n$, $i=1, \dots, c$,

5. Se $\Delta > \varepsilon$ e $l < l_{\max}$, $l = l + 1$ e retornar para o passo (2); senão parar.

Algoritmo 2.1: *Fuzzy C-Means* – FCM.

O algoritmo FCM procura agrupar os pontos criando uma partição em X de modo que se minimize a seguinte função objetivo:

$$J_m(X, U, V) = \sum_{k=1}^n \sum_{i=1}^c u_{ki}^m d_{ki}^2, \quad U \in M_{fcn}, \quad 1 < m < \infty, \quad (2.22)$$

onde, $d_{ki} = \|\mathbf{x}_k - \mathbf{v}_i\|_A$, $\|\mathbf{x}\|_A = \mathbf{x}^T A \mathbf{x}$, $A_{p \times p}$ é a matriz identidade; $V = \{\mathbf{v}_1, \dots, \mathbf{v}_c\}$, $\mathbf{v}_i \in \mathfrak{R}^p$, $1 \leq i \leq c$; \mathbf{v}_i é o centro da i -ésima classe; u_{ki} denota o grau de pertinência de \mathbf{x}_k na classe i e m é um valor que modula o quão nebulosa é a partição obtida.

O algoritmo FCM adota a distância Euclidiana como medida de similaridade entre o ponto e o centro do grupo. O processo iterativo para a minimização da função objetivo (2.22) em geral conduz a um mínimo local.

2.5.2 Algoritmo Gustafson Kessel (GK)

Este algoritmo baseia-se no algoritmo FCM. O algoritmo FCM utiliza a distância Euclidiana como medida de similaridade. Como consequência, seu desempenho é favorecido quando os conjuntos de dados são separáveis ou quando os grupos têm aproximadamente os mesmos tamanhos e formas.

Contudo, se a forma dos grupos é conhecida, pode ser adequado determinar a matriz A para cálculo da medida de distância de $\|\mathbf{x}\|_A$. Utilizando-se desta matriz o algoritmo GK detecta classes elipsoidais de tamanhos e orientações diferentes. Isto é possível definindo-se a seguinte função objetivo (Gustafson and Kessel, 1979):

$$J_m(X, U, V) = \sum_{k=1}^n \sum_{i=1}^c u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i)^T M_i (\mathbf{x}_k - \mathbf{v}_i), U \in F_{fcn}, 1 < m < \infty \quad (2.23)$$

onde M_i é uma matriz simétrica, positiva, e é definida como matriz de covariância para o grupo i . Se $M_i = I_p$, a expressão (2.23) será igual à expressão (2.22), que utiliza a distância Euclidiana. A medida de similaridade abaixo, adotada pelo algoritmo GK, é a distância de Mahalanobis, conforme Seção 2.2.2:

$$d_{ki} = (\mathbf{x}_k - \mathbf{v}_i)^T M_i (\mathbf{x}_k - \mathbf{v}_i), i = 1, \dots, c, \quad k = 1, 2, \dots, n.$$

onde a matriz M_i é determinada como: $M_i = [(\det(F_i))^{1/n+1} F_i^{-1}]$, sendo F_i^{-1} calculada utilizando (2.24).

Os passos do algoritmo GK são descritos a seguir:

1. Escolher o número de grupos c , $2 \leq c \leq n$, o parâmetro $m > 1$, o critério de parada $\varepsilon > 0$, o número máximo l_{\max} de iterações. Inicializar U^0 aleatoriamente e o contador $l=1$.
2. Calcular os c centros das classes \mathbf{v}_i^l $i=1, \dots, c$ com U^l e a equação (2.20):
3. Calcular a matriz de covariância:

$$F_i = \frac{\sum_{k=1}^n [u_{ki}^{l-1}]^m (\mathbf{x}_k - \mathbf{v}_i^l)(\mathbf{x}_k - \mathbf{v}_i^l)^T}{\sum_{k=1}^n [u_{ki}^{l-1}]^m}, \forall i, i=1,2,\dots,c. \quad (2.24)$$

4. Calcular a distância de Mahalanobis:

$$d_{ki} = (\mathbf{x}_k - \mathbf{v}_i^l)^T [(\det(F_i))^{1/n+1} F_i^{-1}] (\mathbf{x}_k - \mathbf{v}_i^l), \quad (2.25)$$

$i=1,2,\dots,c, \quad k=1,2,\dots,n.$

5. Atualizar a matriz de pertinência:

5.1 Para $1 \leq i \leq c$, $1 \leq k \leq n$,

5.1.1 Se $d_{ki} > 0$ então

Atualizar u_{ki}^l via (2.21)

5.1.2 Senão

Se $d_{ki} = 0$ para $i \in I \leq c$, então

Definir u_{ki}^l para $i \in I$ com um número real positivo que satisfaça a condição: $\sum u_{ki}^l = 1$ deste modo:

$$u_{ki}^l = 1 - \sum u_{ki}^l$$

Definir $u_{ki}^l = 0$ para $i \in c - I$.

6. Calcular $\Delta = \|U^l - U^{l-1}\| = \max_{ji} |u_{ji}^l - u_{ji}^{l-1}|$, $j=1,\dots,n$, $i=1,\dots,c$,

7. Se $\Delta > \varepsilon$ e $l < l_{\max}$, $l = l + 1$ e retornar para o passo (2); senão parar.

Algoritmo 2.2: Gustafson Kessel - GK

2.5.3 Algoritmo com Aprendizagem Competitiva Fuzzy (FCL)

Diferente do algoritmo FCM que atualiza os centros por batelada, o algoritmo FCL (Backer and Scheunders, 1999a) atualiza seus centros ponto-a-ponto, isto é, para cada iteração, um ponto é apresentado e associado ao seu grupo por (2.21), e um grupo, representado pelo seu centro, atualizado por:

$$\mathbf{v}_i^{l+1} = \mathbf{v}_i^l + \alpha^l (\mathbf{x}_k - \mathbf{v}_i^l), \quad (2.26)$$

onde α^l é a taxa de aprendizagem, e assume uma função decrescente do número de iterações. Este procedimento é conhecido como *Aprendizagem Competitiva* (CL)

(Kohonen, 1995). Para se obter o algoritmo com aprendizagem competitiva nebulosa é necessário multiplicar a medida de similaridade pelo grau de pertinência do ponto k com o centro i . Obtendo uma extensão de (2.26), mostrada abaixo:

$$\mathbf{v}_i^{l+1} = \mathbf{v}_i^l + \alpha^l u_{ki}^m (\mathbf{x}_k - \mathbf{v}_i^l). \quad (2.27)$$

com esta regra, obtém-se o algoritmo de aprendizagem competitiva nebulosa.

Resumidamente os passos do algoritmo FCL são os seguintes:

1. Escolher o número de grupos c , $2 \leq c \leq n$, o parâmetro $m > 1$, o critério de parada $\varepsilon > 0$, o número máximo l_{\max} de iterações. Inicializar U^0 aleatoriamente e o contador $l=0$. Calcular os c centros das classes \mathbf{v}_i^0 $i=1, \dots, c$ com U^0 .
2. Para $1 \leq i \leq c$, $1 \leq k \leq n$,
 - 2.1 Escolher aleatoriamente um ponto de X ;
 - 2.2 Atualizar a matriz de pertinência com a Eq. (2.21);
 - 2.3 Atualizar a posição do centro i usando a Eq. (2.27);
3. Calcular $\Delta = \|U^{l+1} - U^l\| = \max_{j,i} |u_{ji}^{l+1} - u_{ji}^l|$, $j=1, \dots, n$, $i=1, \dots, c$,
4. Se $\Delta > \varepsilon$ e $l < l_{\max}$, $l = l + 1$ e retornar para o passo (2); senão parar.

Algoritmo 2.4: Aprendizagem Competitiva Nebulosa (FCL).

2.5.4 Algoritmo com Aprendizagem Competitiva Elíptica Nebulosa Sensível à Frequência (FFSECL)

O FFSECL é um algoritmo com aprendizagem competitiva para agrupamento elipsoidal. Este algoritmo, além do método de aprendizagem competitiva, introduz a atualização direta da inversa da matriz de covariância. Com este mecanismo deve-se tornar o algoritmo FFSECL mais eficiente do que o algoritmo GK (Backer and Scheunders, 1999b).

Backer e Scheunders (1999b) utilizaram a seguinte expressão para o cálculo da matriz de covariância :

$$[F_i^{l+1}]^{-1} = \frac{[F_i^l]^{-1}}{1-\alpha} - \frac{\alpha}{1-\alpha} \cdot \frac{[[F_i^l]^{-1}(\mathbf{x}_k - \mathbf{v}_i)] \cdot ([F_i^l]^{-1}(\mathbf{x}_k - \mathbf{v}_i))^T}{1 + \alpha [(\mathbf{x}_k - \mathbf{v}_i)^T [F_i^l]^{-1} (\mathbf{x}_k - \mathbf{v}_i) - 1]}. \quad (2.28)$$

Deste modo, é possível identificar grupos com formas elípticas sem a necessidade de se calcular sempre a inversa da matriz de covariância.

O mecanismo de atualização nebulosa dos centros permanece o mesmo do algoritmo FCL, isto é:

$$\mathbf{v}_i^{l+1} = \mathbf{v}_i^l + \alpha^l u_{ki}^m (\mathbf{x}_k - \mathbf{v}_i^l).$$

A matriz de covariância calculada em (2.28) é adaptada para incluir a matriz de pertinência, onde, novamente, a taxa de aprendizagem irá ser multiplicada pela matriz de pertinência. Assim, a expressão obtida é:

$$[F_i^{l+1}]^{-1} = \frac{[F_i^l]^{-1}}{1 - \alpha u_{ki}^m} - \frac{\alpha u_{ki}^m}{1 - \alpha u_{ki}^m} \cdot \frac{[F_i^l]^{-1}(\mathbf{x}_k - \mathbf{v}_i) \cdot ([F_i^l]^{-1}(\mathbf{x}_k - \mathbf{v}_i))^T}{1 + \alpha u_{ki}^m [(\mathbf{x}_k - \mathbf{v}_i)^T [F_i^l]^{-1} (\mathbf{x}_k - \mathbf{v}_i) - 1]} \quad (2.29)$$

O algoritmo é inicializado considerando os grupos com formas esféricas. Contudo, durante o treinamento as formas dos grupos poderão ser adaptadas iterativamente para se tornarem elipsoidais.

Quando se usa a aprendizagem competitiva, poderão ocorrer instabilidades durante as primeiras iterações para favorecer uma classe maior. Este efeito, conhecido como área do vetor de quantização, pode ser removido usando um fator de aproximação sensível a frequência, isto é, a distância calculada é multiplicada pelo número de vezes que a classe k é modificada na l -ésima iteração, denotado por n_{kl} . Deste modo, as classes com distâncias muito irregulares e com poucos pontos também serão atualizadas (Backer and Scheunders, 1999b). A aproximação sensível à frequência também pode ser aplicada ao algoritmo FCL. Neste caso, a distância d_{ik} é multiplicada por:

$$\mu_i^{l+1} = \mu_i^l + u_{ik}^m, \quad (2.30)$$

que é o total dos valores de pertinência da classe i durante toda a iteração para o ponto k . Apesar do agrupamento nebuloso já cuidar destas classes distantes, (2.30) tem se mostrado mais eficiente (Backer and Scheunders, 1999b).

Os passos do algoritmo apresentado por (Backer, 2002) para o FFSECL são os seguintes:

1. Escolher o número de grupos c , $2 \leq c \leq n$, o parâmetro $m > 1$, o critério de parada $\varepsilon > 0$, o número máximo l_{\max} de iterações. Inicializar U^0 aleatoriamente e o contador $l=0$. Calcular os c centros das classes v_i^0 $i=1, \dots, c$ com U^0 . Igualar todas as matrizes de covariância a matriz de identidade.
2. Para $1 \leq i \leq c$, $1 \leq k \leq n$,
 - 2.1 Calcular a distância de Mahalanobis (2.25);
 - 2.2 Multiplicar estas distâncias pelo termo de frequência μ_i^l (2.30).
 - 2.3 Escolher aleatoriamente um ponto de X ;
 - 2.4 Atualizar a matriz de pertinência com a Eq. (2.21);
 - 2.5 Atualizar a posição do centro i usando a Eq. (2.27);
 - 2.6 Atualizar a inversa da matriz de covariância aplicando (2.29);
 - 2.7 Atualizar o termo frequência usando (2.30);
3. Calcular $\Delta = \|U^{l+1} - U^l\| = \max_{ji} |u_{ji}^{l+1} - u_{ji}^l|$, $j = 1, \dots, n$, $i = 1, \dots, c$,
4. Se $\Delta > \varepsilon$ e $l < l_{\max}$, $l = l + 1$ e retornar para o passo (2); senão parar.

Algoritmo 2.5: Aprendizagem Competitiva Elíptica Nebulosa Sensível a Frequência (FFSECL)

2.5.5 Algoritmo Fuzzy K-Means Modificado (MFKM)

Este método foi desenvolvido para agrupamentos de classes não convexas. A aproximação é baseada em uma medida induzida pelos dados (DIM). A distância é calculada usando técnicas da teoria de grafos, determinando o menor caminho em um grafo (Gath *et al.*, 1997).

Exemplos de classes não convexas de formas arbitrárias podem ser encontradas em imagens médicas, padrões de escritas à mão e análise de imagens faciais. Os algoritmos mencionados nas seções anteriores não são os mais apropriados para identificar diretamente estas classes de formas arbitrárias, pois as geometrias destas classes não podem ser facilmente descritas analiticamente (Gath *et al.*, 1997). Estes algoritmos são particularmente eficientes para a detecção e caracterização em dados multidimensionais de classes convexas, separáveis, ou sobrepostas, hiperelipsoidais de classes compactas (Bezdek, 1981; Gustafson and Kessel, 1979; Gath and Geva, 1989).

O algoritmo MFKM determina grupos não convexas, além de grupos de formas esféricas e elípticas, baseando-se na incorporação de parâmetros geométricos das classes envolvidas no esquema de otimização (Bezdek, 1981; Dave, 1996; Gustafson and Kessel, 1979; Kaymak and Setnes, 2000; Krishnapuram and Frigui, 1992). Contudo, uma

condição necessária para a identificação de classes não convexas é que estas classes não estejam sobrepostas.

O algoritmo MFKM, ao invés de pré-determinar uma métrica em uma função objetivo a ser otimizada, encontra uma métrica para um determinado conjunto de dados, refletindo a estrutura específica do conjunto de dados. O principal conceito é o de métrica induzida por dados (DIM - *data induced metric*), o que permite estender o escopo dos algoritmos de agrupamentos nebulosos para classes não convexas.

A medida induzida por dados considera regiões de classes com alta densidade de padrões, separadas por regiões com baixa densidade de padrões, podendo ser comparada à propagação de luz na ótica. Os padrões que estão dentro de uma classe são conectados por caminhos que atravessam somente regiões densas, ao passo que não há caminhos semelhantes entre padrões para dois grupos diferentes. Isto porque a distância, definida de acordo com o conceito DIM, será menor para os padrões que pertencerem à mesma classe. Esta distância satisfaz os axiomas da métrica citados na Seção 2.2.1. Para obter uma distância de acordo com DIM, uma teoria de aproximação por grafos faz-se necessária (Gath *et al.*, 1997). Esta teoria é discutida a seguir.

O algoritmo MFKM é resumido através dos seguintes passos (Gath *et al.*, 1997):

1. Fase de Inicialização;
 - 1.1 Escolher o número de grupos c , $2 \leq c \leq n$, o parâmetro $m > 1$, o critério de parada $\varepsilon > 0$, o número máximo l_{\max} de iterações. Inicializar U^0 aleatoriamente e o contador $l=1$. Escolher os valores padrões para η e a .
 - 1.2 Preparar a grade e gerar os pesos dos grafos;
 - 1.3 Determinar o centro inicial da classe, v_1^0 ,
2. Para cada iteração $l= 1, 2, \dots$
 - 2.1 Para $1 \leq i \leq c$, $1 \leq k \leq n$,
 - 2.1.1 Calcular U^l utilizando (2.21), onde a distância é definida de acordo com os princípios DIM.
 - 2.2 Atualizar v_i^{l*} , utilizando (2.20).
3. Calcular $\Delta = \|U^l - U^{l-1}\| = \max_{ji} |u_{ji}^l - u_{ji}^{l-1}|$, $j = 1, \dots, n$, $i = 1, \dots, c$,
4. Se $\Delta > \varepsilon$ e $l < l_{\max}$, $l = l + 1$ e retornar para o passo (2); senão parar.

Algoritmo 2.6: Fuzzy K-Means Modificado (MFKM)

* Para que o número de grupos seja encontrado automaticamente, uma alteração nos cálculos da atualização dos centros pode ser feita da seguinte forma:

Passo 2.2. Atualizar ν_i^l , utilizando (2.21). Ao atualizar os centros ν verifica-se se existe algum ponto que não pertence a nenhum grupo já existente. Caso exista, este ponto é considerado um centro de novo grupo. Isto é possível pois cada grupo existente corresponde uma árvore, cujos pontos estarão distribuídos nos nós da mesma. Se o ponto não estiver em nenhuma destas árvores, este ponto é considerado como um novo grupo.

O procedimento para determinar a distância consiste em três fases: uma fase de preparação, uma fase de geração dos pesos de um grafo (executado somente no início do algoritmo) e o cálculo das distâncias entre padrões.

Para a fase de geração dos pesos dos grafos, um hiper-retângulo, contendo todos os padrões, é considerado como um domínio de trabalho. A resolução da métrica é determinada pelo tamanho das células das grades, isto é, pelo número de discretização do retângulo. Por exemplo, no caso de duas dimensões, o número de células da grade é definido empiricamente por:

$$e = g = \text{round}(\eta\sqrt{n}), \quad (2.31)$$

onde n é o número de pontos de X , e e g são os números de células nas direções x e y , respectivamente, e η determina a resolução da separação de um dado dentro de grupos.

Definido o número de células da grade, calculam-se as dimensões q e r das células da seguinte forma:

$$\begin{aligned} q &= \frac{x_{\max} + \varepsilon - (x_{\min} - \varepsilon)}{e}, \\ r &= \frac{y_{\max} + \varepsilon - (y_{\min} - \varepsilon)}{g}, \end{aligned} \quad (2.32)$$

onde $x_{\max} = \max_{j=1}^n(x_j)$, $x_{\min} = \min_{j=1}^n(x_j)$, $y_{\max} = \max_{j=1}^n(y_j)$, $y_{\min} = \min_{j=1}^n(y_j)$. Este exemplo do algoritmo refere-se somente para dados em duas dimensões. Naturalmente o algoritmo pode ser estendido para p -dimensões. Neste caso, as dimensões da grade serão

determinadas por equações semelhantes a (2.32), e cada vértice interno será conectado por (3^p-1) vizinhos.

Observe o exemplo da Figura 2.8 abaixo, onde $n=75$ e $\eta=0.3$. Aplicando estes dados em 2.31, obtêm-se o número de células $e=g=3$. Neste exemplo, $x_{\max}=y_{\max}=1$ e $x_{\min}=y_{\min}=0$, $\varepsilon=0.001$. As dimensões obtidas das células com a equação (2.32) são $q=r=0.334$.

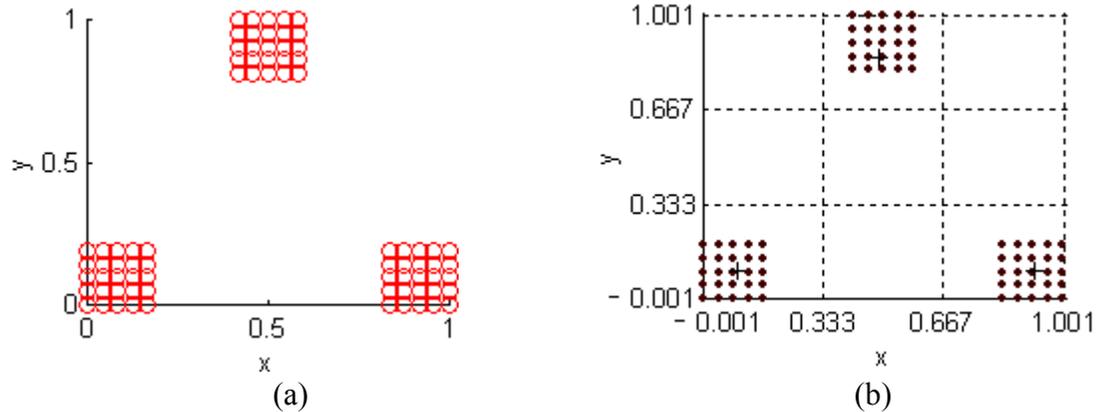


Figura 2.8: (a) Conjunto de dados original e (b) a grade gerada com 3×3 de células.

A fase da geração dos grafos consiste na geração de um espaço discretizado usando a grade. Isto é realizado porque cada célula da grade contém um determinado número de pontos. Aproximadamente n células das grades são estimadas como vértices de um grafo, e pesos são associados aos vértices e não às arestas. Pois, os pesos de um vértice, fazendo uma analogia, representam o tempo necessário de um "raio de luz" passar por este vértice, ou seja, o peso é inversamente proporcional a densidade. Uma forma de se determinar os pesos é utilizar a expressão:

$$w_{ij} = (n_{ij} + a)^{-1}, \quad (2.33)$$

onde n_{ij} é o número de pontos contidos em uma célula (i, j) , e a é um parâmetro que determina se o ponto pertence ou não à célula. Sendo assim, o caminho percorrido na árvore dependerá do número de nós (vértices) em um único caminho. O parâmetro a é escolhido empiricamente no intervalo entre $[0 : 0,25]$.

Na terceira fase, o menor caminho de um determinado nó para todos os outros nós são calculados pelo algoritmo de *Dijkstra* (Cormem *et al.*, 1991). Os pesos são

considerados como as respectivas distâncias. Somente os caminhos conectados ao nó "centro" (uma célula que contém v_i) para os nós não vazios, isto é, vértices com $n_{ij} \neq 0$, deverão ser determinados. A distância entre dois pontos é igual a distância entre dois nós que contiverem estes pontos.

Como é possível observar na literatura, inúmeros algoritmos tem sido desenvolvidos para o agrupamento nebuloso de dados (Al-Sultan, 1996; Davé, 1993; Frigui and Krishnapuram, 1996; Geva *et al.*, 2000; Guha *et al.*, 1998; Guha *et al.*, 1999; Karypis *et al.*, 1999; Halkidi *et al.*, 2001; Yager and Filev, 1994; Guoyao, 1998). De modo geral, os mais utilizados são: o algoritmo FCM e sua família (Backer, 2002; Geva, 2000; Frigui and Krishnapuram, 1996; Kaymak and Setnes, 2000; Gustafson and Kessel, 1979; Halkidi *et al.*, 2001), devido a sua eficácia, simplicidade e eficiência computacional. Contudo, a desvantagem de se requisitar o número de grupos *a priori* tornando-os inviáveis para muitas outras aplicações onde se desconhece o número de grupos (Zahid *et al.*, 1999).

A Tabela 2.3 resume as principais características de cada algoritmo aqui apresentado, classificando estes algoritmos de acordo com as categorias discutidas na Seção 2.3.

Tabela 2.3: Principais características dos algoritmos de agrupamento.

Nome	Categoria	$O()$ *	Geometria	Ruído	Parâmetros	Resultados	Critério de agrupamento
FCM	Agrupamento por Partição, AO.	$O(nc^2p)$	Formas convexas	Não	Número de grupos	Centros dos grupos e matriz de pertinência.	$\min_{U, v_1, v_2, \dots, v_c} (J_m(U, V))$ $J_m(U, V) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m d^2(\mathbf{x}_k, \mathbf{v}_i)$
GK	Agrupamento por Partição, AO.	$O(n^2c^2p^3)$	Formas não convexas	Não	Número de grupos	Centros dos grupos e matriz de pertinência.	$\min_{U, v_1, v_2, \dots, v_c} (J_m(U, V))$ $J_m(U, V) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i) M_i (\mathbf{x}_k - \mathbf{v}_i)$
FCL	Agrupamento por Partição, ACE.	$O(nc^2p)$	Formas convexas	Não	Número de grupos	Centros dos grupos e matriz de pertinência.	$\min_{U, v_1, v_2, \dots, v_c} (J_m(U, V))$ $J_m(U, V) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m d^2(\mathbf{x}_k, \mathbf{v}_i)$ $\mathbf{v}_i^{l+1} = \mathbf{v}_i^l + \alpha^l u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i^l)$
FFSECL	Agrupamento por Partição, ACE.	$O(nc^3p)$	Formas não convexas	Não	Número de grupos	Centros dos grupos e matriz de pertinência.	$\min_{U, v_1, v_2, \dots, v_c} (J_m(U, V))$ $J_m(U, V) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i) M_i (\mathbf{x}_k - \mathbf{v}_i)$ $\mathbf{v}_i^{l+1} = \mathbf{v}_i^l + \alpha^l u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i^l)$
MFKM	Combina agrupamento por partição, hierárquico, baseado em densidade e por grade, AO.	$O(n^3c^2(3^p))$	Formas arbitrárias	Não	Coefficiente de grade	Centros dos grupos, número de grupos e matriz de pertinência.	$\min_{U, v_1, v_2, \dots, v_c} (J_m(U, V))$ utiliza a densidade para calcular os pesos da árvore e utiliza o algoritmo de <i>Dijkstra</i> para encontrar as classes.

* $O()$ é a complexidade computacional do algoritmo, n é o número de pontos, c é o número de grupos e p é o número de atributos.

2.6 Métodos de Validação de Grupos

O resultado do algoritmo de agrupamento é uma partição do conjunto de dados. Esta partição pode ser ou rígida ou nebulosa. Para obter esta partição muitas vezes presume-se a existência de c grupos. Surge, portanto, a necessidade de formular critérios ou métodos quantitativos que ajudem a realizar uma avaliação objetiva dos resultados dos algoritmos de agrupamento. Estes critérios e métodos quantitativos são índices que auxiliam na validação dos agrupamentos realizados pelos algoritmos. O ideal é ter um algoritmo de agrupamento que seja capaz de incorporar um índice de validação. Contudo, este ainda não é o caso da grande maioria dos algoritmos de agrupamento.

Quando um algoritmo não considera um critério de validação, utiliza-se de métodos para determinar o número de grupos e a partição mais adequada dentre os resultados encontrados. Estes métodos utilizam funções de validação aplicadas sobre a partição. As funções de validação têm sido usadas como indicadores da qualidade do resultado da partição pois fornecem índices de validação. Estes índices podem ser considerados como um valor que disponibiliza uma maneira de avaliar os resultados do agrupamento.

Os critérios solicitados pelo índice de validação para definir uma partição aceitável baseiam-se nos três requisitos citados abaixo (Gath and Geva, 1989):

1. Clara separação entre grupos resultantes;
2. Uma determinada concentração de pontos (coesão) em torno do centro de um grupo;
3. Menor número de grupos possível, desde que também obedecendo aos requisitos anteriores.

A seguir os mais importantes índices de validação para partições nebulosas são revistos:

1. Índice de particionamento entrópico (*Partitioning entropy*- PE) (Lin and Lee, 1995; Roubens, 1982):

$$PE(U, c) = -\frac{1}{n} \sum_{k=1}^n \sum_{i=1}^c u_{ki} \log_a u_{ki}, \quad 1 < a < \infty, \quad (2.34)$$

onde a é uma base logarítmica. O índice PE possui valores no intervalo $[0, \log_a c]$, e possui uma função minimizadora, isto é, o valor de PE tende para 0 quando se têm grupos bem definidos em X . Um valor próximo do limite superior do intervalo ($\log_a c$), indica a ausência de estruturas de grupos no conjunto de dados ou a incapacidade do algoritmo de obtê-las.

2. Índice ou coeficiente de particionamento (*Partitioning coefficient* - PC) (Lin and Lee, 1995; Roubens, 1982):

$$PC(U, c) = \frac{1}{n} \sum_{k=1}^n \sum_{i=1}^c u_{ki}^2. \quad (2.35)$$

O valor de PC está no intervalo $[1/c, 1]$, onde c é o número de classes. Quando $u_{ki}=1/c$, o índice atinge o seu valor mais baixo. Um valor do índice próximo de $1/c$ indica a ausência de grupos no conjunto X considerado.

Os índices PC e PE têm alguns inconvenientes (Dave, 1996) tais como:

- i) Dependem do número de classes. O valor para PC é inversamente proporcional ao número de classes e para o PE, o valor é diretamente proporcional;
- ii) São sensíveis ao parâmetro m no caso da família FCM;
- iii) Não possuem conexão direta com os dados, visto que estes índices não utilizam o conjunto X .

3. Índice de hipervolume nebuloso (HPV) (Gath and Geva, 1989; Geva, 1999)

$$HPV = \sum_{i=1}^c h_i, \quad (2.36)$$

onde o hipervolume da i -ésima classe é determinada por $h_i = [\det(F_i)]^{1/2}$ sendo F , a matriz de covariância relacionada ao centro i , isto é:

$$F_i = \frac{\sum_{k=1}^n [u_{ki}^{l-1}]^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T}{\sum_{k=1}^n [u_{ki}^{l-1}]^m}, \quad i = 1, 2, \dots, c. \quad (2.37)$$

4. Índice de partição por densidade média (*Average partition density*)

A densidade média é calculada de acordo com a seguinte fórmula:

$$AD = \frac{1}{c} \sum_{i=1}^c \frac{S_i}{[\det(F_i)]^{1/2}}, \quad (2.38)$$

onde S_i é conhecida como a "soma dos membros dos centros", calculada como:

$$S_i = \sum_{k=1}^n u_{ki}, \quad \forall \mathbf{x}_k \in \{\mathbf{x}_k \mid (\mathbf{x}_k - \mathbf{v}_i)^T F_i^{-1} (\mathbf{x}_k - \mathbf{v}_i) < 1\}. \quad (2.39)$$

5. Índice de partição por densidade (*Partition density*) (Dave, 1996; Gath and Geva, 1989; Geva, 1999):

A partição por densidade é calculada pela divisão do número de pontos nas classes pelo hipervolume nebuloso. A massa de classes é obtida pela soma dos membros de cada classe.

$$PD = \frac{S}{HPV}, \quad (2.40)$$

onde,

$$S = \sum_{k=1}^n \sum_{i=1}^c u_{ki}, \quad \forall \mathbf{x}_k \in \{(\mathbf{x}_k - \mathbf{v}_i)^T F_i^{-1} (\mathbf{x}_k - \mathbf{v}_i) < 1\} \quad (2.41)$$

6. Índice de Xie - Beni (Xie and Beni, 1991):

A função de validação XB pretende identificar a compactação e a separação das classes de modo a encontrar um número ótimo de classes. Então a compactação de uma partição nebulosa é definida por (2.42).

$$Comp = \frac{1}{n} \sum_{i=1}^c \sum_{k=1}^n u_{ki}^m \|\mathbf{v}_i - \mathbf{x}_k\|^2. \quad (2.42)$$

Se os conjuntos de dados possuírem um baixo índice de compactação, então todas as classes são compactas e separadas umas das outras. A separação ou a medida de distância entre os centros das classes de uma partição nebulosa é:

$$Sep = \min_{i \neq k} \|\mathbf{v}_i - \mathbf{x}_k\|^2. \quad (2.43)$$

A função de validação XB é definida como a razão entre a compactação e a separação da partição nebulosa, $XB = Comp/Sep$. Logo, um pequeno valor para XB reflete um pequeno valor de compactação, pois u_{ki} é suposto ser um valor alto quando $\|\mathbf{v}_i - \mathbf{x}_k\|$ é pequeno. O valor ótimo de c é obtido pela minimização XB sobre $c \in \{2, 3, \dots, c_{\max}\}$. Assume-se que ao menor valor de XB corresponde, o número aceitável de classes.

7. Índice de Fukuyama-Sugeno (Fukuyama and Sugeno, 1989; Halkidi *et al.*, 2001)

O índice *Fukuyama-Sugeno* é definido como:

$$FS_m = \sum_{k=1}^n \sum_{i=1}^c u_{ki}^m \left(\|\mathbf{x}_k - \mathbf{v}_i\|_A^2 - \|\mathbf{v}_i - \mathbf{v}\|_A^2 \right), \quad (2.44)$$

onde \mathbf{v} é o centro do vetor de X e A é uma matriz simétrica definida positiva. Quando $A = I$, a distância será o quadrado da distância Euclidiana. Para compactar e separar bem as classes espera-se valores pequenos para FS_m . O primeiro termo entre parênteses refere-se à compactação das classes, enquanto que o segundo refere-se à distância entre as classes.

Outros métodos de validação de classes são estudados em (Bezdek and Pal, 1998; Dave, 1996; Geva *et al.*, 2000; Geva, 1999; Halkidi *et al.*, 2001; Sintas *et al.*, 2000; Xie and Beni, 1991).

A Tabela 2.4 resume as funções de validação.

Tabela 2.4: Funções de validação.

Índice	Função de validação	F(.)	Autor
Partição entrópica	$PE(U, c) = -\frac{1}{n} \sum_{k=1}^n \sum_{i=1}^c u_{ki} \log_a u_{ki}, 1 < a < \infty$	↓	Bezdek, (1974)
Coefficiente de particionamento	$PC(U, c) = \frac{1}{n} \sum_{k=1}^n \sum_{i=1}^c u_{ki}^2$	↑	Dunn, (1976)
Hipervolume nebuloso	$HPV = \sum_{i=1}^c [\det(F_i)]^{1/2}$ $F_i = \frac{\sum_{k=1}^n [u_{ki}^{l-1}]^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T}{\sum_{k=1}^n [u_{ki}^{l-1}]^m}, \forall i, i=1, 2, \dots, c.$	↓	Gath e Geva, (1989)
Partição por densidade média	$AD(k) = \frac{1}{K} \sum_{i=1}^K \frac{S_i}{[\det(F_i)]^{1/2}}$ $S_i = \sum_{k=1}^n u_{ki}, \forall \mathbf{x}_k \in \{\mathbf{x}_k \mid (\mathbf{x}_k - \mathbf{v}_i)^T F_i^{-1} (\mathbf{x}_k - \mathbf{v}_i) < 1\}$	↑	Gath e Geva, (1989)
Partição por densidade	$PD = \frac{S}{HPV}$ $S = \sum_{k=1}^n \sum_{i=1}^c u_{ki}, \forall \mathbf{x}_k \in \{(\mathbf{x}_k - \mathbf{v}_i)^T F_i^{-1} (\mathbf{x}_k - \mathbf{v}_i) < 1\}$	↑	Gath e Geva, (1989)
Xie-Beni	$XB = \frac{\sum_{i=1}^c \sum_{k=1}^n \mu_{ki}^m \ \mathbf{v}_i - \mathbf{x}_k\ ^2}{n \left(\min_{i \neq k} \ \mathbf{v}_i - \mathbf{x}_k\ ^2 \right)}$	↓	Xie e Beni, (1991)
Fukuyama-Sugeno	$FS_m = \sum_{k=1}^n \sum_{i=1}^c u_{ki}^m \left(\ \mathbf{x}_k - \mathbf{v}_i\ _A^2 - \ \mathbf{v}_i - \mathbf{v}\ _A^2 \right)$	↓	Fukuyama e Sugeno, (1989)

Legenda:

F(.) = ↓ : minimiza função;

F(.) = ↑ : maximiza função.

2.7 Resumo

Neste capítulo foram apresentados os principais conceitos sobre agrupamento nebuloso de dados e os algoritmos de agrupamento que se mostram mais promissores na literatura e as suas principais características. Além disso, os índices de validação de resultados de agrupamento mais relevantes foram resumidos. Contudo, deve-se enfatizar que, apesar dos avanços e das diferentes propostas na literatura visando uma validação de grupos objetiva e bem determinada, presentemente nenhuma delas é aceita com unanimidade.

O próximo capítulo abordará o conceito de aprendizagem participativa.

3. Aprendizagem Participativa

3.1 Introdução

Discute-se aqui o conceito de aprendizagem participativa e descreve-se suas principais características. O conceito de aprendizagem participativa foi introduzido por Yager em 1990, como um mecanismo que permite representar o aprendizado sobre um sistema com a participação das crenças do próprio sistema. O modelo de aprendizagem participativa (*participatory learning*), conforme ilustra a Figura 3.1, tem como conceito fundamental o fato de que a aprendizagem e a revisão de crenças sobre um sistema dependem do que já se conhece e o que se acredita sobre o próprio sistema. Isto é, o conhecimento atual sobre o que se quer aprender participa do processo de aprendizagem e influencia a receptividade de novas observações durante o aprendizado.

Um item essencial na aprendizagem participativa é a compatibilidade entre novas observações e o que o sistema já possui como conhecimento. Ou seja, a compatibilidade entre as observações e as crenças existentes.

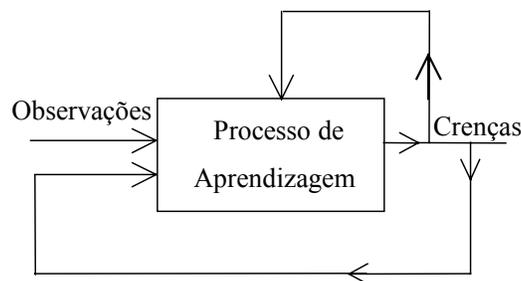


Figura 3.1: Estrutura da aprendizagem participativa.

É importante notar na Figura 3.1 que as crenças atuais do sistema provêm ou do processo de aprendizagem, ou da revisão de suas próprias crenças, ou de ambas, conforme mostra o elo inferior de realimentação. O processo de aprendizagem também é afetado pelas próprias crenças e interfere no mecanismo de processamento de novas informações, conforme sugere o elo de realimentação superior. Este tipo de realimentação é usado para indicar que as crenças e teorias atuais são resultados do

aprendizado proveniente do processamento de novas observações, com o auxílio destas mesmas crenças e teorias. Portanto, o elo superior corresponde ao elemento participativo da aprendizagem.

3.2 Fundamentos da Aprendizagem Participativa

Seja $\mathbf{v} \in [0,1]^p$, uma variável que contém as crenças do sistema. O objetivo é estimar os valores desta variável. Para tal, vamos supor que informações sobre estes valores são obtidas via uma seqüência de observações $\mathbf{x}_k \in [0,1]^p$, onde \mathbf{x}_k é uma manifestação dos valores de \mathbf{v} na k -ésima observação. Portanto, utilizamos os vetores \mathbf{x} em um mecanismo de aprender os valores de \mathbf{v} . Dizemos que o processo de aprendizagem é participativo se a contribuição da observação \mathbf{x}_k ao processo de aprendizagem depende de sua aceitação ou não, através das estimativas atuais dos valores de \mathbf{v} . Implícita nesta caracterização é a idéia de que, para ser útil e assim contribuir na aprendizagem de \mathbf{v} , a observação \mathbf{x}_k deve, de alguma forma, ser compatível com as estimativas de \mathbf{v} já disponíveis. Seja \mathbf{v}_k a estimativa de \mathbf{v} após k observações. Então, a idéia de aprendizagem participativa significa que, para que \mathbf{x}_k seja relevante na estimativa dos valores de \mathbf{v}_k , \mathbf{x}_k deve ser próximo de \mathbf{v}_k . Intuitivamente, a aprendizagem participativa mostra a tendência em se aprender aquilo que não é muito diferente de si próprio. Uma forma de se atualizar a estimativa dos valores, ou crenças, de \mathbf{v} é através de um algoritmo do tipo

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \alpha \rho_k (\mathbf{x}_k - \mathbf{v}_k), \quad (3.1)$$

onde $k=1, \dots, n$, e n é o número de observações; \mathbf{v}_{k+1} representa as novas crenças do sistema; $\mathbf{v}_k \in [0,1]^p$ são as crenças atuais; $\mathbf{x}_k \in [0,1]^p$ é a observação atual; $\alpha \in [0,1]$ é a taxa de aprendizagem; $\rho_k \in [0,1]$ é o grau de compatibilidade atual entre \mathbf{x}_k e \mathbf{v}_k , determinado por:

$$\rho_k = F(S_{k1}, S_{k2}, \dots, S_{kp}),$$

onde S_{kj} é uma medida de similaridade,

$$S_{kj} = G_{kj}(\mathbf{v}_{kj}, \mathbf{x}_{kj}),$$

e $S_{kj} \in [0,1]$, $j=1, \dots, p$, e F é um operador de agregação (Pedrycz and Gomide, 1998), $S_{kj}=1$ indica similaridade plena, enquanto $S_{kj}=0$ significa nenhuma similaridade. É

importante notar que G_{kj} mapeia pares (v_{kj}, x_{kj}) em um grau de similaridade, o que significa que os valores de v_{kj} e x_{kj} não precisam, necessariamente, estarem no p -cubo unitário. Além disso, a função G_{kj} permite que dois valores de v_k e x_k tenham $S_{kj}=1$ mesmo que os valores não sejam iguais. Note também que esta formulação, ao contrário da usual, permite o uso de diferentes medidas de similaridade para diferentes componentes dos vetores, isto é, diferentes j 's na notação acima. Uma possível formulação é a seguinte:

$$\rho_k = 1 - \frac{1}{p} \sum_{j=1}^p d_{kj}, \quad (3.2)$$

onde $d_{kj} = |x_{kj} - v_{kj}|$. Claramente, ρ_k fornece uma medida de compatibilidade entre as observações x_{kj} com as crenças atuais v_{kj} . Note que, neste caso, ρ_k é o complemento da média dos valores absolutos da diferença entre x_{kj} e v_{kj} .

Observe que a expressão (3.2) é um caso especial da distância de Hamming descrita na Seção 2.2.2. Em alguns domínios, principalmente em aprendizagem de máquina, é comum escolher a norma Euclidiana como alternativa. Em geral, a medida de compatibilidade ρ_k pode ser descrita como:

$$\rho_k = 1 - d_k, \quad (3.3)$$

onde d_k é definido como $d_k = \|\mathbf{x}_k - \mathbf{v}_k\|$ e $\|\cdot\|$ denota uma norma.

Analisando a expressão (3.2), são observados dois casos extremos. No primeiro, onde $\rho_k = 0$, a observação \mathbf{x}_k é totalmente contraditória ou incompatível com as crenças \mathbf{v}_k do sistema. Conseqüentemente:

$$\mathbf{v}_{k+1} = \mathbf{v}_k,$$

não ocorrendo nenhum aprendizado a partir desta observação. Portanto, esta observação \mathbf{x}_k é ignorada.

No segundo caso extremo de (3.2), $\rho_k = 1$. A observação está totalmente de acordo com as crenças atuais. Neste caso, supondo $\alpha = 1$ tem-se:

$$\mathbf{v}_{k+1} = \mathbf{x}_k,$$

e o sistema fica totalmente aberto para novas informações.

Para que $\rho_k = 1$, a partir de (3.2), deve-se ter:

$$\frac{1}{p} \sum_{j=1}^p |x_{kj} - v_{kj}| = 0.$$

o que implica, para todo $j=1, \dots, p$

$$x_{kj} - v_{kj} = 0.$$

Logo $x_{kj} = v_{kj}$, então:

$$v_{k+1j} = v_{kj} + \alpha(x_{kj} - v_{kj})$$

$$v_{k+1j} = v_{kj}.$$

Portanto, pode-se concluir, que a revisão das crenças ocorre somente quando os vetores v_{kj} e x_{kj} são próximos, mas não exatamente iguais. Considerando o algoritmo de aprendizagem participativa básico, para qualquer j obtém-se:

$$v_{k+1j} = v_{kj} + \rho_k (x_{kj} - v_{kj}).$$

A maior variação do processo no valor de v_{k+1} ocorre quando

$$\Delta_{kj} = \rho_k |x_{kj} - v_{kj}| \alpha,$$

é máximo. Sem perda de generalidade, vamos supor que $\alpha = 1$, e a distância de Hamming no cálculo de ρ_k . Assim,

$$\rho_k = 1 - \frac{1}{p} \sum_{j=1}^p |x_{kj} - v_{kj}|.$$

Seja,

$$\xi_{kj} = |x_{kj} - v_{kj}|,$$

substituindo ξ_{kj} e ρ_k em Δ_{kj} obtém-se :

$$\Delta_{kj} = \xi_{kj} \left(1 - \frac{1}{p} \sum_{m=1}^p \xi_{km}\right),$$

o qual pode ser reescrito como:

$$\Delta_{kj} = \xi_{kj} \left(1 - \left(\frac{1}{p} \xi_{kj} + \frac{1}{p} \sum_{m \neq j}^p \xi_{km} \right) \right).$$

Denotando ξ_{kj} por a e $\sum_{m \neq k}^p \xi_{km}$ por b , a igualdade acima pode ser expressa como:

$$\Delta_k = a \left(1 - \left(\frac{1}{p} a + \frac{1}{p} b \right) \right) = \left(a - \frac{a^2}{p} - \frac{ab}{p} \right),$$

onde $a \in [0, a_1]$ e $b \in [0, b_2]$ e

$$a_1 = \max[1 - v_{kj}, v_{kj}]$$

$$b_2 = \sum_{m \neq k} \max[1 - v_{km}, v_{km}].$$

Supondo que $p \geq 2$ observa-se que o máximo de Δ_j ocorre quando

$$\sum_{m \neq j}^p \xi_{km} = 0,$$

com

$$\xi_{kj} = a_1 = \max[1 - v_{kj}, v_{kj}].$$

Logo

$$\Delta_j = a_1(1 - a_1 / p).$$

Então, o máximo de aprendizado ocorre quando todos os outros componentes são semelhantes e o componente j da observação atual é o valor mais distante das crenças atuais. Isto é, a aprendizagem máxima ocorre de um modo passo a passo, componente por componente do vetor de crenças v_k .

3.3 Sistema de Aprendizagem Participativa

Sabe-se que ρ_k proporciona uma medida do grau de compatibilidade entre as observações e as crenças atuais do sistema. Contudo, observe que ρ_k ignora o fato de que as crenças podem estar em conflito com as observações durante um certo período de tempo e, sendo assim, o sistema estará ignorando estas observações. Se este tipo de

situação ocorre durante uma seqüência de observações, isto é, muitas observações apresentam-se incompatíveis com o conhecimento do sistema, então este deverá ser alertado e se tornar mais aberto a aprender estas observações ignoradas. Este alerta é necessário porque não são as observações, mas sim as crenças adquiridas pelo sistema é que podem estar erradas.

Uma das possíveis maneiras de introduzir um mecanismo de alerta é monitorar o valor do índice de compatibilidade das crenças atuais com as observações e fornecer um índice de alerta para subsequente aprendizagem, conforme sugere a Figura 3.2. Neste caso, o índice de alerta deve ser tal que, quanto maior o seu valor, menor deve ser a confiança do sistema atual. Desta forma, proporciona-se uma maior chance para que as observações em conflito contribuam para atualizar as crenças do sistema.

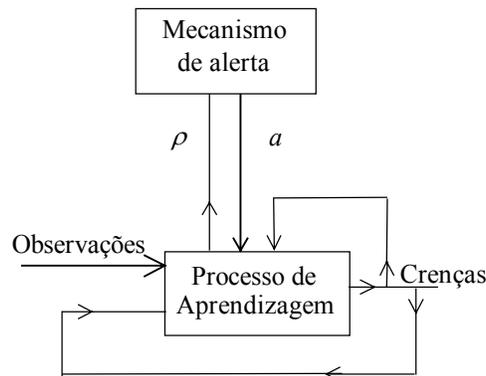


Figura 3.2: Aprendizagem participativa com índice de alerta.

A Figura 3.2 mostra a aprendizagem participativa com o índice de alerta. Neste caso, o mecanismo de alerta monitora o desempenho do sistema acompanhando a compatibilidade das crenças atuais com as observações. Esta informação retorna um alerta para o sistema. Este alerta irá interferir no processo de aprendizagem.

O índice de alerta será denotado por $a_k \in [0,1]$. Quanto maior a_k , mais alerta estará o sistema e maior o interesse pelas observações conflitantes. A dinâmica de atualização do índice de alerta é especificada da seguinte forma:

$$a_{k+1} = a_k + \beta((1 - \rho_{k+1}) - a_k) . \quad (3.4)$$

Analisando a expressão (3.4) é possível verificar dois casos extremos: o primeiro, com $\rho_{k+1}=1$, isto é, alto grau de compatibilidade. Portanto,

$$a_{k+1} = a_k + \beta a_k,$$

e, neste caso, o índice de alerta diminui. Por outro lado, se $\rho_{k+1} = 0$,

$$a_{k+1} = a_k + \beta(1 - a_k),$$

o índice de alerta aumenta. A taxa $\beta \in [0, 1]$ modula o grau de variação do índice de alerta. O valor de β pode ser visto como o complemento do grau de conservadorismo do sistema. Neste trabalho, assumimos que β é uma constante.

O índice de alerta pode ser visto como um grau de desconfiança na atual estrutura de crenças, ou também como o inverso do índice ρ_k . Incorporando o índice de alerta à aprendizagem participativa básica (3.1), obtém-se:

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \alpha(\rho_k)^{1-a_k} (\mathbf{x}_k - \mathbf{v}_k). \quad (3.5)$$

Neste caso, observa-se que se $a_k = 0$, se obtém o sistema original. Por outro lado, se o índice de alerta aumentar, o termo $(\rho_k)^{1-a_k}$ também aumenta. No caso particular em que $a_k = 1$ (alerta máximo), $1 - a_k = 0$. Portanto:

$$(\rho_k)^{1-a_k} = (\rho_k)^0 = 1$$

e

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \alpha(\mathbf{x}_k - \mathbf{v}_k).$$

Neste caso, o efeito de ρ_k torna-se inócuo.

De modo geral, a aprendizagem participativa pode ser formulada como:

$$\mathbf{v}_{k+1} = \mathbf{v}_k + G(a_k, \rho_k)(\mathbf{x}_k - \mathbf{v}_k),$$

onde

$$G(a_k, \rho_k) = \alpha(\rho_k)^{1-a_k}.$$

Uma forma alternativa para G é:

$$G(a_k, \rho_k) = \lambda(\rho_k \alpha)^{1-a_k}.$$

No caso de um ambiente plenamente compatível a taxa de aprendizagem é $\lambda\alpha$, pois $\rho_k = 1$, $a_k = 0$. Por outro lado, quando o sistema está em alerta máximo, a taxa de aprendizagem é λ .

Uma outra alternativa para G é:

$$G(a_k, \rho_k) = \lambda(a_k + \rho_k\alpha - a_k\rho_k\alpha).$$

Neste caso, quando o alerta é máximo $a_k = 1$, obtém-se:

$$G(a_k, \rho_k) = \lambda$$

e quando o alerta for mínimo, $a_k = 0$ e $\rho_k = 1$, a taxa de aprendizagem será:

$$G(a_k, \rho_k) = \lambda\alpha.$$

A Figura 3.3 fornece uma visão geral do sistema de aprendizagem participativa.

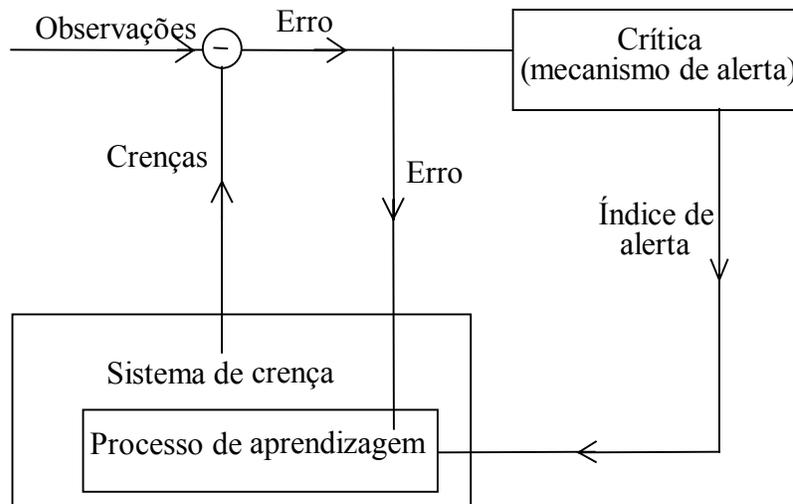


Figura 3.3: Sistema de aprendizagem participativa.

Pode-se dizer que a aprendizagem participativa introduz em seu sistema um equilíbrio, pois, enquanto ρ_k determina quanto o sistema pode aumentar a credibilidade nas suas próprias crenças, a_k acompanha o comportamento das crenças perante as observações com um índice de alerta, o que pode ser considerado como uma crítica. É importante notar que a crítica, ou o índice de alerta, é independente do sistema de crenças. A independência da crítica com as crenças é crucial para se acompanhar e avaliar, com imparcialidade, o desempenho do sistema. A crítica afeta o processo de

aprendizagem através do índice de alerta e providencia uma avaliação objetiva do desempenho do sistema de crença. Assim é possível avaliar o quanto se pode confiar no que o sistema aprendeu e está aprendendo.

3.4 Resumo

Este capítulo abordou o conceito de aprendizagem participativa de Yager descrevendo suas características particulares. A primeira característica principal deste tipo de aprendizagem é a influência do conhecimento adquirido na receptividade de novas observações durante a aprendizagem. A segunda característica importante é a inclusão de um índice de alerta que avalia o grau de confiabilidade das crenças no sistema.

No próximo capítulo, como proposta principal deste trabalho, é apresentada a aplicação da aprendizagem participativa na resolução do problema de agrupamento nebuloso de dados .

4. Aprendizagem Participativa em Agrupamento

4.1 Introdução

Este capítulo desenvolve dois novos algoritmos de agrupamento baseados no conceito de aprendizagem participativa. A principal característica destes algoritmos é a capacidade de determinar um número aceitável de grupos em um conjunto X , sendo, portanto, algoritmos não supervisionados, conforme definido na Seção 2.1.1.

Vale ressaltar de imediato que, em analogia com o paradigma de aprendizagem participativa, os pontos x_k e os centros v_i dos algoritmos propostos neste capítulo correspondem às observações e às crenças, respectivamente. O grau de compatibilidade ρ_{ki} da observação k com a crença i atual do sistema é o grau de compatibilidade entre x_k e o centro v_i . O mesmo acontece com o índice de alerta a_{ki} , onde se avalia a incompatibilidade das crenças com relação às observações, isto é, o índice avalia o quão distante x_k está em relação aos centros v_i , $i = 1, \dots, c$.

Nas próximas seções, os algoritmos desenvolvidos neste trabalho serão descritos e comparados com os algoritmos apresentados na Seção 2.5 .

4.2 Algoritmo A - Aprendizagem Participativa

O algoritmo **A**, proposto nesta seção, está baseado nos conceitos de aprendizagem participativa e realiza as adaptações necessárias para resolução de problemas de agrupamento. Estas adaptações permitiram que este algoritmo seja capaz de determinar um número adequado de grupos em X .

Uma alteração importante feita pelo algoritmo **A** em relação ao sistema de aprendizagem participativa é a introdução de um parâmetro adicional τ , além dos dois (α e β) já existentes. Este novo parâmetro τ é um limiar cujo propósito é detectar quando uma observação está demasiadamente incompatível com o conhecimento do sistema, isto é, quando um ponto está muito distante dos centros das classes já conhecidas. Deste

modo, o sistema é motivado a considerar a observação como uma nova crença. Uma opção, adotada neste trabalho, para uma observação ser um novo conhecimento do sistema, é declarar \mathbf{x}_k como um possível centro que representa um novo grupo, tornando-se uma nova crença “aprendida” pelo sistema.

Os valores do parâmetro τ , para os casos particulares considerados neste trabalho, são obtidos experimentalmente. Portanto, deixa-se em aberto a determinação deste parâmetro por outros métodos, como por exemplo através dos algoritmos genéticos.

Em consonância com a Seção 2.4, o algoritmo **A** particiona um conjunto $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, $\mathbf{x}_k \in R^p$, $k=1,2,\dots, n$, em c , $2 \leq c \leq n$ subconjuntos de X . A partição procurada é representada por uma matriz de compatibilidade ρ ($n \times c$). O elemento ρ_{ki} $i = 1,2, \dots, c$ de ρ representa o grau de similaridade do ponto \mathbf{x}_k com o grupo i , grupo este representado pelo seu centro $\mathbf{v}_i \in R^p$. Esta matriz de compatibilidade possui as características de uma partição possibilística conforme (2.18), Seção 2.4 isto é:

$$M_{pcn} = \left\{ U \in [0,1]^{nc} \mid \sum_{i=1}^c u_{ki} > 0, k = 1, \dots, n. \right\} \quad (4.1)$$

O algoritmo **A** procede de acordo com o seguinte: Dados os valores iniciais para os parâmetros α , β e τ , escolhe-se aleatoriamente dois pontos de X como inicialização de V^0 , cujos elementos são os centros de grupos. Calculam-se os índices de compatibilidade ρ_{ki} e de alerta a_{ki} . Em seguida, verifica-se para todos os \mathbf{v}_i se o índice de alerta de \mathbf{x}_k é maior do que o limiar τ . Caso o índice de alerta seja maior que o limiar τ , então \mathbf{x}_k é declarado como o centro de um novo grupo. Se não existir então o sistema ajusta o centro mais próximo do ponto observado, isto é, o centro \mathbf{v}_s que possui o maior índice de compatibilidade com \mathbf{x}_k . Após o ajuste dos centros ou a criação de um centro, verifica-se se há classes redundantes. Esta verificação é necessária porque ao se ajustar um centro, este poderá ficar muito próximo de outro centro, ambos representando o mesmo grupo. Para a aprendizagem participativa isto significa que há crenças redundantes, e um conhecimento poderá ser descartado porque não representa perdas de informação sob o ponto de vista da aprendizagem.

O algoritmo de aprendizagem participativa para agrupamento, proposto neste trabalho, possui os seguintes passos:

<p>Entrada: $\mathbf{x}_k \in [0,1]^p, k = 1, \dots, n, \alpha \in [0,1], \beta \in [0,1], \tau \in [0,1], \varepsilon > 0$ e $m > 0$. Definir o número máximo l_{\max} de iterações. Inicializar $c=2$, $V^0 = \{\mathbf{v}_1, \mathbf{v}_2\}$, U^0 de acordo com $V^0, l=1, a_{ki}^0 = 0, k=1,2, \dots, n$ e $i=1, \dots, c$.</p> <p>Saída: $c, \mathbf{v}_i \in [0,1]^p, i=1, \dots, c$ e ρ.</p> <p>1 Para $k=1, \dots, n$; 1.1 Para $i=1, \dots, c$; $\rho_{ki}^l = 1 - \frac{1}{p} \sum_{j=1}^p x_{kj} - v_{ij} \quad // \text{Calcular a compatibilidade} \quad (4.2)$ 1.2 Para $i=1, \dots, c$; Calcular a_{ki}^l $a_{ki}^l = a_{ki}^{l-1} + \beta [(1 - \rho_{ki}^l) - a_{ki}^{l-1}] \quad // \text{Calcular índice de alerta} \quad (4.3)$ 1.3 Se $a_{ki}^l \geq \tau, \forall i \in \{1, \dots, c\}$ então Criar centro Senão Ajustar o \mathbf{v}_s de acordo com: $\mathbf{v}_s^l = \mathbf{v}_s^{l-1} + \alpha \rho_{ks}^{1-a_{ks}^l} (\mathbf{x}_k - \mathbf{v}_s^{l-1}) \quad (4.4)$ Onde: $s = \arg \max_i \{\rho_{ki}\}$ 1.4 Atualizar número de centros Calcular o índice de compatibilidade entre os centros: Para $i = 1, \dots, c-1$ Para $j = i+1, \dots, c$ $\rho_{v_i}^l = 1 - \sum_{h=1}^p v_{ih}^l - v_{jh}^l ^2 \quad (4.5)$ $\lambda_{v_i}^l = \beta (1 - \rho_{v_i}^l) \quad (4.6)$ Se $\lambda_{v_i}^l \leq 0.95 \cdot \tau$ Eliminar \mathbf{v}_i Recalcular U eliminando i. 2. Determinar o erro $erro = \ V^l - V^{l-1}\ = \max_{ij} v_{ij}^l - v_{ij}^{l-1} \quad (4.7)$ 3. Se $erro > \varepsilon$ e $l < l_{\max}$, $l = l + 1$ e retornar para o passo (1); senão parar. 4. Recalcular a matriz de pertinência. Fim procedimento.</p>

Algoritmo A: Aprendizagem participativa para agrupamento nebuloso de dados.

O algoritmo termina quando o número de iterações for igual ao máximo l_{\max} especificado pelo usuário, ou quando os centros não se alterarem significativamente de

uma iteração para as seguintes. Como as posições dos centros são atualizadas *ponto a ponto*, a pertinência de um ponto k a um grupo i , poderá não ser mais a mesma quando um outro ponto $k+1$ for processado. Neste caso, quando um dos critérios de parada forem satisfeitos, é necessário atualizar os graus de compatibilidade de todos os pontos x com os c centros para que a matriz de compatibilidade corresponda às classes atuais.

Para o caso das crenças redundantes, uma alternativa para decidir quais são estes centros é verificar o índice de compatibilidade de cada centro com todos os outros. Neste trabalho utiliza-se o quadrado da norma Euclidiana para se determinar as compatibilidades entre os centros. O valor do limiar utilizado para eliminar um centro redundante é determinado experimentalmente. Verificou-se que, para a maioria dos resultados satisfatórios com os conjuntos de dados tratados neste trabalho, os quais são mostrados no próximo capítulo, o valor deste limiar está em torno de λ , onde:

$$\lambda = 0.95 \cdot \tau . \tag{4.8}$$

Elimina-se um centro sempre que o índice λ_{v_i} de um centro i para outro for menor ou igual a λ , ou seja, quando o índice de compatibilidade entre dois centros for muito grande. O índice λ_{v_i} dado por (4.6), é inspirado no índice de alerta (4.3). Contudo, λ_{v_i} não considera o seu histórico e, portanto, $\lambda_{v_i}^{l-1}$ é descartado. Optou-se por este índice para que, deste modo, o mesmo fator τ , que auxilia na criação de um centro, pudesse também influenciar na sua exclusão. Para compreender melhor os centros redundantes, observe a Figura 4.1.

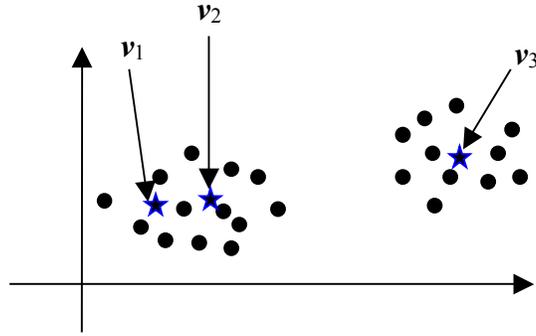


Figura 4.1: Exemplo de centros redundantes.

Os centros v_1 e v_2 estão representando o mesmo grupo. Neste caso, a compatibilidade entre v_1 e v_2 deverá ser grande e, portanto, o índice λ_{12} , λ do centro v_1 com v_2 , deverá ser menor ou igual a λ , equação (4.8), eliminando-se, por exemplo, o centro v_1 analisado. O mesmo não acontece quando comparadas a compatibilidade λ_{23} entre os centros v_2 e v_3 , que deverá ser baixa, indicando que os centros estão representando grupos distintos e, portanto, as crenças do sistema permanecerão as mesmas, não sendo eliminado nenhum centro.

Esta seção desenvolveu um algoritmo de agrupamento com a aprendizagem participativa. Este algoritmo será a base para o algoritmo da próxima seção.

4.3 Algoritmo B - Aprendizagem Participativa com Matriz de Covariância

O algoritmo **B** baseia-se no algoritmo da Seção 4.2. Também, é um algoritmo não supervisionado. Portanto, este algoritmo também determina uma partição nebulosa com um número *aceitável* de classes. O algoritmo **B** mostra a flexibilidade da aprendizagem participativa em adotar outras medidas para se avaliar a compatibilidade entre as observações e as crenças do sistema.

A distância adotada pelo algoritmo neste trabalho, é a mesma adotada pelo algoritmo GK Seção 2.5.2. A distância de Mahalanobis foi escolhida por ser capaz de capturar grupos que possuam tanto formatos esféricos como elipsoidais (Gustafson and

Kessel, 1979). A distância é determinada conforme a expressão (2.25), novamente mostrada abaixo:

$$d_{ki} = (\mathbf{x}_k - \mathbf{v}_i^l)^T \left[(\det(F_i))^{1/n+1} F_i^{-1} \right] (\mathbf{x}_k - \mathbf{v}_i^l), \quad (4.9)$$

$$i = 1, 2, \dots, c, \quad k = 1, 2, \dots, n.$$

onde F_i é a matriz de covariância associada ao i -ésimo centro, determinada por (2.24), isto é:

$$F_i = \frac{\sum_{k=1}^n [u_{ki}^{l-1}]^m (\mathbf{x}_k - \mathbf{v}_i^l)(\mathbf{x}_k - \mathbf{v}_i^l)^T}{\sum_{k=1}^n [u_{ki}^{l-1}]^m}, \quad \forall i, \quad i = 1, 2, \dots, c. \quad (4.10)$$

A partição procurada neste algoritmo é uma partição nebulosa representada por uma matriz de pertinência U ($n \times c$) (ver Seção 2.1.1). O elemento u_{ki} ($i=1,2,\dots,c$) de U representa o grau de pertinência do ponto k ao grupo i , grupo este representado pelo seu centro $\mathbf{v}_i \in R^p$. Os graus de pertinência são determinados conforme a expressão (2.21), repetida abaixo:

$$u_{ki}^l = \frac{1}{\sum_{j=1}^c (d_{ki} / d_{kj})^{1/(m-1)}} \quad (4.11)$$

onde m é o fator que modula a nebulosidade da partição. Portanto, a matriz de pertinência nebulosa calculada terá as mesmas propriedades e restrições da partição nebulosa (2.16).

$$M_{fcn} = \left\{ U \in [0,1]^{nc} \mid \sum_{i=1}^c u_{ki} = 1, k = 1, \dots, n, \sum_{k=1}^n u_{ki} > 0, i = 1, \dots, c \right\}$$

O algoritmo de agrupamento **B** utilizando matriz de covariância, desenvolvido neste trabalho, é resumido com os seguintes passos:

Entrada: $\mathbf{x}_k \in [0,1]^p, k = 1, \dots, n, \alpha \in [0,1], \beta \in [0,1], \tau \in [0,1], \varepsilon > 0$ e $m > 0$.
Definir o número máximo l_{\max} de iterações. Inicializar $c=2$,
 $V^0 = \{\mathbf{v}_1, \mathbf{v}_2\}$, U^0 de acordo com V^0 , $l=1$, $a_{ki}^0 = 0, k = 1, 2, \dots, n$ e $i = 1, \dots, c$.

Saída: $c, \mathbf{v}_i \in [0,1]^p, i=1, \dots, c$ e U .

- 1 Para $k = 1, \dots, n$;
 - 1.1 Para $i = 1, \dots, c$;

Calcular a matriz de covariância com a Eq.(4.10)
Calcular $d_{ik}(\mathbf{x}_k, \mathbf{v}_i)$ com a Eq. (4.9)
Calcular ρ_{ik}^l

$$\rho_{ki}^l = 1 - d_{ki} \quad //\text{Calcular a compatibilidade} \quad (4.12)$$
 - 1.2 Para $i=1, \dots, c$;

Calcular a_{ki}^l

$$a_{ki}^l = a_{ki}^{l-1} + \beta \left[(1 - \rho_{ki}^l) - a_{ki}^{l-1} \right] \quad //\text{Calcular índice de alerta}$$
 - 1.3 Se $a_{ki}^l \geq \tau, \forall i \in \{1, \dots, c\}$ então

Criar centro
Senão
Ajustar o \mathbf{v}_s de acordo com:

$$\mathbf{v}_s^l = \mathbf{v}_s^{l-1} + \alpha \rho_{ks}^{1-a_{ks}^l} (\mathbf{x}_k - \mathbf{v}_s^{l-1})$$

Onde: $s = \arg \max_i \{\rho_{ki}\}$
 - 1.4 Atualizar número de centros

Calcular o índice de compatibilidade entre os centros:
Para $i = 1, \dots, c-1$
Para $j = i+1, \dots, c$

$$\rho_{v_i}^l = 1 - \sum_{h=1}^p \left| v_{ih}^l - v_{jh}^l \right|^2$$

$$\lambda_{v_i}^l = \beta (1 - \rho_{v_i}^l)$$

Se $\lambda_{v_i}^l \leq 0.95 \cdot \tau$
Eliminar \mathbf{v}_i
Recalcular U eliminando i .
2. Determinar o erro
$$erro = \|V^l - V^{l-1}\| = \max_{ij} |v_{ij}^l - v_{ij}^{l-1}|$$
3. Se $erro > \varepsilon$ e $l < l_{\max}$, $l = l + 1$ e retornar para o passo (1);
senão parar.
4. Recalcular a matriz de pertinência.

Fim procedimento.

Algoritmo B: Aprendizagem Participativa com covariância.

O algoritmo B procede da mesma maneira que o algoritmo A descrito na seção anterior.

O algoritmo termina quando o número de iterações for igual ao máximo l_{\max} especificado pelo usuário, ou quando os centros não se alterarem significativamente de uma iteração para as seguintes. Assim como no algoritmo **A**, as posições dos centros são atualizadas *ponto a ponto*. Portanto, a pertinência de um ponto k a um grupo i , poderá não ser mais a mesma quando um outro ponto $k+1$ for processado. Neste caso, quando um dos critérios de parada forem satisfeitos, é necessário atualizar os graus de pertinência de todos os pontos x com os c centros para que a matriz de pertinência corresponda às classes atuais.

Note que, no algoritmo **B**, o valor de ρ_{ki}^l permanece indicando o grau de compatibilidade entre o ponto observado e todas as classes (crenças) conhecidas pelo sistema, o cálculo para o índice de alerta a_{ki}^l permanece o mesmo, o mesmo acontecendo com o parâmetro adicional τ que é uma referência para a disparidade entre os centros e o ponto analisado. Contudo, na expressão (4.3) a distância é alterada, trocando-se a distância de Hamming pela distância de Mahalanobis como é observado na expressão (4.10). Esta alteração foi realizada para ampliar a capacidade do algoritmo para detectar grupos com formas elipsoidais.

A Tabela 4.1 apresenta um resumo das principais características dos algoritmos **A** e **B**.

Tabela 4.1: Principais características dos algoritmos A e B.

Alg.	Categoria	Tipo de dado	$O()$ *	Geometria	Ruído	Parâmetro	Resultado
A	Agrupamento por Partição possibilística	Numérico	$O(nc^3p)$	Formas convexas	Não	α, β, τ	Centros das classes e matriz de pertinência
B	Agrupamento por Partição nebulosa e possibilística	Numérico	$O((n^2c^2p^3) + (nc^3p))$	Formas não convexas	Não	α, β, τ	Centros das classes e matriz de pertinência

* n = número de pontos em X ; c = número de centros; p = dimensão de X . A Análise de complexidade está discutida na Seção 5.4.2.

A seguir discute-se **A** e **B** com os algoritmos apresentados na Seção 2.5.

4.4 Discussão sobre os Algoritmos

Esta seção discute as principais características dos algoritmos **A** e **B** comparando com os algoritmos FCM, GK, FCL, FFSECL, MFKM apresentados na Seção 2.5. A Tabela 4.2 resume as características funcionais destes algoritmos.

Tabela 4.2: Características funcionais de cada algoritmo.

Características Funcionais	Algoritmos de agrupamento						
	FCM	GK	FCL	FFSECL	MFKM	A	B
Função objetivo	S	S	N	N	S	N	N
Conceitos da teoria dos conjuntos nebulosos	S	S	S	S	S	S	S
Atualização dos centros	Batelada	Batelada	Ponto-a-Ponto	Ponto-a-Ponto	Batelada	Ponto-a-Ponto	Ponto-a-Ponto
Medida de similaridade	DE	DM	DE	DM	DIM	DH	DM
Paradigma de aprendizagem	N	N	AC	AC	N	AP	AP
Estimação da densidade	N	N	N	N	S	S	S
Parâmetros	Nc	Nc	Nc e Ta	Nc e Ta	Cg	Tc	Tc
Partição	F	F	F	F	F	P	F

Legenda:

- N: Não; S: Sim;
- DE : Distância Euclidiana;
- DH : Distância de Hamming;
- DIM : Métrica de indução de dados (*Data Induced Metric*);
- DM : Distância de Mahalanobis;
- AC : Aprendizagem competitiva;
- AP : Aprendizagem participativa;
- Nc : Número de centros;
- Ta : Taxa de aprendizagem;
- Cg : Coeficiente de grade;
- NMC : Número máximo de centros;
- Tc : Taxa de compatibilidade admitida;
- F : Nebulosa;
- P : Possibilística.

Com a primeira característica da Tabela 4.2, a função objetivo, pode-se dividir os algoritmos de agrupamento entre a otimização alternada – AO e estimação alternada de grupos – ACE. Isto é, entre os algoritmos que otimizam uma função objetivo entre aqueles que utilizam-se de outros mecanismos que não sejam decorrentes de um problema de otimização. Os algoritmos AO são: o FCM, GK e MFKM, e os algoritmos ACE são: o FCL, FFSECL, **A** e **B**.

A Tabela 4.2 mostra a existência de algumas similaridades entre os algoritmos FCL, FFSECL, **A** e **B**. A principal semelhança entre estes algoritmos são as expressões de atualização dos centros (2.27) e (4.4), novamente apresentadas abaixo:

$$\mathbf{v}_i = \mathbf{v}_i + \alpha u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i) \quad i = 1, \dots, c, \quad k = 1, 2, \dots, n \quad (4.13)$$

$$\mathbf{v}_s = \mathbf{v}_s + \alpha \rho_{ks}^{1-a_{ks}} (\mathbf{x}_k - \mathbf{v}_s) \quad k = 1, 2, \dots, n, \quad s = \arg \max_i (\rho_{ki}), \quad i = 1, \dots, c \quad (4.14)$$

Contudo, estas atualizações de centros possuem uma diferença. Os algoritmos FCL, FFSECL atualizam todos os centros de acordo com cada pertinência u_i , $i = 1, \dots, c$, do ponto k . Para os algoritmos **A** e **B**, neste trabalho, adotou-se a atualização de um único centro s para cada ponto k . Deste modo, apenas o centro com maior possibilidade é alterado, ou seja, a crença com maior compatibilidade com a observação k é alterada.

Portanto, pode-se concluir que, quem explora o conceito de aprendizagem competitiva é a expressão (4.14) que escolhe o centro s vencedor para ser atualizado. Conseqüentemente, os algoritmos propostos neste trabalho podem ser também considerados algoritmos de aprendizagem participativa competitiva.

A expressão (4.14) pode ser reescrita da seguinte forma:

$$\mathbf{v}_s = \mathbf{v}_s + \alpha (\rho_{ks}^{1-a_{ks}})^m (\mathbf{x}_k - \mathbf{v}_s) \quad k = 1, 2, \dots, n \text{ e } s = \arg \max_i (\rho_{ki}) \quad i = 1, \dots, c \quad ^1; \quad (4.15)$$

deste modo, a aprendizagem participativa incorpora o fator que modula a nebulosidade da partição. Como pode ser observado nas expressões (4.13) e (4.15) o conceito de aprendizagem competitiva nebulosa poderia ser um caso especial da aprendizagem participativa, caso também realizasse a atualização apenas do centro vencedor.

¹ Testes foram realizados com o fator $m > 1$ no algoritmo AP-**B**, não se observando uma influência no número de grupos.

Outra diferença entre (4.13) e (4.14) é a representação dos graus de pertinência para a atualização dos centros. A matriz U representa os graus de pertinência de uma partição nebulosa, e P é uma matriz de partição possibilística, sendo que estas partições possuem características e restrições diferentes.

Contudo, a principal diferença está na aplicação de uma função para o índice de alerta a_{ks} pela expressão (4.14). Já a expressão (4.13), aplica um fator constante m de *fuzzificação*. Este índice de alerta foi crucial para os algoritmos de agrupamento **A** e **B**, influenciando diretamente no processo de aprendizagem participativa não supervisionada.

Uma das características, não citada na Tabela 4.2, são as subdivisões *aglomerativa* e *divisiva* presentes no agrupamento hierárquico, comentadas na Seção 2.3.3. Estas subdivisões podem ser aplicadas em outros tipos de algoritmos de agrupamento, mesmo não se utilizando técnicas hierárquicas.

Os algoritmos não supervisionados deste trabalho, MFKM, A e B, são algoritmos divisivos, isto é, são algoritmos que partem de um número baixo de grupos (1 ou 2 por exemplo), e por divisões sucessivas, obtém-se 2, 3, ... grupos, se este for o caso. Um exemplo de algoritmo não supervisionado aglomerativo é o algoritmo E-FCM (Kaymak and Setnes, 2000). Este algoritmo parte de um número grande para c , normalmente um pouco menor que n , e procede ao longo das iterações eliminando grupos até que um número adequado seja determinado.

Esta seção apresentou algumas comparações entre os algoritmos de agrupamento apresentados neste trabalho. O que se pode concluir nesta seção é que os algoritmos vão adquirindo estruturas mais complexas à medida que se deseja aumentar a capacidade dos algoritmos em detectar grupos com formas diversificadas. Quando este algoritmo é do tipo não supervisionado a complexidade é ainda maior.

4.5 Resumo

Neste capítulo foram apresentados dois novos algoritmos para agrupamento nebuloso de dados, ambos desenvolvidos aplicando conceitos de aprendizagem participativa. Também foram realizadas comparações entre os algoritmos propostos e os principais algoritmos de agrupamento apresentados na literatura.

No próximo capítulo serão apresentados e discutidos os resultados dos algoritmos FCM, GK, FCL, FFSECL, MFKM e AP-**B**. Isto porque o algoritmo AP-**A** obteve resultados muito semelhantes com o algoritmo FCM como mostra o Apêndice I. Além disso, o algoritmo AP-**B** obteve resultados significativamente melhores comparados com o algoritmo AP-**A**.

5. Resultados e Discussões

5.1 Introdução

Este capítulo apresenta os resultados obtidos pelos algoritmos, citados nos capítulos dois e quatro deste trabalho. Os testes, para se obter estes resultados, foram realizados em conjuntos de dados selecionados de acordo com suas características e grau de dificuldade da estrutura de grupos. Os resultados obtidos são comparados frente aos critérios e requisitos de desempenho necessários aos algoritmos de agrupamento. Os algoritmos também são analisados segundo os índices de validação apresentados no Capítulo 2.

5.2 Requisitos e Critérios de Desempenho

Segundo Han e Kamber (2001) o agrupamento de dados é um campo que exige requisitos particulares. Os requisitos típicos de algoritmos de agrupamento de dados são as seguintes:

- a) **Escalabilidade:** Os algoritmos apresentam geralmente um melhor desempenho computacional no agrupamento de um número pequeno de dados quando comparado com o desempenho para um conjunto maior de dados, por exemplo acima de 1000 dados. Há a necessidade de se preservar este desempenho computacional quando se trata de um número considerável de dados.
- b) **Habilidade de se adaptar aos diferentes tipos de atributos:** Além de dados numéricos, para os quais muitos dos algoritmos são projetados, existem aplicações que requererem agrupamentos de outros tipos de dados, como binário, caracteres, etc... ou uma mistura destes.
- c) **Encontrar grupos com formas arbitrárias:** Como observado no Capítulo 2, as distâncias: Euclidiana e de Manhattan capturam formas esféricas e cúbicas respectivamente, e são as principais medidas utilizadas pelos algoritmos de

agrupamento. Contudo, os grupos poderão conter outras formas que irão necessitar de outros critérios. Estudar e aplicar outras métricas torna-se, portanto, importante para desenvolver algoritmos que possam detectar grupos de formas arbitrárias.

- d) **Numero mínimo de parâmetros:** O parâmetro típico solicitado pelos algoritmos é o número desejado de grupos. No entanto, outros parâmetros podem ser necessários, e o resultado do agrupamento pode ser sensível à variação destes parâmetros. Em geral, valores apropriados para os parâmetros são difíceis de determinar. Portanto, quanto menor o número de parâmetros e quanto mais independente o algoritmo for do parâmetro, melhor o algoritmo.
- e) **Habilidade de trabalhar com ruído:** Os ruídos são observações que se desviam da média de outras observações de forma que se suspeite de que estas observações foram geradas por um mecanismo diferente. Há a necessidade de se verificar o comportamento dos algoritmos de agrupamento de dados perante os ruídos porque os conjuntos de dados do mundo real contêm erros, pontos isolados, desconhecidos ou dados imprecisos.
- f) **Insensibilidade à ordem de apresentação dos dados:** Alguns algoritmos são sensíveis à ordem em que os dados são apresentados. Alguns conjuntos de dados, quando apresentados em uma ordem diferente para um mesmo algoritmo, podem gerar grupos dramaticamente diferentes.
- g) **Alta dimensionalidade:** Com a percepção humana é possível julgar a qualidade do agrupamento em até três dimensões. Ainda é um grande desafio encontrar e avaliar grupos em espaços de grande dimensão. Neste caso, para se avaliar os agrupamentos é necessário utilizar índices que verificam a qualidade destes. Existe a necessidade de encontrar funções de validação que garantam um agrupamento aceitável sem a necessidade de utilizar a percepção humana.
- h) **Agrupamento com restrições:** Aplicações do mundo real podem requerer agrupamentos sob vários tipos de restrições. A tarefa é encontrar grupos de dados que satisfaçam estas restrições.

i) Legibilidade e usabilidade: O usuário espera por resultados de agrupamento que sejam legíveis, compreensíveis e usáveis. Isto é, os grupos precisam estar associados a uma determinada interpretação e analisados de acordo com uma aplicação.

Além dos requisitos de Han e Kamber (2001), Bezdek e Pal (1992), sugerem os seguintes itens de desempenho de algoritmos de agrupamento:

1. Escolha do número de grupos;
2. Inicialização dos centros dos grupos;
3. Ordem em que os dados são apresentados;
4. Propriedades geométricas dos dados;
5. Diversidade de formas geométricas nos grupos;
6. Compactação dos dados apresentados;
7. Variação na densidade dos dados entre os grupo;
8. Grau de separação entre os diversos grupos (sobreposição).

Apesar dos inúmeros trabalhos sobre agrupamento publicados, até o presente momento não se conhece um algoritmo que consiga atender a todos os requisitos de Han e Kamber (2001) simultaneamente e tampouco os itens de desempenho de Bezdek e Pal (1992). O que se percebe nos algoritmos disponíveis na literatura é a ênfase em um dos itens em detrimento dos outros.

Neste trabalho, os requisitos que não foram tratados foram:

- ruídos (e): Apesar de não serem menos importantes que os outros requisitos, o tratamento de ruídos foge do escopo deste trabalho. Contudo, pode-se observar que o novo algoritmo mostra uma flexibilidade quanto ao tratamento de ruído, e certamente este é um item a ser testado e avaliado no futuro.

- agrupamento com restrições (h): O agrupamento com restrições está presente principalmente nos algoritmos com funções objetivos, como por exemplo o algoritmo FCM e o GK, citados neste trabalho. Contudo, não são tratadas restrições nos algoritmos

de agrupamento com aprendizagem participativa elaborados neste trabalho, e portanto não estará no escopo deste trabalho.

- Este trabalho também não teve como objetivo testar a habilidade dos algoritmos em se adaptar com diferentes tipos de atributos (b). Todos os algoritmos aqui apresentados oferecem capacidade de adaptação para diferentes tipos de atributos. O algoritmo proposto neste trabalho também possui uma capacidade de adaptação para diferentes tipos de atributos e pode ser facilmente adaptado a outros tipos de dados. Contudo, este requisito foge do escopo deste trabalho.

Os pontos que serão aqui discutidos foram os requisitos e os itens de desempenho possíveis de serem observados com os testes realizados com os algoritmos de agrupamento juntamente com o auxílio da literatura. As discussões foram feitas principalmente com base no comportamento dos algoritmos perante os conjuntos de dados disponíveis na literatura. Estes conjuntos de dados, a serem apresentados na próxima seção, são diversificados quanto às formas dos grupos, número de dados e atributos para auxiliar na avaliação.

5.3 Características dos Conjuntos de Dados

Os algoritmos apresentados na Seção 2.5 e o algoritmo AP-**B** proposto são testados utilizando conjunto de dados reais e sintéticos, conforme as Figuras 5.1 a 5.5, todos eles organizados de acordo com seus respectivos grupos. Os conjuntos de dados artificiais foram selecionados para que se possa considerar diferentes graus de dificuldade para o agrupamento. O algoritmo AP-**A** também foi testado com estes conjuntos de dados e seus resultados se encontram no Apêndice I.

A seguir cada conjunto de dados será apresentado e comentado e na Tabela 5.1 é mostrado o número total de pontos, número de grupos e o número de pontos para cada grupo de cada conjunto de dados mostrados na Figura 5.1 a 5.5.

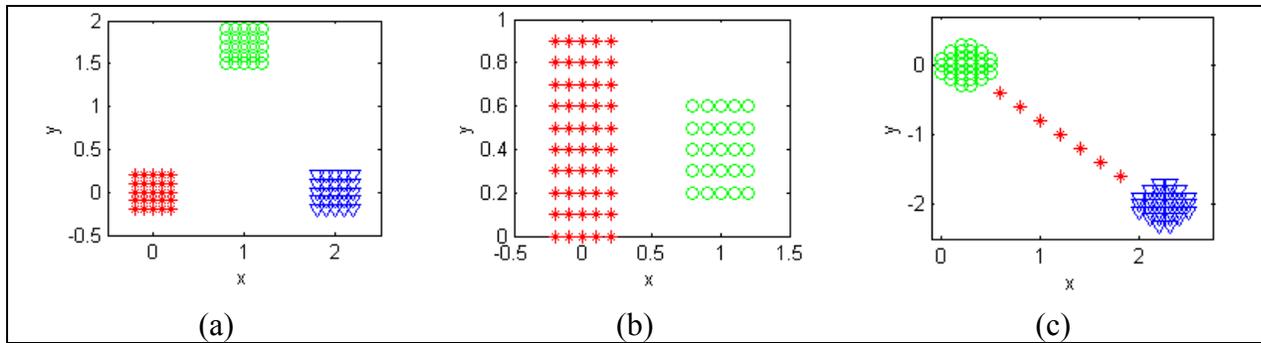


Figura 5.1: Conjunto de dados originais de a, b, c.

Figura 5.1.a: apresenta um conjunto de dados sugerido por Yager e Filev (1994), com um baixíssimo grau de dificuldade. Possui grupos bem definidos e separáveis, todos com o formato de um retângulo de lados iguais e com os mesmos números de pontos. Devido a estas características, este conjunto é utilizado para verificar se o algoritmo de agrupamento está correto. Isto porque, parte-se do princípio que qualquer algoritmo de agrupamento possa encontrar os grupos desejados neste conjunto de dados, independente das condições iniciais fornecidas, como a inicialização dos centros ou a ordem de apresentação dos dados. É possível observar os seguintes comportamentos dos algoritmos: convergência, sensibilidade à inicialização dos centros, sensibilidade aos parâmetros e à ordem em que os dados são apresentados.

Figura 5.1.b: mostra um conjunto de dados sugerido por Duda e Hart (1973) com características semelhantes à Figura 5.1.a, porém seus grupos possuem tamanhos diferentes. Pode-se analisar com este conjunto de dados: sensibilidade à inicialização dos centros, sensibilidade aos parâmetros, à ordem em que os dados são apresentados e à variação na densidade dos dados entre os grupo.

Figura 5.1.c: mostra um conjunto de dados também sugerido por Yager e Filev (1994). Este conjunto sugere tanto dois como três grupos. Os sete pontos localizados entre os dois grupos das extremidades podem ser considerados como: um grupo, pontos pertencentes aos outros dois grupos, bem como serem considerado como ruído. Isto mostra a subjetividade na determinação do número de grupos e a dificuldade da tarefa de agrupamento.

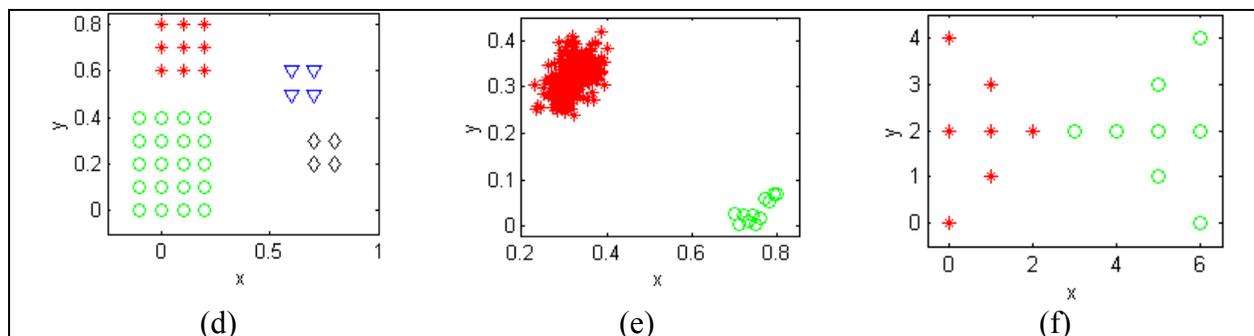


Figura 5.2: Conjunto de dados originais de d, e, f.

Figura 5.2.d: apresenta um conjunto de dados que possui quatro grupos, e dentre eles, dois possuem apenas quatro pontos cada. Com esse conjunto de dados avaliam-se nos algoritmos suas dependências à compactação dos dados, à variação na densidade dos dados entre os grupos e o grau de separação entre os grupos.

Figura 5.2.e: mostra um conjunto de dados que possui dois grupos distantes e com uma grande diferença de densidade entre eles. O conjunto possui 312 pontos, divididos em 301 pontos para o primeiro grupo e 11 pontos para o segundo. Este conjunto de dados tem por principal objetivo avaliar o comportamento dos algoritmos segundo a variação na densidade dos dados entre os grupos.

Figura 5.2.f: mostra um conjunto de dados sugerido pelo Klir (1995) também conhecido como *Borboleta*. É uma base de dados com poucos pontos, apenas 15, e pontos dispersos. Esta característica testa o algoritmo quanto a sua eficiência para grupos com baixa compactação. Este conjunto também mostra a subjetividade para a escolha do número de grupos e o agrupamento de dados. Por exemplo, o ponto central do conjunto de dados poderá pertencer tanto ao grupo da esquerda como ao grupo da direita.

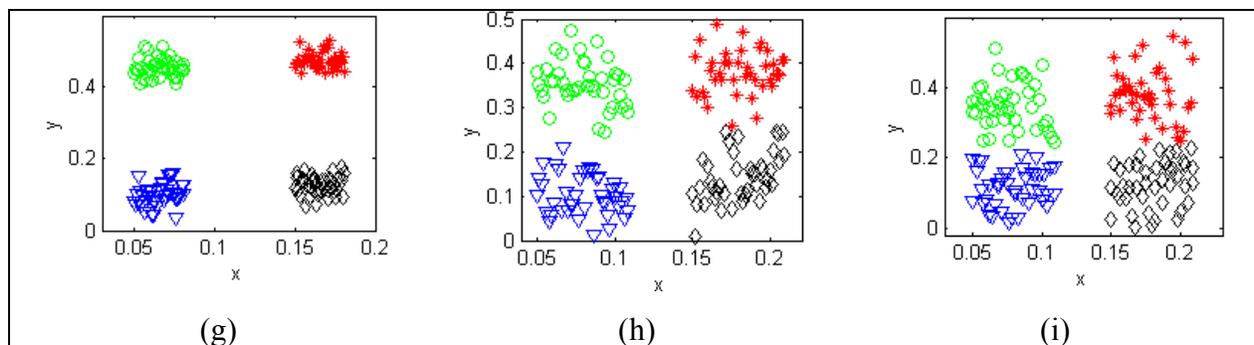


Figura 5.3: Conjunto de dados originais de g, h, i.

Figuras 5.3.g, 5.3.h e 5.3.i: são conjuntos de dados semelhantes aos sugeridos em (Windham, 1982; Roubens, 1982). Como mostram as figuras, há uma expectativa que os algoritmos encontrem quatro grupos pelo menos nas Figuras 5.3.g e 5.3.h. Porém, no conjunto de dados da Figura 5.3.i há uma sobreposição entre os grupos induzindo ao agrupamento para apenas dois grupos. Portanto estes conjuntos de dados possibilitam avaliar o comportamento dos algoritmos perante grupos sobrepostos.

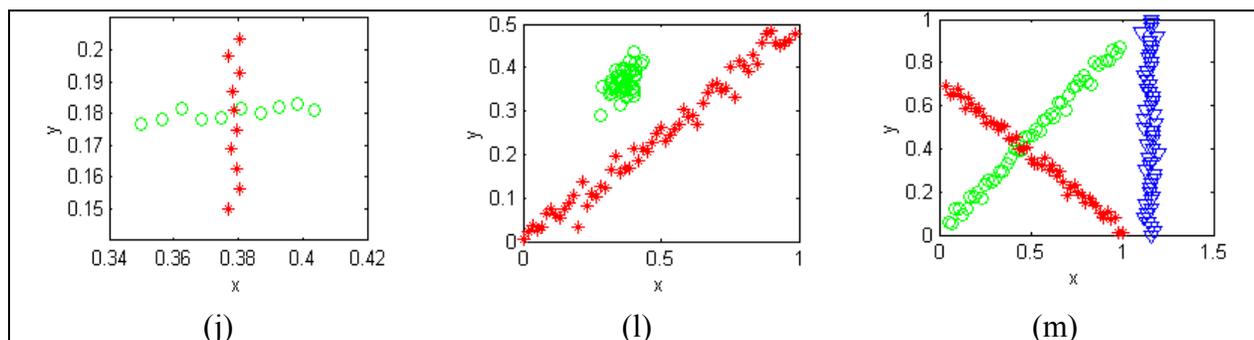


Figura 5.4: Conjunto de dados originais de j, l, m.

Figura 5.4.j: mostra o conjunto de dados reproduzido de Gustafson e Kessel (1979). Com poucos pontos e sobreposição de grupos com formas quase elipsoidais, tornou-se um exemplo clássico para ilustrar algoritmos que utilizam a distância de Mahalanobis.

Figura 5.4.l: apresenta um conjunto de dados de Liu e Xie (1995) que possui dois grupos com formas geométricas distintas, um com forma circular e outro quase elíptico. Pode-se observar, com este conjunto de dados, a capacidade de um algoritmo agrupar formas diferentes no mesmo conjunto de dados.

Figura 5.4.m: mostra um conjunto de dados muito semelhante ao sugerido por Gustafson e Kessel (1979) da Figura 5.4.j. Este conjunto de dados foi criado para verificar o comportamento dos algoritmos diante de um conjunto de dados que possua uma distribuição quase elipsoidal e grupos que tenham uma sobreposição de pontos.

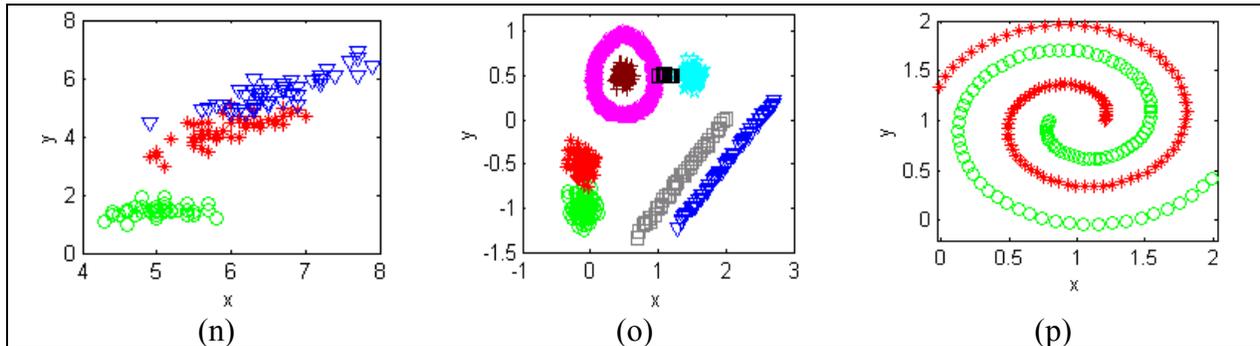


Figura 5.5: Conjunto de dados originais de n, o, p.

Figura 5.5.n: apresenta um conjunto de dados real *Iris*. Este conjunto é frequentemente utilizado em testes de algoritmos de agrupamento (Anderson, 1935). O conjunto possui as medidas, em centímetros, da altura e da largura das cepas e das pétalas, para 50 flores das espécies: *Setosa*, *Versicolor* e *Virginica*. Nesta figura apenas são apresentados os atributos mais representativos, isto é, o primeiro e o terceiro atributo. Esta base de dados procura mostrar a dificuldade de se agrupar classes que não são linearmente separáveis.

Figura 5.5.o: mostra um conjunto de dados similar ao exemplo de Jain (Jain *et al.*, 1999). Este conjunto de dados contém grupos que possuem os mesmos centros, além da sobreposição e de algumas formas quase elipsoidais. Com estas características avaliam-se os algoritmos perante formas arbitrárias no mesmo conjunto de pontos, a sensibilidade na inicialização dos centros, sensibilidade à ordem em que os dados são apresentados, sensibilidade à variabilidade no número de dados em cada grupo, e sensibilidade à sobreposição de grupos. Com este conjunto de dados é possível observar os comportamentos dos algoritmos com conjunto de dados com uma escala maior de pontos comparado com as mostradas anteriormente. Também é possível verificar o comportamento dos algoritmos perante um número maior de grupos a serem encontrados.

Figura 5.5.p: apresenta um conjunto de dados reproduzido de (Castro and Von Zuben, 2001). Este é um conjunto típico testado em algoritmos hierárquicos, em geral um algoritmo por partição terá maior dificuldade para se obter os mesmos resultados.

Seguem detalhes adicionais para cada conjunto de dados.

Tabela 5.1: Descrição dos conjuntos de dados.

Fig.	Nomes	n^o total de pontos	n^o de grupos²	n^o de pontos em cada grupo	Referência
a	tres_quadrados_distintos	75	3	25-25-25	Yager e Filev, (1994)
b	dois_grupos_distintos	75	2	50-25	Duda e Hart, (1973)
c	dois_ou_tres_grupos	67	3/2	30-7-30/34-33	Yager e Filev, (1994)
d	quatro_grupos_tam_diferentes	37	4	20-9-4-4	Liu e Xie, (1995)
e	dois_grupos_distantes	312	2	301-11	Kaymak e Setnes, (2000)
f	Borboleta	15	3/2	6-3-6/8-7	Klir, (1995)
g	quatro_grupos_definidos	160	4	40-40-40-40	Roubens, (1982); Windham, (1982)
h	quatro_grupos_mais_dispersos	160	4	40-40-40-40	Roubens, (1982); Windham, (1982)
i	quatro_grupos_sobrepostos	168	4/2	42-42-42-42 /84-84	Roubens, (1982); Windham, (1982)
j	Cruz	20	2	10-10	Gustafson e Kessel, (1979)
l	Cruz_Linha	148	3	50-50-48	Liu e Xie, (1995)
m	Círculo_Linha	111	2	60-71	Liu e Xie, (1995)
n	Íris_2D	150	3	50-50-50	Anderson, (1935)
o	Dados_Jain	834	8	314-100-100-100-100-10-53-57	Jain <i>et al.</i> , (1999)
p	Espiral	190	2	95-95	Castro e Von Zuben, (2001)

² Os conjuntos de dados com “ / “ são conjuntos que possuem grupos não bem definidos. Ambos os números de grupos poderão ser considerados corretos.

A seguir são apresentados os resultados dos algoritmos estudados neste trabalho utilizando-se os conjuntos de dados mostrados nesta seção.

5.4 Resultados Experimentais

Em todos os experimentos aqui apresentados foram adotados os seguintes parâmetros e condições iniciais:

- critério de parada $\varepsilon = 0.001$ ou máximo de iterações $l_{\max} = 8$;
- parâmetro $m = 2$;
- dados normalizados;
- os mesmos números c de centros quando necessários;
- a ordem de processamento dos dados;
- e as mesmas inicializações dos centros quando necessárias;

Para avaliar o desempenho segundo dois itens (sensibilidade à inicialização dos centros e sensibilidade à ordem de apresentação dos dados) de Bezdek e Pal (1992) os algoritmos foram testados com as seguintes condições iniciais para todos os algoritmos aqui utilizados:

- inicialização aleatória na apresentação dos dados e inicialização dos centros sempre fixa e determinada pelo usuário;
- inicialização aleatória dos centros e a inicialização na ordem de apresentação dos dados fixa;

Os parâmetros do algoritmo **B** foram determinados empiricamente, sendo adotados os seguintes valores: $\alpha = 0.01$; $\beta = 0.9$; τ varia conforme o conjunto de dados e será mostrado abaixo de cada resultado do algoritmo. O mesmo será feito para o algoritmo MFKM, quanto ao parâmetro que define o coeficiente de grade na base de dados, e o parâmetro α para determinar se o ponto está fora da célula ou não foi escolhido como 0.1. Para os algoritmos FCL e FFSECL o valor da taxa de aprendizado foi $\alpha = 0.01$, com decréscimo em cada iteração 0.99α .

Os resultados apresentados a seguir foram obtidos com a inicialização dos centros determinada pelo usuário e a entrada de dados não aleatória. Deste modo supõe-se obter o melhor resultado de cada algoritmo.

Os agrupamentos apresentados são obtidos com o maior grau de pertinência. As funções de pertinência obtidas pelo algoritmo AP-B, para os conjuntos de dados “a” a “n”, são reproduzidas no Apêndice II.

Legenda:

- FCM : *Fuzzy-C Means*;
- GK : *Gustafson Kessel algorithm*;
- FCL : *Fuzzy Competitive Learning*;
- FFSECL : *Fuzzy Frequency Sensitive Elliptical Competitive Learning*;
- MFKM : *Modified Fuzzy K-means*;
- AP : Algoritmo com Aprendizagem Participativa e matriz de covariância – AP-B;
- Centro : Representado por "●";
- η : fator de grade;
- τ : fator de compatibilidade.

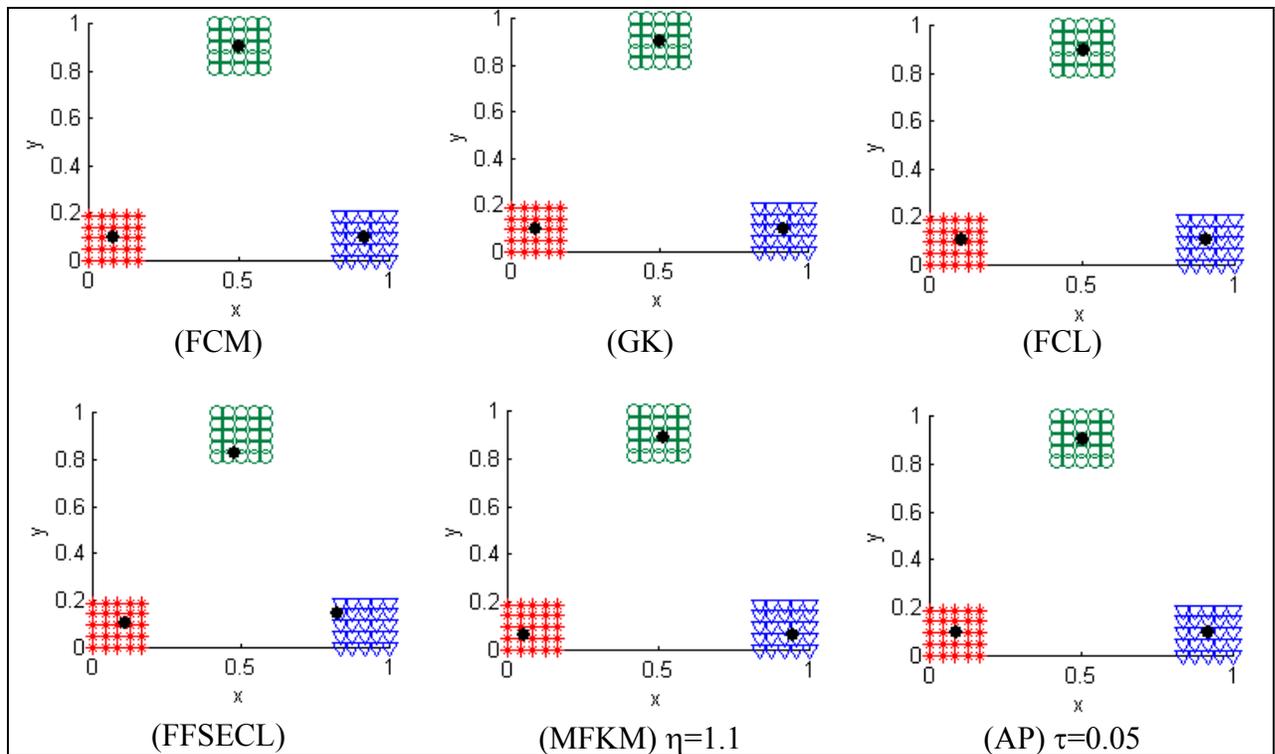


Figura 5.6: Agrupamento para o conjunto de dados da Figura 5.1.a.

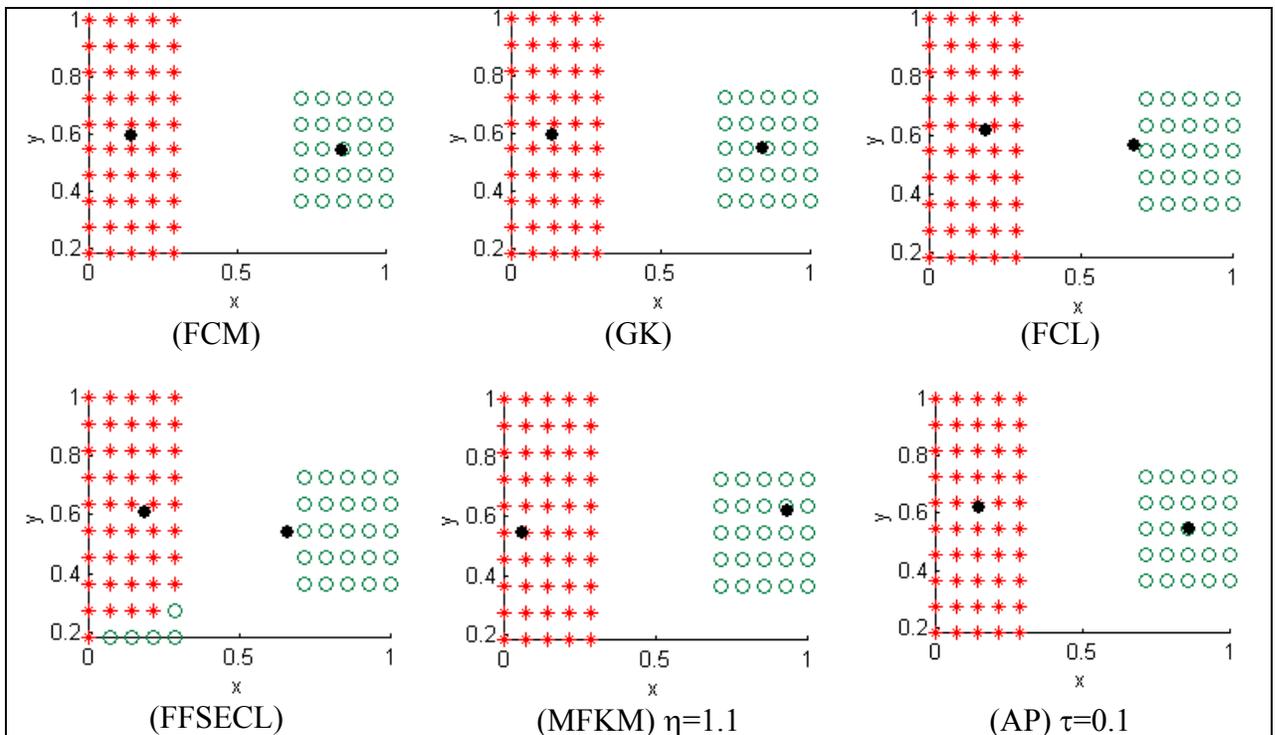


Figura 5.7: Agrupamento para o conjunto de dados da Figura 5.1.b.

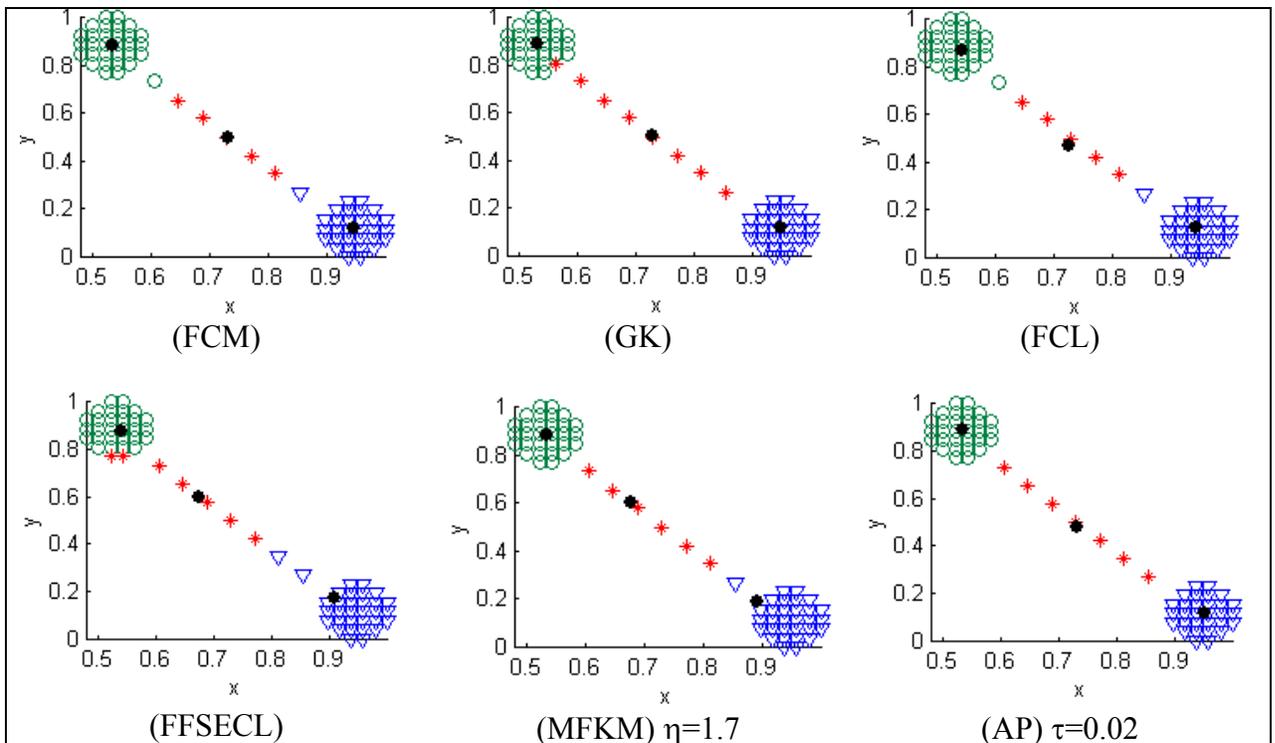


Figura 5.8: Agrupamento para o conjunto de dados da Figura 5.1.c.

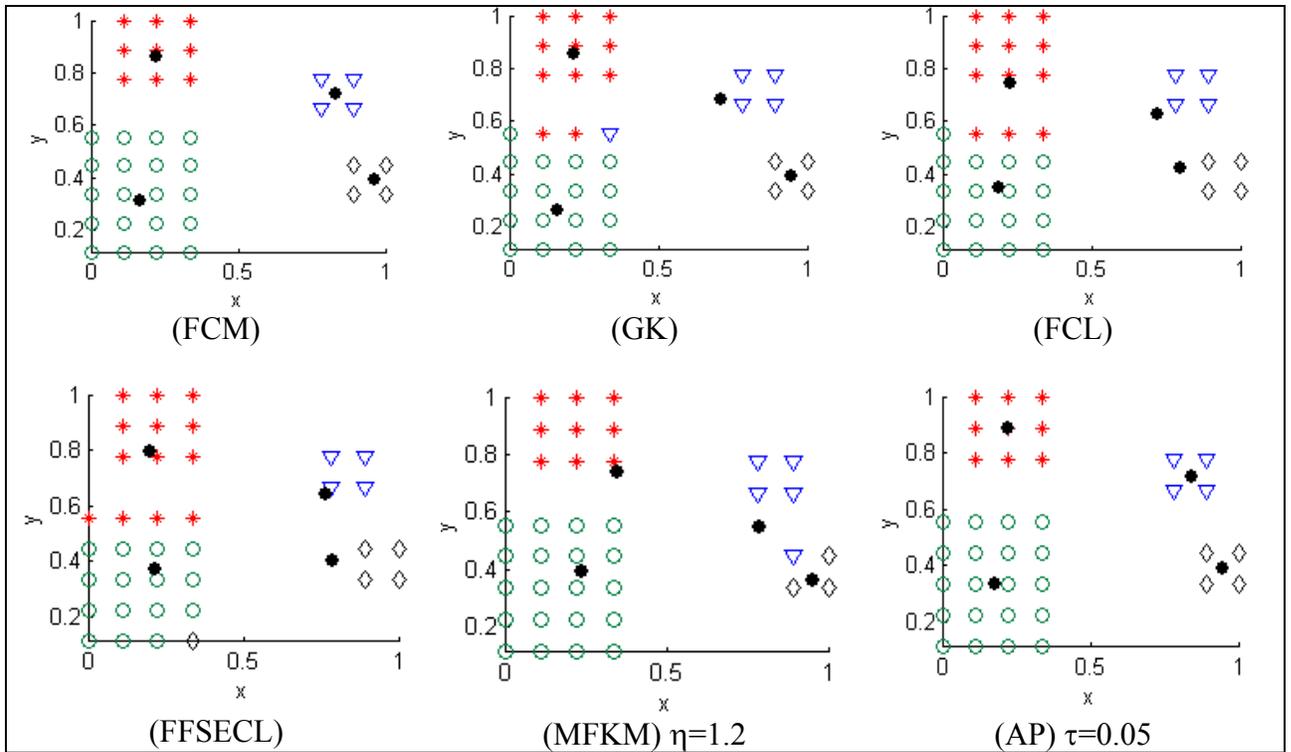


Figura 5.9: Agrupamento para o conjunto de dados da Figura 5.2.d.

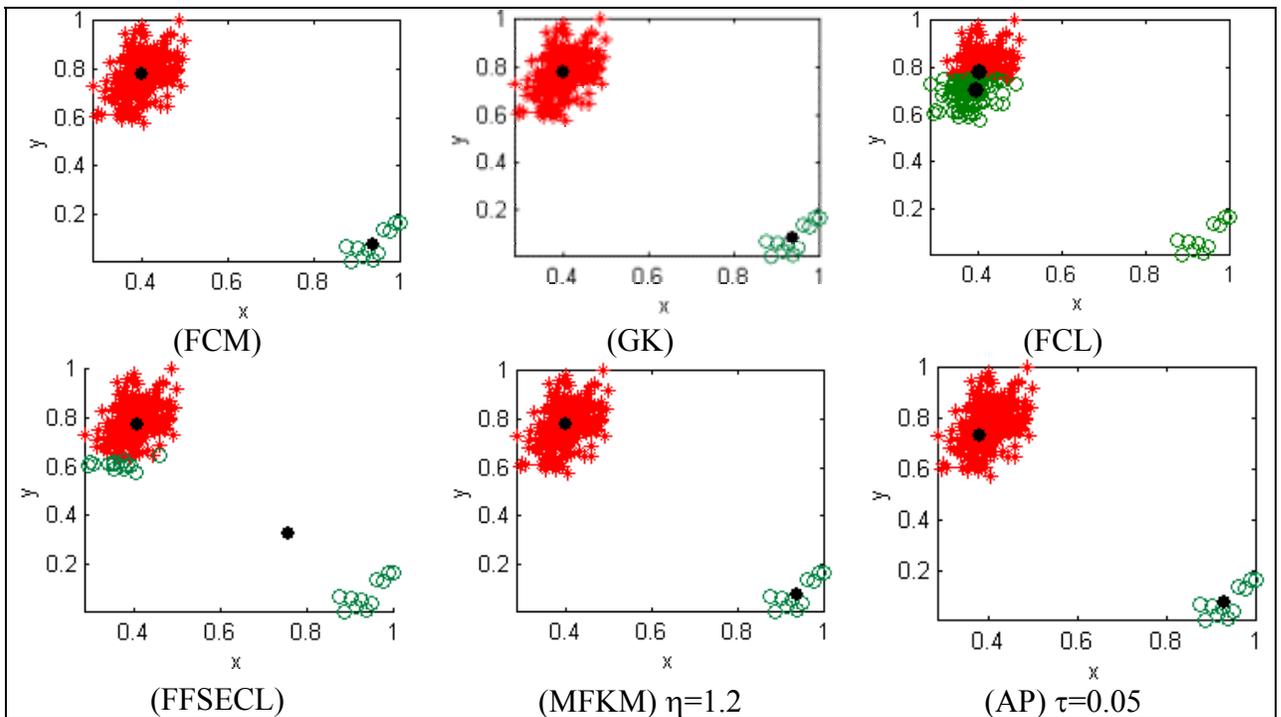


Figura 5.10: Agrupamento para o conjunto de dados da Figura 5.2.e.

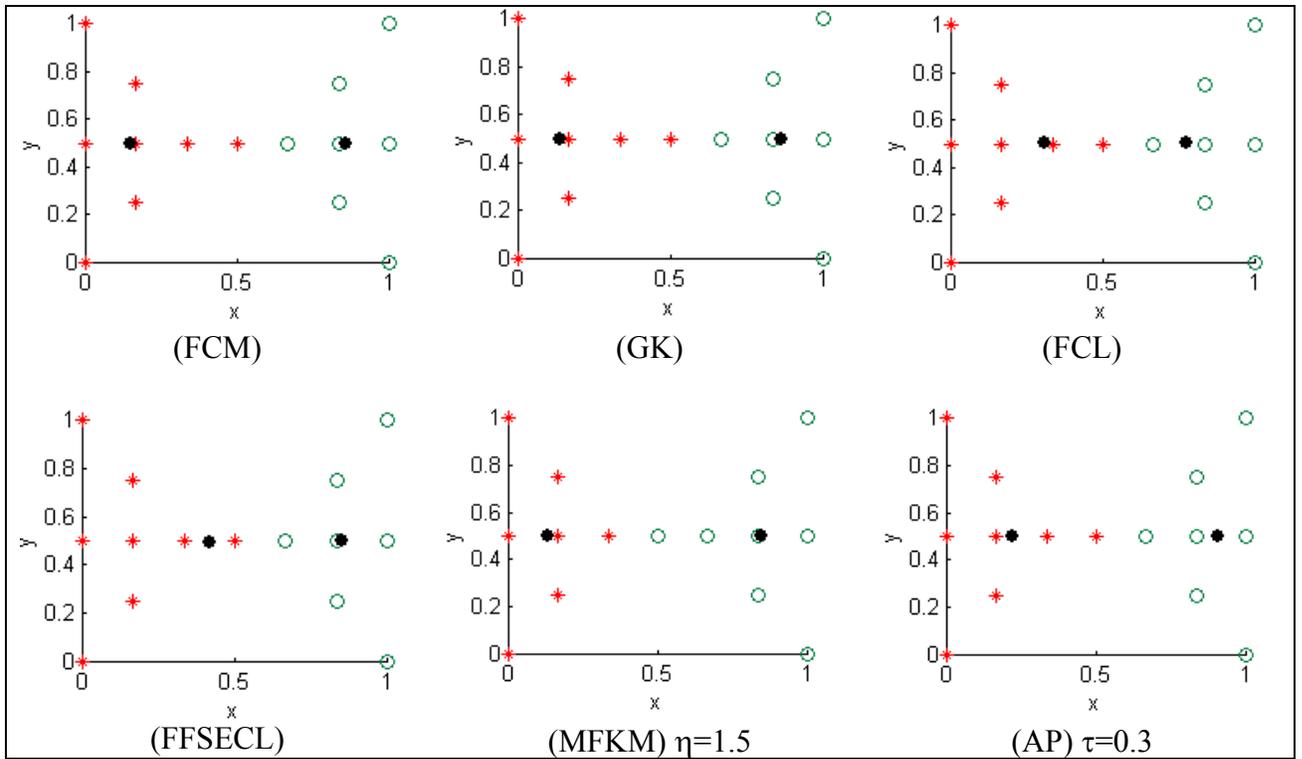


Figura 5.11: Agrupamento para o conjunto de dados da Figura 5.2.f.

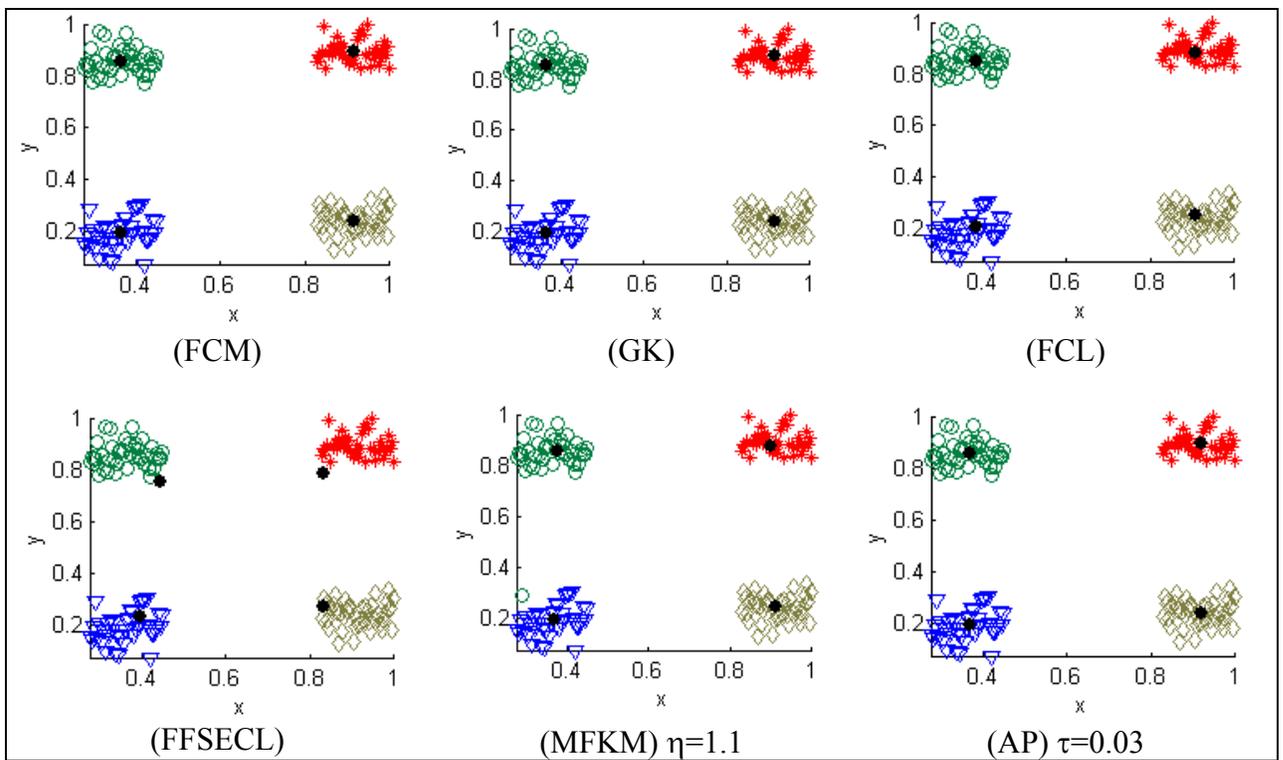


Figura 5.12: Agrupamento para o conjunto de dados da Figura 5.3.g

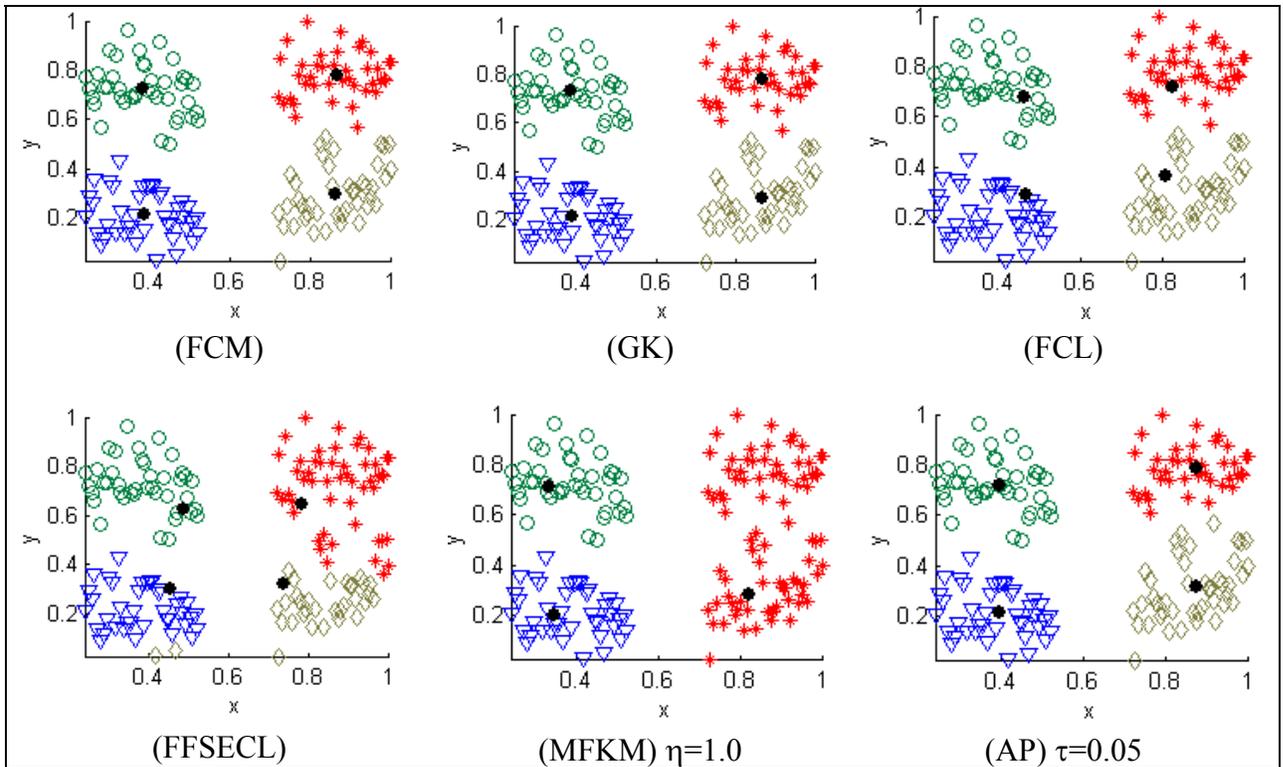


Figura 5.13: Agrupamento para o conjunto de dados da Figura 5.3.h.

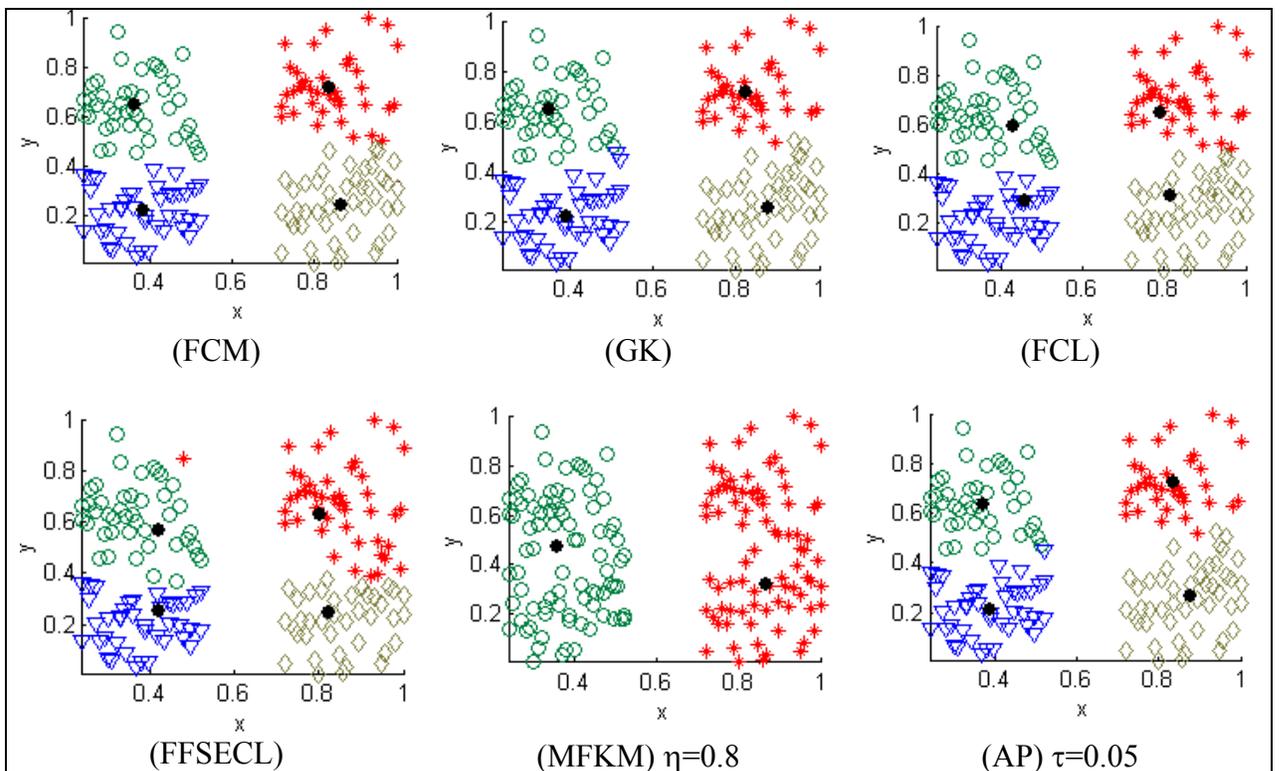


Figura 5.14: Agrupamento para o conjunto de dados da Figura 5.3.i.

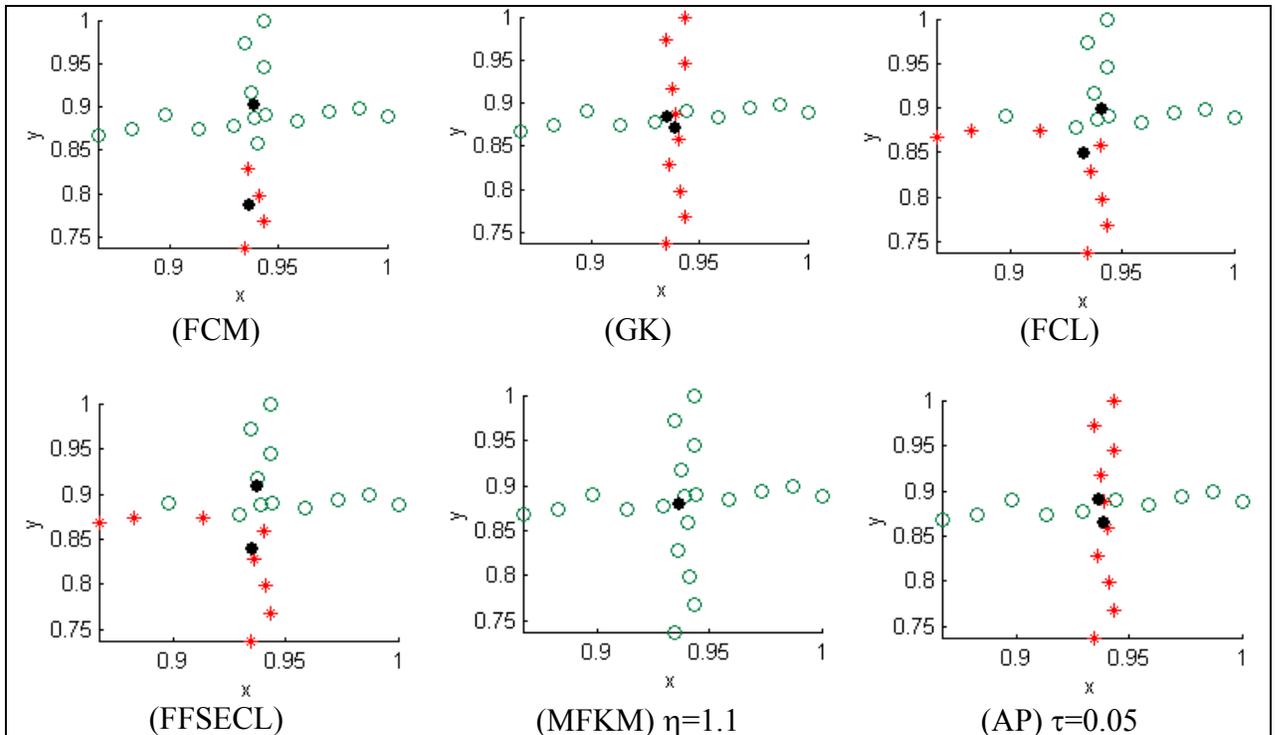


Figura 5.15: Agrupamento para o conjunto de dados da Figura 5.4.j.

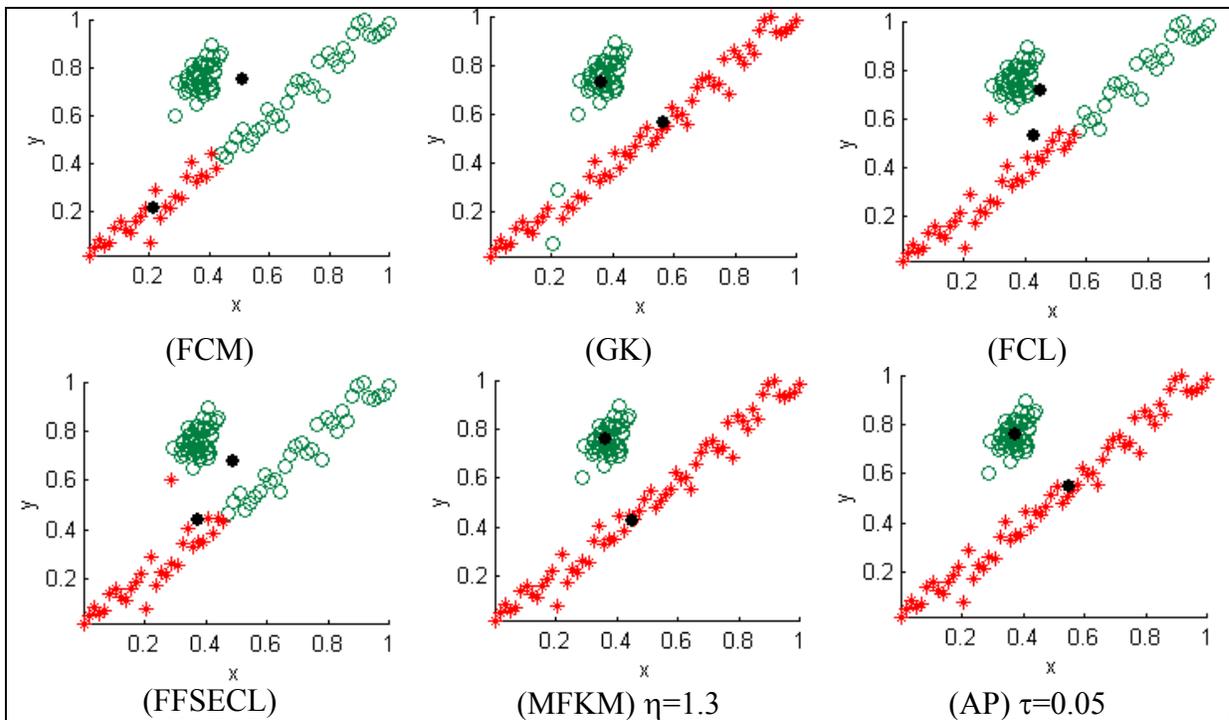


Figura 5.16: Agrupamento para o conjunto de dados da Figura 5.4.l.

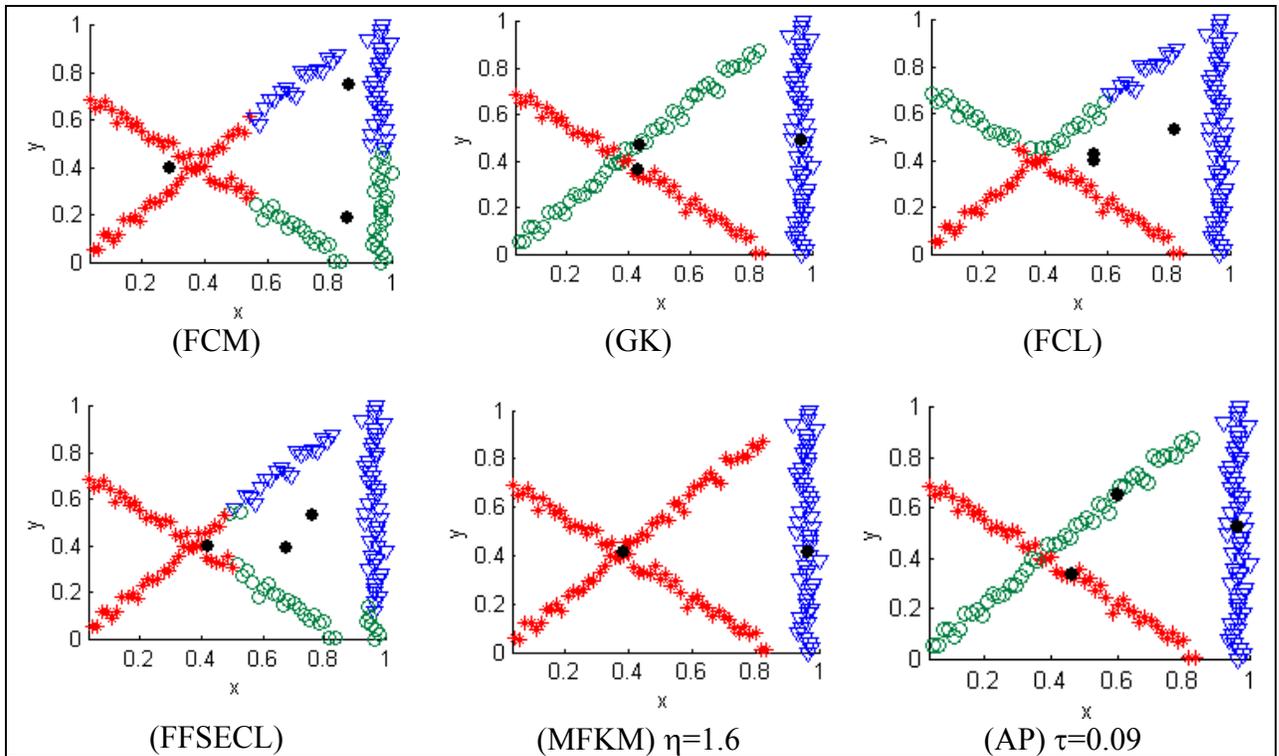


Figura 5.17: Agrupamento para o conjunto de dados da Figura 5.4.m.

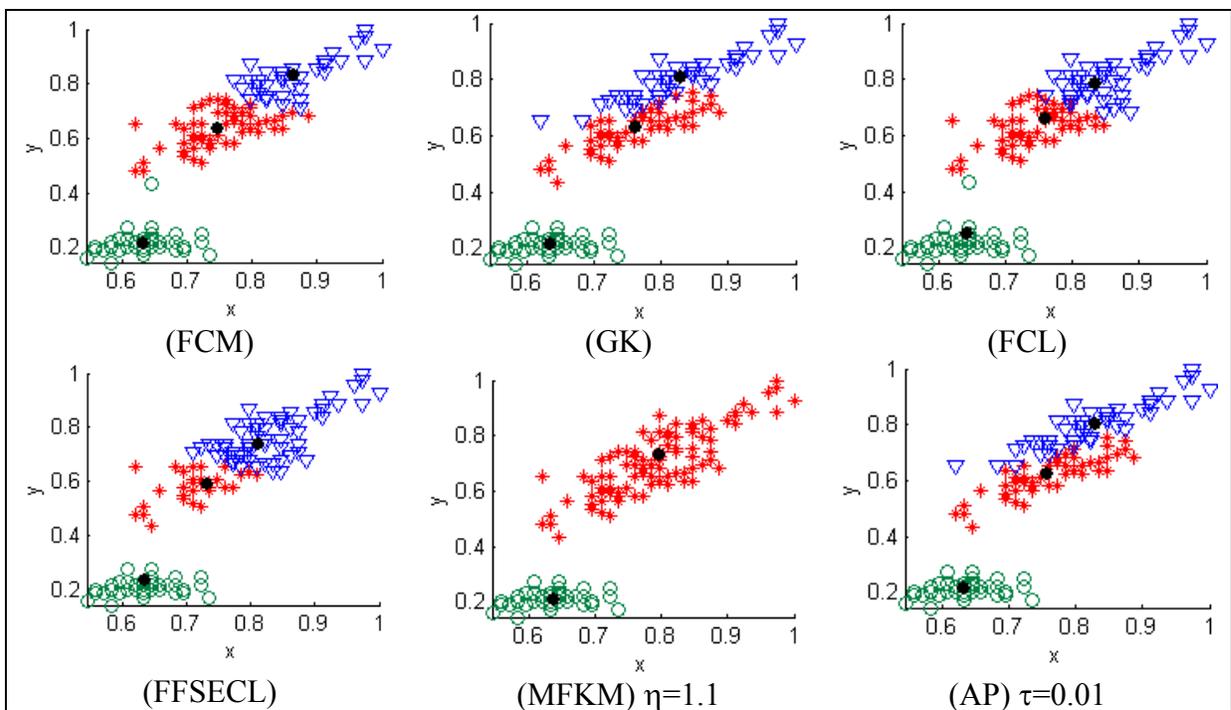


Figura 5.18: Agrupamento para o conjunto de dados da Figura 5.5.n.

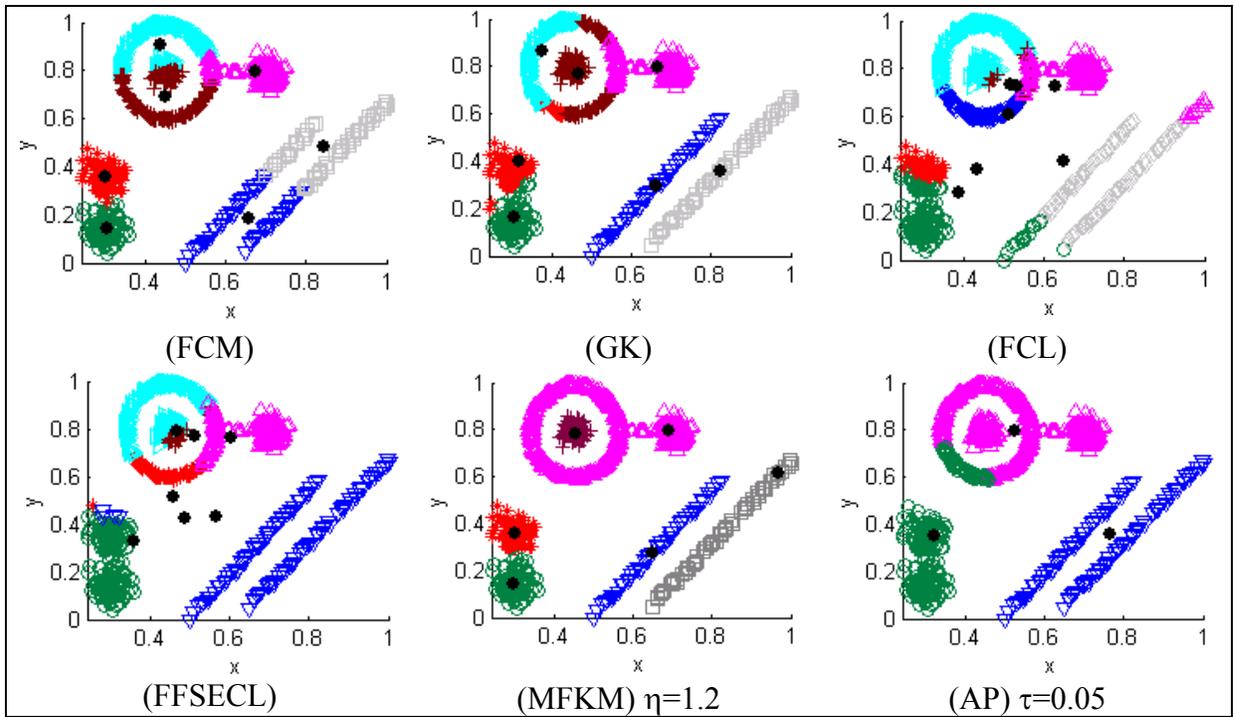


Figura 5.19: Agrupamento para o conjunto de dados da Figura 5.5.o.

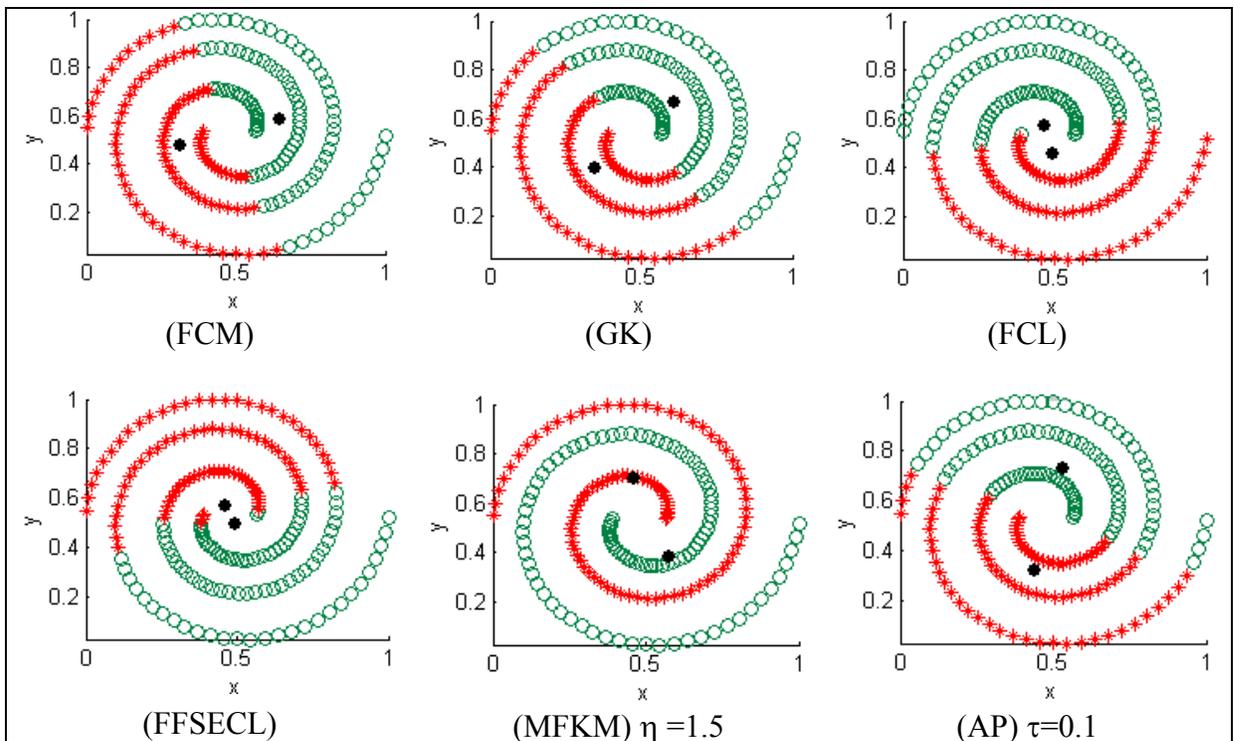


Figura 5.20: Agrupamento para o conjunto de dados da Figura 5.5.p.

Foi testada a capacidade dos índices (função) de validação em determinar o agrupamento mais apropriado em oito conjuntos de dados (a, d, e, f, g, i, n, m). Os valores dos indicadores destas funções são apresentados no Apêndice III onde são discutidos na Seção 5.6.

5.4.1 Tempo de Processamento dos Algoritmos

Os tempos de processamento da Tabela 5.2 foram obtidos pela média aritmética de cinco execuções de cada algoritmo, com a inicialização aleatória dos centros sendo as mesmas para os algoritmos que necessitam deste parâmetro. A apresentação dos dados foi aleatória em cada execução, sendo esta apresentação a mesma para todos os algoritmos. Os algoritmos foram implementados em Java, Windows NT e máquina Pentium 4, 256 Mb memória RAM.

Tabela 5.2: Tempo médio de processamento dos algoritmos.

Figura	Algoritmos - tempo médio em segundos					
	FCM	GK	FCL	FFESCL	MFKM	AP
a	0,03	6,30	0,09	4,92	18,03	18,77
b	0,01	5,20	0,04	2,54	1,81	21,05
c	0,04	10,33	0,07	4,55	24,81	27,07
d	0,01	1,35	0,15	0,36	0,73	1,98
e	0,02	21,93	0,01	0,90	268,35	34,15
f	0,10	55,93	0,30	18,16	43,17	110,10
g	0,13	95,34	0,33	18,10	207,36	148,39
h	0,18	185,43	0,30	18,43	126,98	194,76
i	0,01	0,49	0,02	0,59	1,10	1,00
j	0,00	0,11	0,00	0,06	0,46	0,10
l	0,02	10,00	10,23	0,02	13,15	25,77
m	0,37	185,36	0,40	11,47	150,49	201,40
n	0,02	10,46	0,01	0,84	7,00	6,35
o	0,36	1470,50	0,41	22,40	2719,75	1693,17
p	0,04	104,55	4,42	6,33	33,44	24,09
Média do tempo de processamento	0,0894	144,2187	1,119	7,31133	241,1087	167,21

5.4.2 Análise de complexidade dos algoritmos

As complexidades dos algoritmos FCM, GK, FCL, FFSECL, MFKM, AP-B são avaliadas em relação ao número de pontos em X , em relação ao número de grupos, e em relação à dimensão de X . Para comprovar a análise de complexidade calculou-se o tempo de processamento destes algoritmos.

Estes algoritmos foram implementados em Java, em ambiente Windows 2000, e máquina Athlon 1.4 GHz, 256 Mb memória RAM. Os tempos de processamento foram obtidos pela média aritmética de 30 execuções de cada algoritmo, com inicialização aleatória dos centros as mesmas para os algoritmos que necessitam deste parâmetro. A apresentação dos dados foi aleatória, sendo esta apresentação as mesmas para todos os algoritmos.

Os tempos de processamento dos algoritmos em relação ao número de pontos, foram obtidos com conjuntos de dados, onde o aumento foi somente no número de pontos, de modo que o número de grupos “desejáveis” permanece igual a dois grupos e a dimensão também igual a dois.

Estes resultados são discutidos na Seção 5.5.

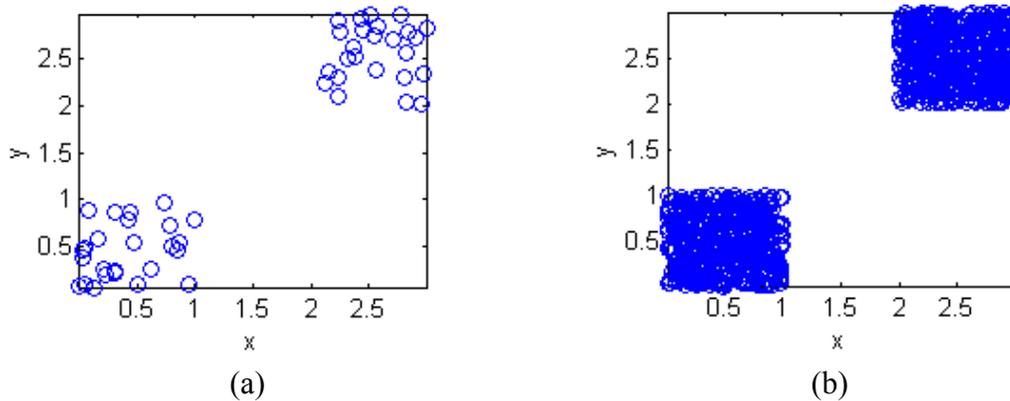


Figura 5.21: (a) Conjunto de dados original, com 50 pontos. (b) Conjunto de dados original, com 1000 pontos.

Tabela 5.3: Tempo de processamento dos algoritmos em relação ao número de pontos.

Tempo de processamento (em seg.) dos algoritmos em relação ao número de pontos ³						
nº. pontos	FCM	GK	FCL	FFSECL	MFKM	AP
50	0,00	0,48	0,00	0,05	0,59	1,05
100	0,00	1,09	0,00	0,04	1,27	2,88
160	0,00	3,38	0,01	0,07	3,56	5,67
240	0,00	7,03	0,01	0,10	9,30	10,90
300	0,00	10,51	0,01	0,35	14,37	18,99
500	0,00	41,75	0,01	0,35	63,35	34,83
800	0,00	86,84	0,01	0,41	245,15	96,67
1000	0,00	148,33	0,02	0,55	518,68	133,91

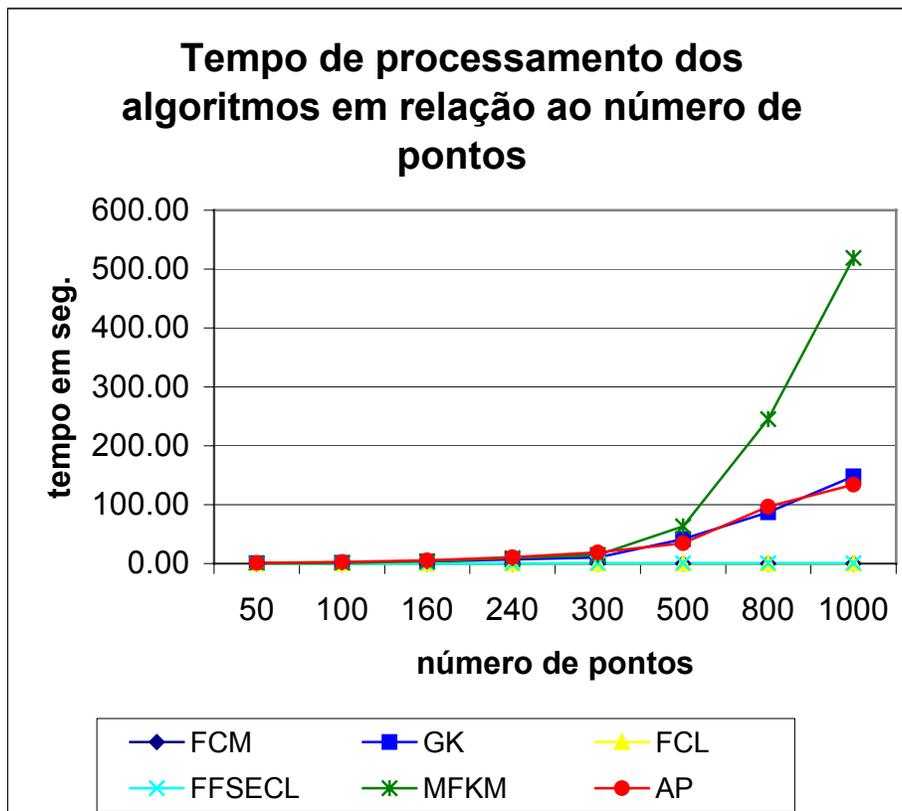


Figura 5.22: Tempo de processamento dos algoritmos em relação ao número de pontos.

³ Os tempos de processamento menores que 1 milissegundo não foram considerados aqui.

Os tempos de processamento dos algoritmos em relação ao número de grupos, foram obtidos com conjuntos de dados onde cada grupo possui somente 13 pontos, deste modo, o número de grupos “desejáveis” aumenta sem que a escala dos grupos aumente de modo significativo, e influencie no resultado. Sendo que, a dimensão permanece igual a dois.

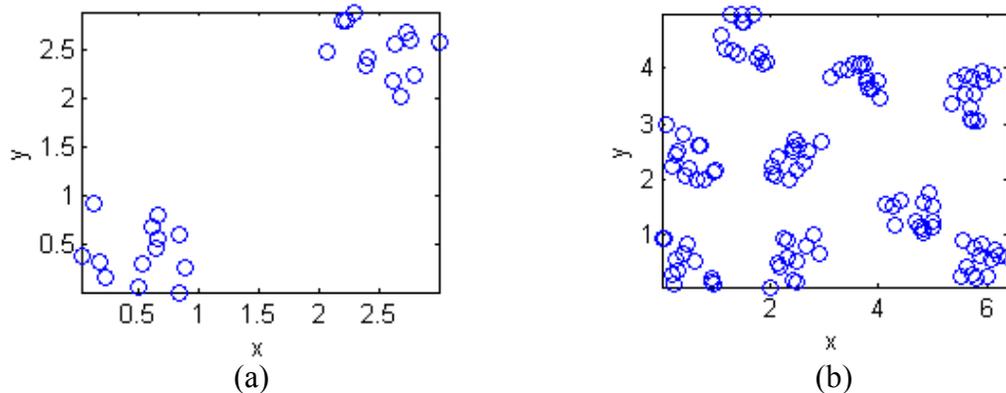


Figura 5.23: (a) Conjunto de dados original, com 2 grupos. (b) Conjunto de dados original, com 9 grupos.

Tabela 5.4: Tempo de processamento dos algoritmos em relação ao número de grupos.

n ^o grupos	Tempo de processamento (em seg) dos algoritmos em relação ao número de grupos ⁴					
	FCM	GK	FCL	FFSECL	MFKM	AP
2	0,00	0,15	0,00	0,04	0,31	0,24
3	0,00	0,53	0,01	0,11	0,42	1,38
4	0,01	2,09	0,01	0,34	0,68	3,42
5	0,01	5,13	0,01	0,68	1,70	6,89
6	0,02	9,16	0,01	1,15	3,00	12,70
7	0,02	18,39	0,03	1,82	6,60	27,35
8	0,04	30,91	0,03	2,42	9,57	39,39
9	0,05	47,95	0,04	3,31	13,49	64,51

⁴ Os tempos de processamento menores que 1 milissegundo não foram considerados aqui.

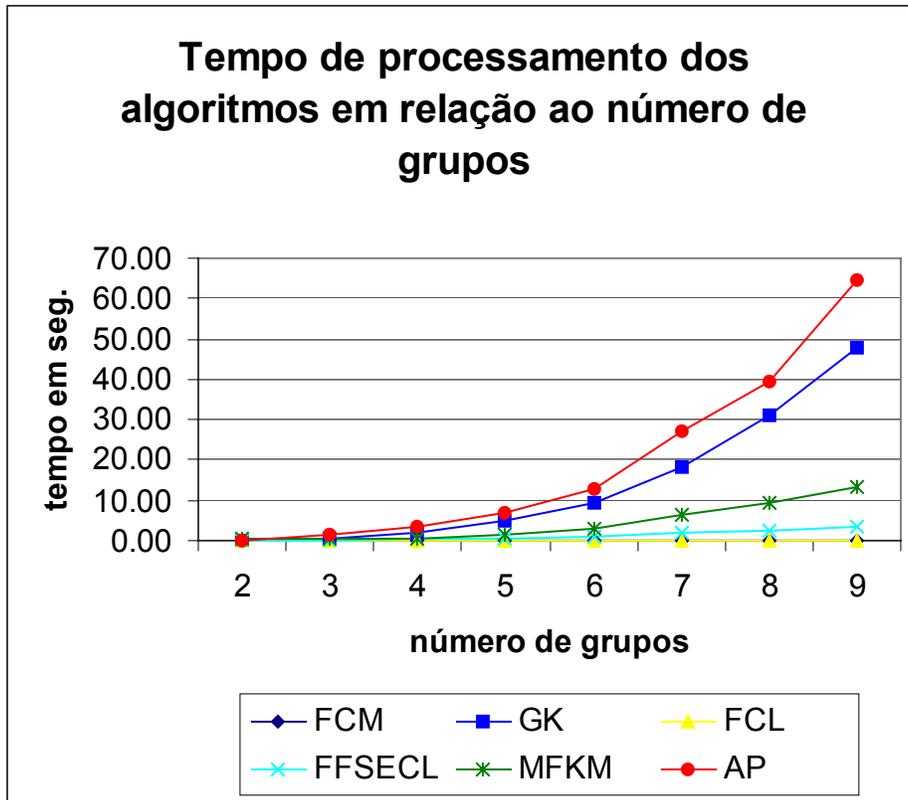


Figura 5.24: Tempo de processamento dos algoritmos em relação ao número de grupos.

Os tempos de processamento dos algoritmos em relação à dimensão de X , foram obtidos com conjuntos de dados, onde o aumento foi somente na dimensão de X , de modo que o número de grupos “desejáveis” permanece igual a dois grupos e o número de pontos permanece igual a 26.

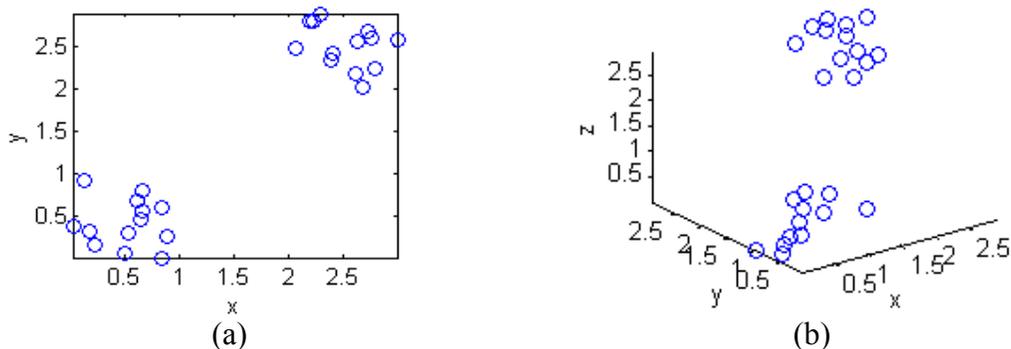


Figura 5.25: (a) Conjunto de dados original, com 2 dimensões. (b) Conjunto de dados original, com 3 dimensões.

Tabela 5.5: Tempo de processamento dos algoritmos em relação à dimensão.

dimensão	Tempo de processamento (em seg) dos algoritmos em relação à dimensão ^{5 e 6}				
	FCM	GK	FCL	FFSECL	AP
2	0,00	0,15	0,00	0,04	0,24
3	0,00	0,21	0,00	0,03	0,33
4	0,00	0,27	0,00	0,04	0,35
5	0,00	0,36	0,00	0,04	0,43
6	0,00	0,48	0,00	0,05	0,52
7	0,00	0,61	0,01	0,06	0,61
8	0,00	0,69	0,01	0,07	0,69
9	0,00	0,85	0,01	0,09	0,89

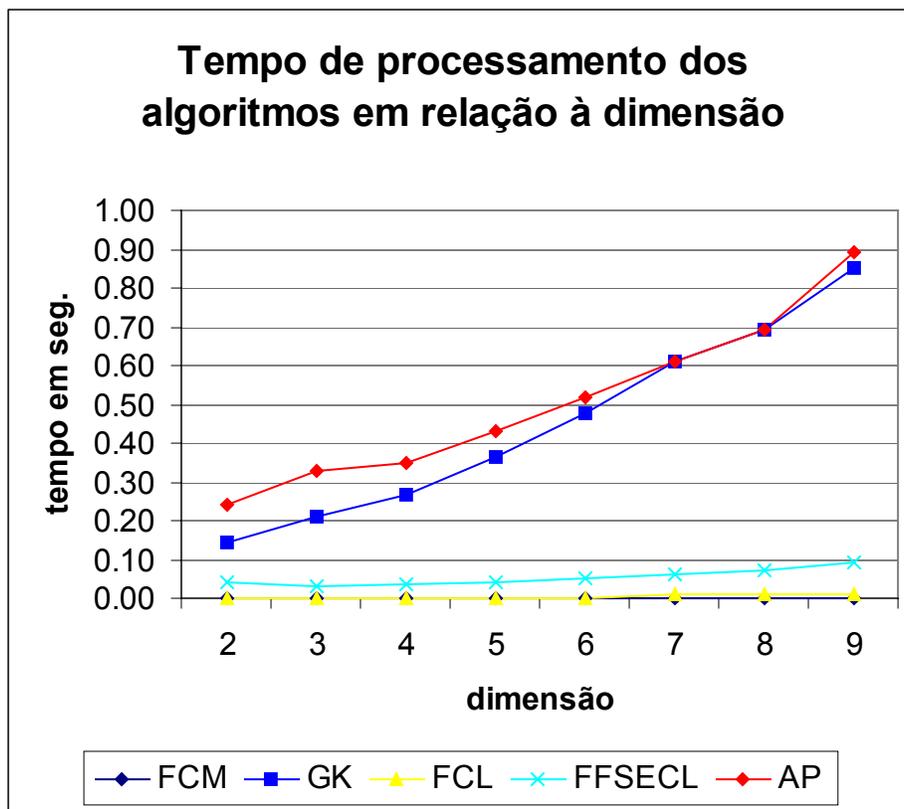


Figura 5.26: Tempo de processamento dos algoritmos em relação à dimensão.

⁵ Os tempos de processamento menores que 1 milissegundo não foram considerados aqui.

⁶ Os tempos de processamento para o algoritmo MFKM não são mostrados na Tabela 5.5, devido a inviabilidade de implementação deste algoritmo para conjuntos de dados com grandes dimensões.

5.5 Discussões

Nesta seção os algoritmos serão discutidos e avaliados tanto pelos requisitos de Han e Kamber (2001) como os critérios de análise de desempenho de Bezdek e Pal (1992). A Tabela 5.6 resume a avaliação para os algoritmos de agrupamento, aqui tratados, com os requisitos que foram considerados neste trabalho, juntamente com os critérios de análise. Esta avaliação foi baseada nos testes e resultados apresentados na seção anterior.

Tabela 5.6: Requisitos e critérios de análise de desempenho dos algoritmos.

Ponto	Requisitos e itens de avaliação.	Algoritmos de agrupamento					
		FCM	GK	FCL	FFSECL	MFKM	AP
(1)	Sensível à escalabilidade	N	S	N	N	S	S
(2)	Agrupar formas arbitrárias	N	S	N	N	S	S
(3)	Sensível aos parâmetros	S	S	S	S	+-	+-
(4)	Sensível à ordem de apresentação dos dados	S	N	S	S	N	+-
(5)	Sensível à dimensão	N	S	N	N	S	S
(6)	Legibilidade e usabilidade	S	S	S	S	S	S
(7)	Necessita escolher o número de grupos	S	S	S	S	N	N
(8)	Sensível à inicialização dos centros dos grupos	S	S	S	S	S	S
(9)	Sensível à densidade dos dados	S	S	S	S	S	+-
(10)	Sensível à variação de densidade entre os grupos	S	S	S	S	+-	+-
(11)	Sensível ao grau de separação entre os diversos grupos	S	N	S	S	S	N

Legenda:

N: Não; S: Sim; +/-: Mais ou menos.

- 1) Os algoritmos GK, MFKM e AP mostraram-se sensíveis à escalabilidade, conforme pode ser observado no tempo médio de processamento da Tabela 5.2 para o conjunto de dados da Figura 5.5.o (834 pontos). Estes algoritmos não mantiveram a média de tempo de processamento que estavam apresentando para os outros conjuntos de dados de escalas menores. Para o algoritmo GK o tempo de processamento aumenta devido aos cálculos com a matriz de covariância, como já citado na Seção 2.2.2. Além do aumento do número de dados, há também um aumento significativo no número desejado de grupos, fornecido pelo usuário. Esta variação de tempo pode ser confirmada observando o tempo obtido pelo algoritmo GK com o conjunto de dados da Figura 5.2.e, onde mesmo apresentando um aumento significativo de pontos no conjunto de dados (312 pontos), não se observou um aumento significativo no tempo de processamento. Isto ocorre porque o número de grupos desejado foi de apenas 2. Para o algoritmo MFKM além do aumento do tempo de processamento com o número de pontos do conjunto de dados, este tempo é ainda maior quando se quer um número maior de grupos. Isto ocorre porque é necessário aumentar o coeficiente de grade e a sensibilidade do algoritmo para detectar um número maior de grupos. No algoritmo proposto AP, o tempo de processamento aumenta não somente devido ao número de pontos de um determinado conjunto de dados, mas principalmente quando se deseja aumentar o número de grupos e conseqüentemente a sensibilidade de capturar um grupo (λ). Com o aumento desta sensibilidade há um aumento das operações de inserção e exclusão de centros, além do aumento das operações com a matriz de covariância devido à aplicação da distância de Mahalanobis. Esta sensibilidade na escala é melhor observada nas Figura 5.22 e Figura 5.24, que mostram o tempo de processamento em relação à escala do conjunto de dados e ao número de grupos, comprovando os resultados apresentados da Tabela 5.2.
- 2) Para os algoritmos de agrupamento de dados, acredita-se que depois de encontrar o número de grupos, a maior dificuldade é classificar os pontos em suas classes de modo o mais adequado possível. Isso está diretamente relacionado à forma em que estes grupos se encontram nos seus conjuntos de

dados. Os conjuntos de dados das Figura 5.1.a, ..., 5.5.p mostram a variedade e a dificuldade de se classificar cada ponto em sua classe, isto fica ainda mais evidente nos conjunto de dados Figura 5.5.o e Figura 5.5.p. Apesar do algoritmo MFKM conseguir encontrar o maior número de grupos com uma maior diversidade de formas no mesmo conjunto de dados, o algoritmo possui certas restrições principalmente quanto à sobreposição dos grupos, impedindo-o de realizar um agrupamento adequado nestes casos, como é possível observar nos resultados das Figuras 5.14, 5.15, 5.16 e 5.18. Para o algoritmo FFSECL não foi possível obter os resultados esperados, principalmente diante dos conjuntos de dados mostradas nas Figuras 5.4.j, 5.4.l e 5.4.m que possuem grupos de formas quase elipsoidais, o que se pode concluir é, que não foi possível a reprodução deste algoritmo com o material obtido.

- 3) A necessidade de parâmetros de entrada é um dos requisitos mais importantes para uma análise de algoritmos de agrupamento de dados. Quanto menor a sensibilidade aos parâmetros, melhor o algoritmo. O número de grupos ainda é o parâmetro mais crítico a ser determinado pelo usuário, como é o caso dos algoritmos FCM, GK, FCL e FFSECL. Para os algoritmos MFKM e AP, como é possível observar, seus parâmetros, η e τ respectivamente, variam conforme o conjunto de dados. Contudo seus resultados ainda não são totalmente dependentes quanto os algoritmos dependentes do parâmetro que fornece o número de grupos. Existem muitos estudos direcionados para a validação de grupos conforme citados na Seção 2.6, onde algumas funções auxiliam na procura de um agrupamento aceitável, e conseqüentemente no número de grupos ou em algum outro parâmetro solicitado pelo algoritmo. Contudo, assim como não se encontra um algoritmo que seja totalmente independente de seus parâmetros, ainda não há uma função de validação que consiga avaliar adequadamente todos os agrupamentos encontrados pelos algoritmos, como mostram os resultados que podem ser observados no Apêndice III. Portanto, um dos desafios que permanece é: tornar os algoritmos de agrupamento menos sensíveis aos parâmetros de entrada.

4) Além da dependência dos parâmetros determinados pelo usuário, o algoritmo pode sofrer influência nos resultados pela ordem de entrada dos dados X fornecida para o algoritmo. Os algoritmos FCM, FCL, FFSECL mostraram-se sensíveis à ordem de apresentação dos dados gerando resultados significativamente diferentes, como mostra a Figura 5.27. Apesar do algoritmo AP apresentar resultados diferentes, este algoritmo apresenta uma característica peculiar aos outros algoritmos. Ele apresenta resultados de agrupamento de dados mais adequados quando os dados de entradas são apresentados aleatórios, isto é, se apresentar todos os dados de um grupo e depois todos os dados de outro, o resultado poderá ser prejudicado.

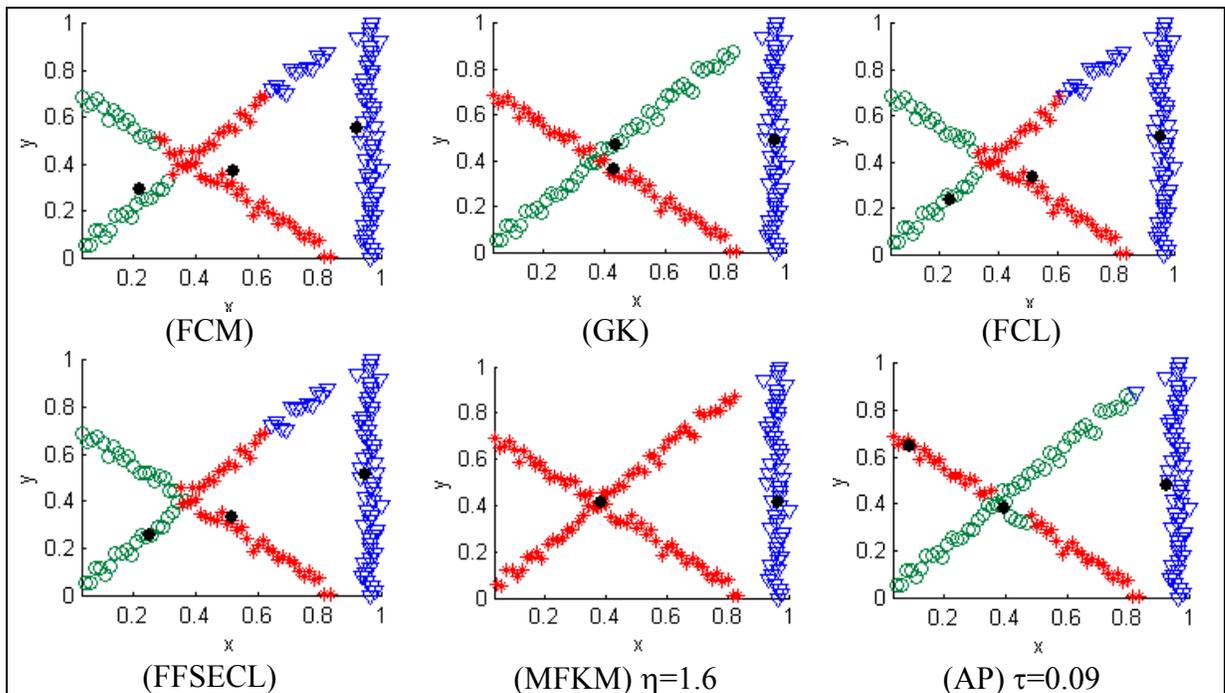


Figura 5.27: Algoritmos com sensibilidade à ordem de apresentação dos dados.

5) O algoritmo que apresentou maior sensibilidade ao número de atributos foi o algoritmo MFKM. O algoritmo MFKM forneceu uns dos melhores resultados perante grupos com formas arbitrárias como é mostrado nas Figuras 5.19 e 5.20, contudo este algoritmo não é indicado para conjunto de dados com altas dimensões. Esta restrição surge porque este algoritmo aplica os conceitos de

grade, mais especificamente porque o número de células vizinhas cresce exponencialmente com a dimensionalidade (3^p-1 , $p = \text{dimensão}$). Os algoritmos GK e AP são sensíveis ao número de atributos por utilizarem a distância de Mahalanobis, esta característica está mais detalhada na Seção 2.2.2, contudo, ainda não há o crescimento exponencial como observado no algoritmo MFKM. A Figura 5.28 apresenta os resultados dos algoritmos tratados neste trabalho, exceto para o algoritmo MFKM, para o agrupamento do conjunto de dados *Íris* com mais um de seus atributos. A sensibilidade do tempo de processamento em relação à dimensão é observada na Figura 5.26. Os resultados do MFKM não foram obtidos devido à grande complexidade de se implementar este algoritmo para conseguir agrupar dados acima de duas dimensões.

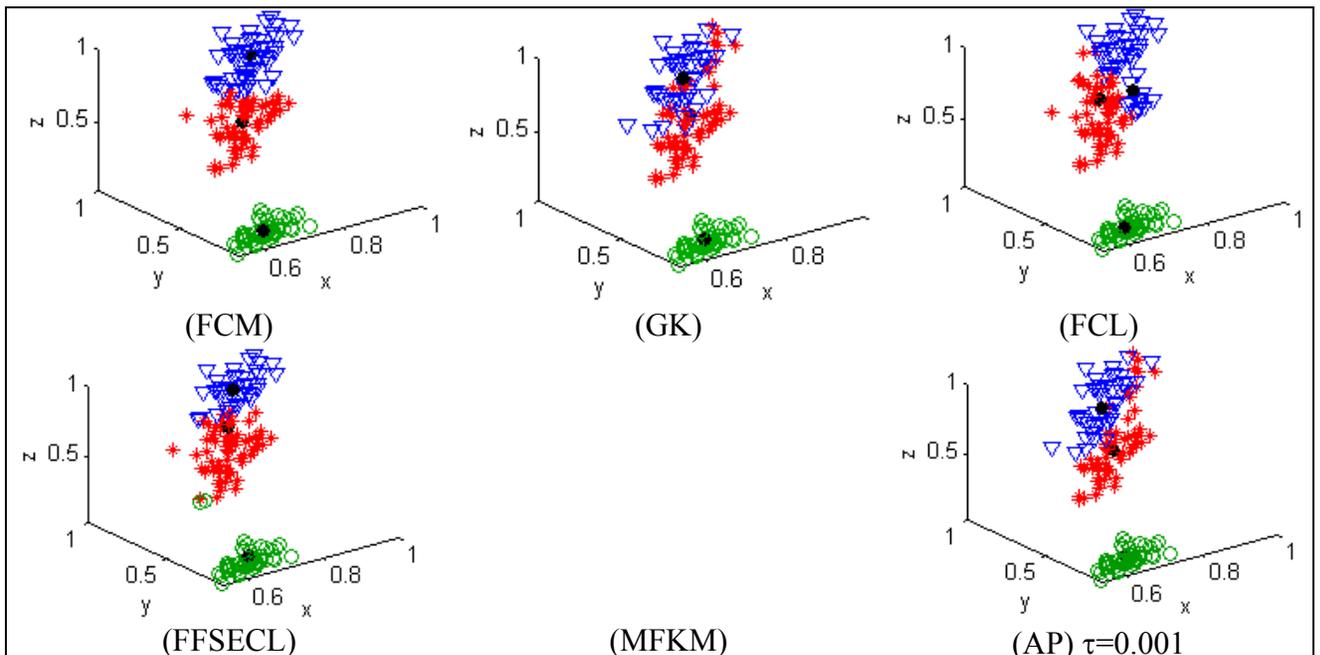


Figura 5.28: Algoritmos com sensibilidade ao número de atributos.

- 6) Os algoritmos apresentados neste trabalho mostram seus resultados em forma de matrizes (a matriz de pertinência e os centros dos respectivos grupos encontrados), desta maneira os resultados são facilmente compreensíveis e passíveis de serem tratados. Isto não acontece, por exemplo, em uma rede de Kohonen que se obtém como resultado um mapeamento dos dados, que por si

só não é possível obter um resultado de agrupamento de dados, há ainda uma necessidade de se "tratar" este mapeamento para só depois agrupar. Com a matriz de pertinência nebulosa consegue-se obter uma interpretação e aplicação semântica muito mais abrangente do que no caso de os mesmos resultados fossem rígidos, como é possível observar no Apêndice II para as funções de pertinências geradas pelo algoritmo AP com os conjuntos de dados de "a" até "n".

- 7) A escolha do número de grupos é para muitos estudiosos da área "um divisor de águas" para os algoritmos de agrupamento. O que se pode observar com os resultados das Figuras 5.6 à 5.20 é, o algoritmo AP apresentou resultados muito promissores quanto à determinação do número de grupos com as formas de grupos esférica e elipsoidais.

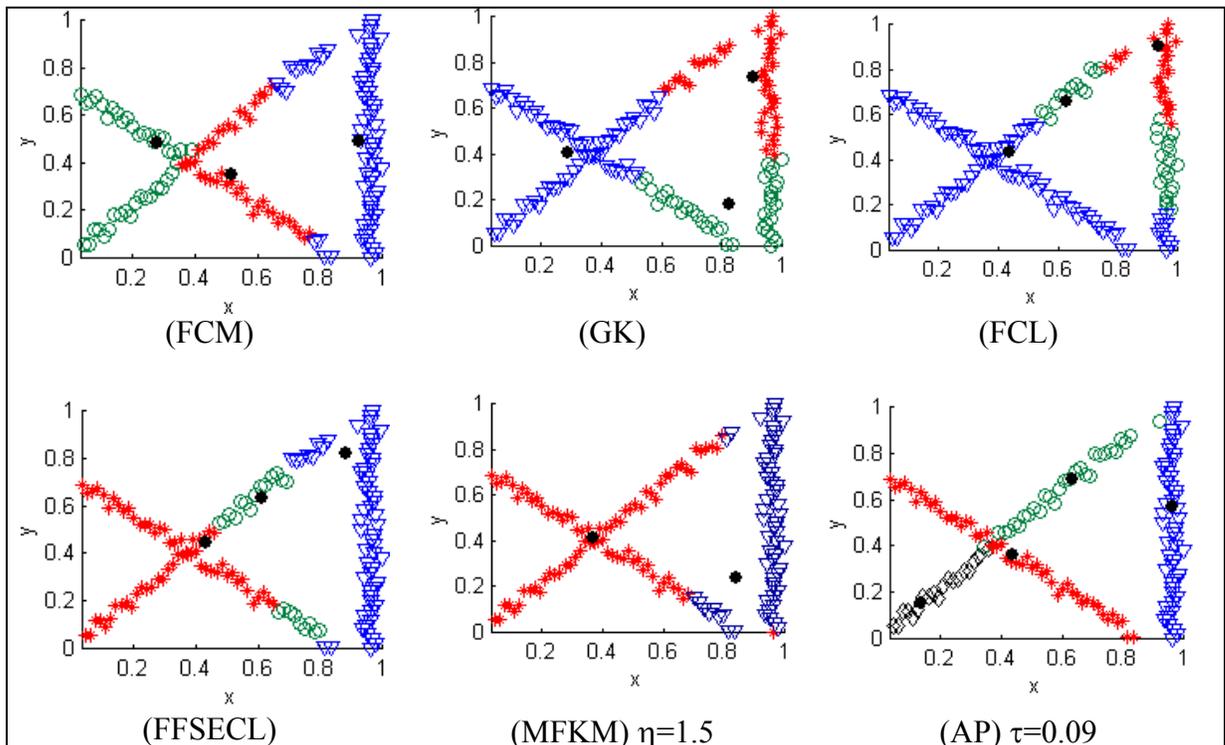


Figura 5.29: Algoritmos com sensibilidade à inicialização dos centros dos grupos.

- 8) A inicialização dos centros ou a inicialização da matriz de pertinência influenciam muito nos quatro primeiros algoritmos (FCM, GK, FCL e

FFSECL). Esta característica pode ser observada rodando os algoritmos dez vezes, todas com inicialização aleatória dos centros. Para os algoritmos AP e MFKM, a inicialização também foi aleatória com dois e um centro respectivamente. O que se pode observar que os algoritmos FCM, GK, FCL e FFSECL apresentaram resultados totalmente diversos dependendo da inicialização dos centros como pode ser observado na Figura 5.29. Para os casos do algoritmo MFKM e do algoritmo AP os seus resultados apesar de apresentarem resultados diferentes, não se alteraram de modo tão significativo e tão adverso como para os outros algoritmos.

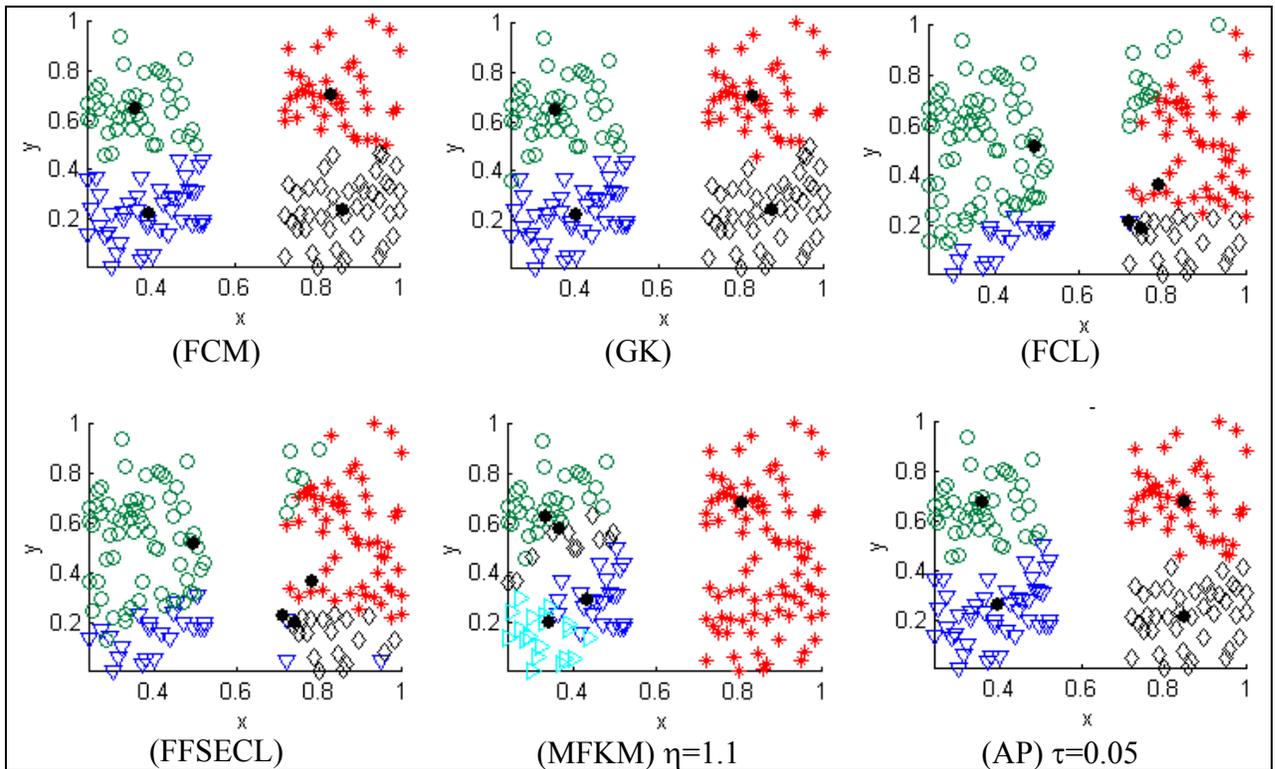


Figura 5.30: Algoritmos com sensibilidade à densidade dos dados.

- 9) A sensibilidade à compactação dos dados pelos algoritmos é observada principalmente nos resultados das Figuras 5.9, 5.10, 5.11, 5.12, 5.13, 5.14 e 5.18, eles mostram claramente a dificuldade dos algoritmos, principalmente dos FCL, FFSECL e MFKM, diante de classes que possuem pontos dispersos. Para o algoritmo MFKM esta sensibilidade é explicada por se utilizar o conceito de

densidade em seu algoritmo, como é possível observar nos resultados das Figuras 5.13 e 5.14, de modo que, ao se tentar aumentar a sensibilidade do algoritmo em detectar um número maior de grupos, os resultados obtidos não foram satisfatórios, como é mostrado na Figura 5.30. Não foram observadas grandes variações nos resultados para os algoritmos FCM, GK e AP. Os resultados da Figura 5.30 abaixo foram testados com inicialização aleatória dos centros e apresentação aleatória dos dados.

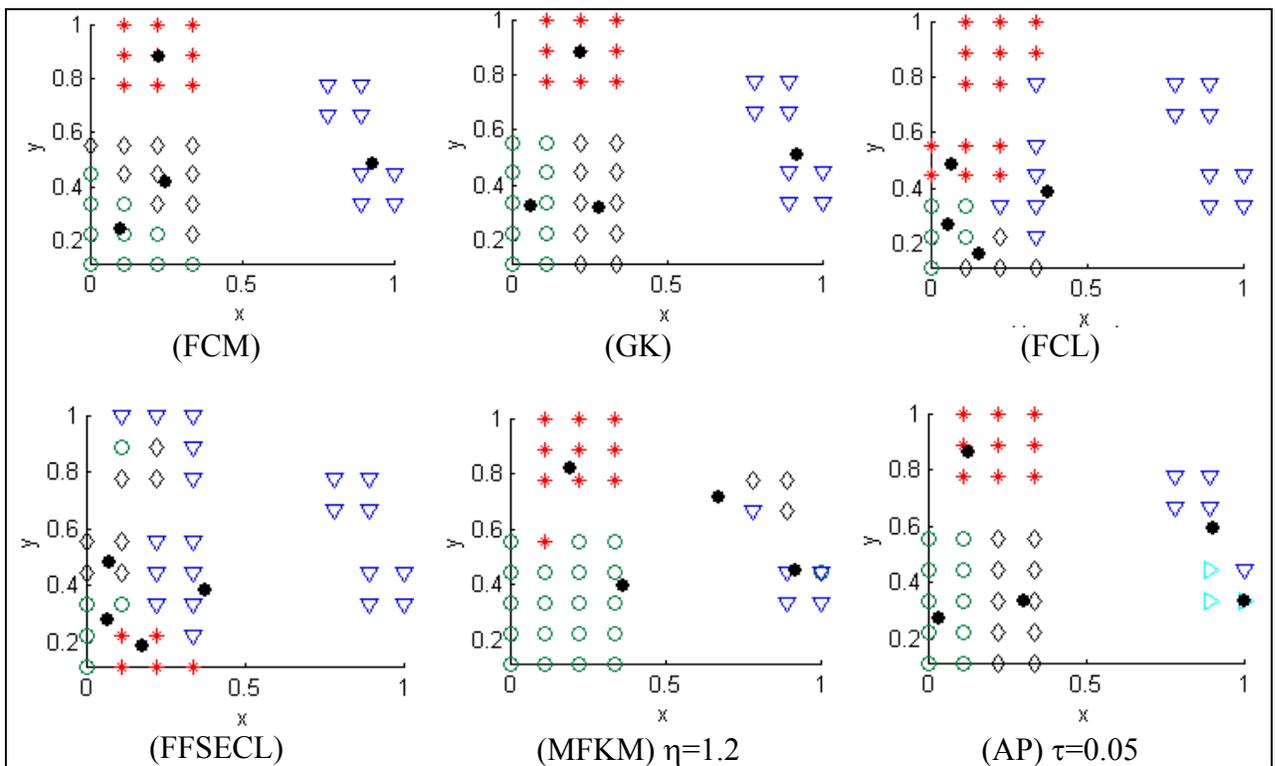


Figura 5.31: Algoritmos com sensibilidade à variação da densidade entre os grupos.

10) A sensibilidade à variação do número de dados em cada grupo dentro de um mesmo conjunto de dados é observada principalmente nos resultados das Figuras 5.9 e 5.10. Para o conjunto de dados da Figura 5.9 os algoritmos FCM, GK, FCL e FFSECL não conseguem encontrar os seus quatro grupos quando não apresentados os centros iniciais já próximos de seus respectivos grupos. O mesmo foi observado para o conjunto de dados da Figura 5.10, contudo o algoritmo FCL e o FFSECL não foram capazes de encontrar os grupos

existentes, independentemente dos parâmetros de entrada. Este comportamento de instabilidade não foi observado para outros conjuntos de dados que apresentavam grupos com um número de dados representativos como por exemplo nas Figuras 5.6 e 5.7. Os resultados da Figura 5.31 foram testados com inicialização aleatória dos centros e apresentação aleatória dos dados.

- 11) O grau de separação entre os grupos ou a sobreposição destes pode influenciar radicalmente nos resultados de um algoritmo. São os casos dos conjuntos de dados das Figuras 5.3.h, 5.3.i, 5.4.j, 5.4.m, 5.5.n, o, p, onde todos estes conjuntos possuem grupos que não são linearmente separáveis. Os algoritmos FCM, FCL, FFSECL são dependentes desta propriedade para que consiga obter um "bom" agrupamento. Para o caso do algoritmo MFKM esta dependência é indireta, porque ele consegue agrupar classes que não são linearmente separáveis, como é o caso do conjunto de dados da Figura 5.5.p. Contudo, a restrição é a sobreposição de dados, isto é, pontos que podem pertencer para dois grupos ao mesmo tempo, que o algoritmo MFKM não é capaz de agrupar, como é o exemplo dos resultados da Figura 5.4.j e e 5.4.m. Os algoritmos GK e AP apresentam resultados satisfatórios para os casos 5.3.h, 5.3.i, 5.4.j, 5.4.m e 5.5.n.. Contudo, seus critérios de medidas de distâncias são incapazes de determinar todos os grupos do conjunto de dados da Figura 5.5.o e incapaz de encontrar os grupos existentes no conjunto de dados da Figura 5.5.p.

5.6 Discussões sobre os Resultados das Funções de Validação

Esta seção apresenta uma discussão sobre os resultados dos testes, detalhados no Apêndice III, realizados com as funções de validação e os algoritmos FCM, GK, FCL, FFSECL, MFKM e AP-B utilizando os conjuntos de dados “a, e, g, i, f, d, n, l”.

A literatura recomenda a aplicação de funções de validação para se obter indicadores de qualidade para os agrupamentos realizados pelos algoritmos. Contudo, é possível observar, no Apêndice III, que não houve uma função de validação que

forneceu um resultado satisfatório para todas as formas e número de grupos presentes nos conjuntos de dados considerados.

Portanto, observa-se que não há uma função de validação que consiga determinar o número desejado de grupos e a natureza do agrupamento satisfatoriamente em qualquer circunstância. Esta dificuldade de encontrar uma função de validação confirma as discussões existentes na literatura, mostrando que permanece em aberto uma conclusão definitiva sobre este assunto.

Também é possível concluir que há a necessidade de se encontrar uma função de validação adequada para o algoritmo de agrupamento com aprendizagem participativa, para que, deste modo, seja possível encontrar automaticamente um valor adequado para o parâmetro τ .

5.7 Análise do Algoritmo AP-B

Esta seção mostra uma análise do algoritmo AP-B proposto neste trabalho apresentando um resumo com suas vantagens e desvantagens de acordo com os resultados e discussões realizadas nas seções anteriores .

Analisando o algoritmo AP-B em relação ao MFKM, ambos são algoritmos de aprendizagem não supervisionada, isto é, além de realizarem o agrupamento de dados, eles determinam o número de grupos. Contudo, o algoritmo AP-B não realiza o agrupamento de formas muito complexas como as mostradas nas Figuras 5.5.o. e 5.5.p., casos em que o algoritmo MFKM tem sucesso. Por outro lado, o algoritmo MFKM não é capaz de encontrar grupos com pontos sobrepostos como aqueles das Figuras 5.4.m. e 5.4.j, casos que são resolvidos pelo algoritmo AP-B.

O algoritmo AP-B, apresenta um tempo de processamento maior em relação ao número de grupos quando comparado com o algoritmo MFKM, como mostra a Figura 5.24. Contudo, o algoritmo AP-B utiliza um tempo de processamento menor em relação ao número de pontos. Como o número de grupos nunca será maior que o número de pontos e normalmente o número de grupos é bem menor que o número de pontos, isto é, $c \ll n$, o aumento no número de pontos tem um impacto maior quanto ao tempo de processamento no algoritmo MFKM que no algoritmo AP-B. A principal desvantagem do

algoritmo MFKM com relação ao algoritmo AP-**B** proposto, é a sua complexidade de $O(3^p)$, tornando o algoritmo MFKM inviável para conjuntos de dados de grande dimensão.

Os resultados dos algoritmos AP-**B** e GK com os conjuntos de dados das Figuras “a” a “p” são muito semelhantes, porque utilizam a mesma medida de similaridade. Contudo, vale salientar que o algoritmo AP-**B** determina o número de grupos além de realizar o agrupamento dos dados, enquanto o algoritmo GK é um algoritmo de agrupamento supervisionado, isto é, necessita conhecer *a priori* o número de grupos.

Os algoritmos FCL e FFSECL utilizam a aprendizagem competitiva e são similares aos algoritmos AP’s. Contudo, os algoritmos FCL e FFSECL não realizam a atualização apenas do centro vencedor. Além disso os algoritmos FCL e FFSECL são algoritmos de agrupamento supervisionados.

O desempenho do algoritmo AP-**B** sob o ponto de vista de agrupamento, não é independente dos seus parâmetros, principalmente com o parâmetro que determina o limiar de incompatibilidade (τ). Contudo, não há algoritmo que seja totalmente independente de seus parâmetros, assim como não há uma função de validação que determine a melhor partição para todos os tipos de conjunto de dados.

O algoritmo AP-**B** mostrou ser uma alternativa para o agrupamento de conjuntos de dados onde se desconhece o número de grupos e as formas de seus grupos, pois nestas circunstâncias ele apresentou um compromisso aceitável entre eficiência computacional e agrupamento.

5.8 Resumo

Neste capítulo, foram apresentados os resultados de agrupamento fornecidos pelos algoritmos considerados neste trabalho utilizando os conjuntos de dados sintéticos e reais com diferentes graus de dificuldade de agrupamento. Estes resultados foram analisados e discutidos segundo os requisitos e critérios de desempenho conforme sugerido pela literatura na área. O algoritmo AP-**B** apresentou um melhor desempenho em oito dos onze requisitos e critérios estudados em comparação com outros algoritmos aqui apresentados. Além disso o seu desempenho computacional e resultados de agrupamento

se mostram competitivos com os algoritmos clássicos de agrupamento, tornando-o assim uma alternativa para resolver o problema de agrupamento que não se conhece o número de grupos e as suas formas *a priori*.

6. Conclusões

De acordo com os resultados da seção anterior, observa-se que o algoritmo derivado do paradigma de aprendizagem participativa mostrou ser uma alternativa satisfatória para agrupamento nebuloso, comparando-se com os algoritmos FCM, GK, FCL, FFSECL e MFKM. O algoritmo AP obteve um desempenho aceitável e promissor, determinando o número de classes e os seus respectivos centros e obtendo uma matriz de pertinência com sucesso para uma grande parte dos conjuntos de dados considerados neste trabalho.

Para trabalhos futuros, ficam os seguintes casos a serem testados e/ou aplicados:

1. O problema de se ajustar o parâmetro τ poderá ser analisado com o auxílio de funções de validação apresentadas na Seção 2.6, pois estas funções podem indicar um valor aceitável para o número de classes e, portanto, o valor adequado para τ . Alternativamente, é possível utilizar algoritmos genéticos para se determinar um valor aceitável para o parâmetro τ . Contudo, em ambos os casos o resultado dependerá da escolha da função de validação correta. Como pode ser observado nos resultados do apêndice III, esta questão merece aprofundamento e ainda permanece em aberto na literatura.
2. O algoritmo AP deverá ser testado para diferentes tipos de dados, como binário por exemplo.
3. Também poderá ser testado o caso de ruído nos dados, devido a sua grande aplicação a situações onde há erros e informação incompleta nos conjuntos de dados.
4. Ampliar o algoritmo de modo que seja possível agrupar satisfazendo restrições.
5. E, finalmente, determinar que funções de validação são as mais apropriadas para o algoritmo AP.

Referências Bibliográficas

1. AL-SULTAN, K. S., KHAN, M.. *Computational experience on four algorithms for the hard clustering problem*. Pattern Recognition Letters. 3 ed, v. 17, p. 295-308, mar. 1996.
2. ANDERSON, E.. *The irises of the gaspé peninsula*. Bulletin of the American Iris Society n.9, p. 2-5, 1935.
3. ARAÚJO, E. de O.. *A heuristic adjustment to the calculation of the dissimilarity in the fcm algorithm*. In: Proc. 9th IFSA World Congress and 20th NAFIPS International Conference, jul. 25-28, Vancouver, Canada, v. 1, 2001. p. 25-30.
4. BACKER, S. de, SCHEUNDERS, P.. *A competitive elliptical clustering algorithm*. Pattern Recognition Letters. 11-13 ed, v. 20, p. 1141-1147, nov. 1999a.
5. BACKER, S. de, SCHEUNDERS, P.. *Texture segmentation by frequency-sensitive elliptical competitive learning*. In: Proc. ICIAP99/, International Conference on Image Analysis and Processing, Venesa, p. 64-69, 1999b.
6. BACKER, S. de. *Unsupervised pattern recognition: dimensionality reduction and classification*. University of Antwerp, dissertation. 2002. 138 p., p. 72-90. WWW [http: 143.129.203.3/visielab/theses/debacker/SteveThesis.pdf](http://143.129.203.3/visielab/theses/debacker/SteveThesis.pdf) .
7. BALL, G. H., HALL D. J.. *A novel method of data analysis and pattern classification*. Menlo Park, CA: Stanford Research Institute. Technical Report. 1965.
8. BARALDI A., BLONDA P.. *A survey of fuzzy clustering algorithms for pattern recognition*. International Computer Science Institute. Berkeley, California, TR-98-038. out. 1998.
9. BEZDEK, J. C., PAL, S. K.. *Fuzzy models for pattern recognition: methods that search for structures in data*. New York: IEEE, 1992.
10. BEZDEK, J. C., PAL, Nikhil R.. *Some new indexes of cluster validity*. IEEE – Transactions on Systems, Man, and Cybernetics. part B: Cybernetics. v. 28, n. 3, p. 301-315, jun. 1998.
11. BEZDEK, J.C.. *Cluster validity with fuzzy sets*. Journal of Cybernetics. v. 3, n. 3, p. 58-72, 1974.
12. BEZDEK, J.C.. *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum Press, 1981.

13. CASTRO, L. N. de, VON ZUBEN, F. J.. *aiNet: an artificial immune network for data analysis*. Book Chapter in *Data Mining: A Heuristic Approach*, H. A. Abbass, R. A. Sarker, and C. S. Newton (eds.), Idea Group Publishing, USA, Chapter XII, 2001. p. 231-259. WWW <http://www.dca.fee.unicamp.br/%7Eelnunes/immune.html> .
14. COLE, R. M.. *Clustering with genetic algorithms*. Tese de mestrado em University of Western Austrália, 1998. WWW <http://www.cs.uwa.edu.au/pub/robvis/theses/RowenaCole.ps.gz>.
15. CORMEM, Th.H., LEISERSON Ch.E., RIVEST R.L.. *Introduction to Algorithms*. MIT Press & McGraw-Hill, 1991. 1028 p.
16. DAVÉ, R. N.. *Robust fuzzy clustering algorithms*, In: IEEE International Conference on Fuzzy Systems, 1, mar. – abr. 1993, San Francisco, California, 1993. p. 1281-1286.
17. DAVE, R. N.. *Validating fuzzy partitions obtained trough c-shells clustering*. Pattern Recognition Letters. 6 ed, v. 17, p. 613-623, mai. 1996.
18. DOMINGUES, H. H.. *Espaços métricos e introdução à topologia*. SP: Atual, 1982.184 p.
19. DUDA, R. O., HART, P. E.. *Pattern classification and scene analysis*. New York: Wiley. 1973. 482 p.
20. DUDA, R. O.. *Pattern classification for HCI*. Department of Electrical Engineering, San Jose State University, 1997. WWW: <http://www.engr.sjsu.edu/~knap./HCIRODPR>
21. DUNN, J. C.. *A Fuzzy relative of the isotada*. Process and Its Use in Detecting Compact Well-Separated Clusters, Journal of Cybernetics 3. p. 32-57, 1973.
22. DUNN, J. C.. *A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters*. Journal of Cybernetics, p. 32-57, 1976.
23. FRIGUI, H., KRISHNAPURAM, R.. *A robust algorithm for automatic extraction of an unknown number of clusters from noisy data*. Pattern Recognition Letters. 1-4 ed, v. 17, p. 1223-1232, dez. 1996.
24. FUKUYAMA Y., SUGENO M.. *A new method for choosing the number of clusters for fuzzy c-means method*. In Proc. 5th Fuzzy System Symposium. 1989, p. 247-250.
25. GATH, I., GEVA, A. B.. *Unsupervised optimal fuzzy clustering*. IEEE Transactions Pattern Analysis and Machine Intelligence, v. PAMI-11, n. 7, p. 773-781, jul. 1989.
26. GATH, I., ISKOZ A. S., CUTSEM, B. V. M.. *Data induced metric and fuzzy clustering of non-convex patterns of arbitrary shape*. Pattern Recognition Letters. v. 18, p. 541-553, 1997.

27. GEVA, A. B. *et al.*. ***A comparison of cluster validity criteria for a mixture of normal distributed data.*** Pattern Recognition Letters. 5 ed, v. 21, p. 511-529, mai. 2000.
28. GEVA, A. B.. ***Unsupervised optimal fuzzy clustering.*** IEEE Transactions Pattern Analysis and Machine Intelligence, v. PAMI-11, n. 7, jul. 1989, p. 773-781. in: Bezdek and Pal, 1992.
29. GEVA, A. B.. ***Hierarchical-fuzzy clustering of temporal-patterns and its application for time-series prediction.*** Pattern Recognition Letters. 14 ed, v. 20, p. 1519-1532, dez. 1999.
30. GUHA S., RASTOGI, R., SHIM, K.. ***Cure: an efficient clustering algorithm for large databases.*** In: Proc. of SIGMOD, 1998.
31. GUHA S., RASTOGI, R., SHIM, K.. ***Rock: a robust clustering algorithm for categorical attributes.*** In: ICDE 1999. p. 512-521.
32. GUOYAO, F.. ***Optimization methods for fuzzy clustering.*** Fuzzy Sets and Systems. ed. 2, v. 93, p. 301-309, fev. 1998.
33. GUSTAFSON, D. E., KESSEL, W.. ***Fuzzy clustering with a fuzzy covariance matrix.*** In: Proc IEEE CDC jan. 10-12, 1979, San Diego, CA, 1979. p. 761-766., in: Bezdek and Pal, 1992.
34. HALKIDI, M., BATISTAKIS, Y. VAZIRGIANNIS, M. ***On clustering validation techniques.*** In: JIIS 17, 2-3. 2001:107-145. WWW: http://www.db_net.aueb.gr/mhajk/papers/validity_survey.pdf.
35. HAN, J., KAMBER, M.. ***Data mining: concepts and techniques.*** Morgan Kaufmanns Publishers. San Francisco, USA. 2001. 550 p.
36. JAIN A., DUBES R.. ***Algorithms for clustering data.*** Englewood Cliffs: Prentice Hall, 1988. 320 p.
37. JAIN A. K., MURTY M. N., FLYNN P. J.. ***Data clustering: a review.*** ACM Computing Surveys. v. 31, n. 3, set. 1999.
38. KARYPIS, G., HONG, E., KUMAR, V.. ***Chameleon: hierarchical clustering using dynamic modeling.*** Computer. p. 68-74, ago. 1999.
39. KAYMAK, U., SETNES, M.. ***Extended fuzzy clustering algorithms.*** In: ERIM – Report Series Research in Management – ERS – 2000-51-LIS, publicado novembro 2000, Erasmus Research Institute of Management (ERIM)/ Faculdade Bedrijfskunde. WWW: <http://www.irim.eur.nl>. p. 24.
40. KLIR, G., YUAN, B.. ***Fuzzy sets and fuzzy logic: theory and applications.*** New Jersey: Prentice Hall, 1995.
41. KOHONEN, T.. ***Self-organizing maps.*** Berlin; Heidelberg; New York: Springer, 1995.
42. KRISHNAPURAM, R., NASRAOUI, O., FRIGUI H.. ***The fuzzy c spherical shells algorithm: a new approach.*** IEEE Transactions on Neural Networks, v. 3, n. 5, p. 663-671. set. 1992.

43. LIN, C. T., LEE C. S. G.. *Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems*. London: Printice Hall, 1995. p. 180-202.
44. LIU, J., XIE, W.. *A genetic-based approach to fuzzy clustering*. In Proc. FUZZ-IEEE'95. Yokohama, Japão, mar. 20-24, 1995. v. 4, p. 2233-2240.
45. PEDRYCZ, W., GOMIDE, F.. *An introduction to fuzzy sets: analysis and design*, Hardcover, Cambridge, Massachusetts, EUA: MIT, mai. 1998, 475p.
46. ROUBENS, M.. *Fuzzy clustering algorithms and their cluster validity*. European Journal of Operational Research, second Euro IV. Special Issue. v. 10, n3, p. 294-301, jul. 1982.
47. RUNKLER, T. A., BEZDEK J. C.. *Alternating cluster estimation: a new tool for clustering and function approximation*. IEEE Transactions on Fuzzy Systems. v. 7, n. 4, p. 377-393, ago. 1999.
48. RUSPINI, E. H.. *A new approach to clustering*. Information Control. n. 15 p. 22-32. 1969.
49. SINTAS, A. F., CADENAS, J. M., MARTIN, F.. *Partition validity and defuzzification*. Fuzzy Sets and Systems. v. 112, p. 433-447, 2000.
50. VON ZUBEN, F.. Notas de aula – IA353, tópico: 9, 2002. WWW <http://www.dca.fee.unicamp.br/~vonzuben/courses/ia353.html> .
51. WINDHAN, M. P.. *Cluster validity for the fuzzy c-means clustering algorithm*. IEEE Transactions Pattern Analysis and Machine Intelligence, v. PAMI-4,n 4, p. 357-363, jul. 1982.
52. XIE, X., BENI, G.. *A validity measure for fuzzy clustering*. Ing IEEE Transactions Pattern Analysis and Machine Intelligence. v. 13, n.8, p. 841-847, 1991.
53. YAGER, R., FILEV, D.. *Approximate clustering via mountain method*. IEEE Transactions on Systems, Man and Cybernetics. v. 24, n. 8, p. 1279–1284, ago. 1994.
54. YAGER, R. R.. *A model of participatory learning*. IEEE Transactions on Systems, Man and Cybernetics. v. 20, n. 5, p. 1229-1234, set/out. 1990.
55. ZAHID N. *et al.*. *Unsupervised fuzzy clustering*. Pattern Recognition Letters. 2 ed, v. 20, p. 123-129, fev. 1999.

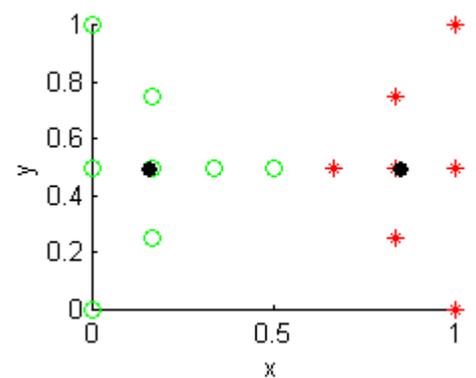
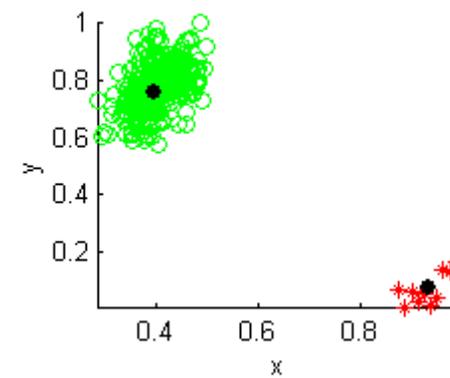
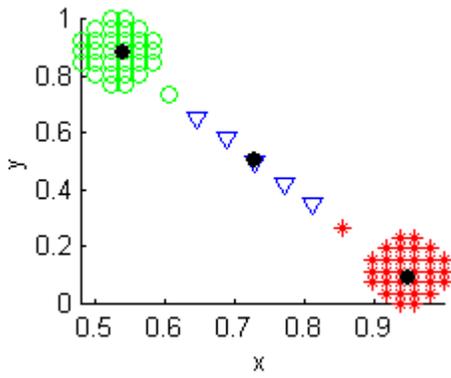
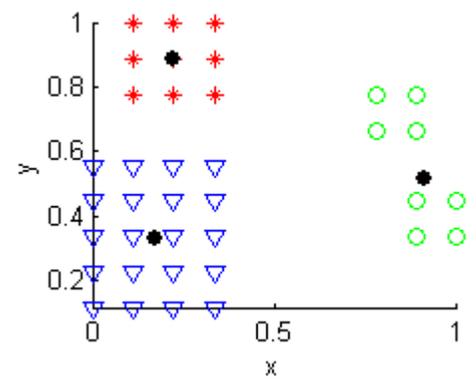
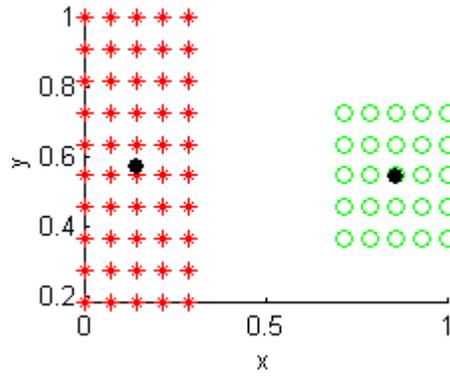
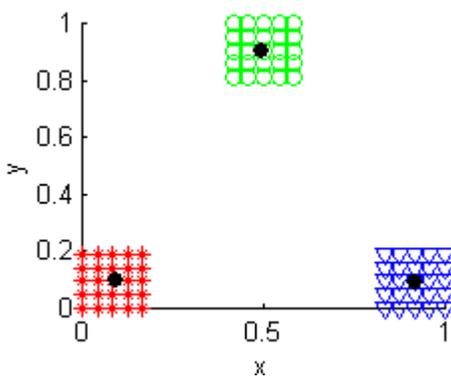
Índice Remissivo por Autores

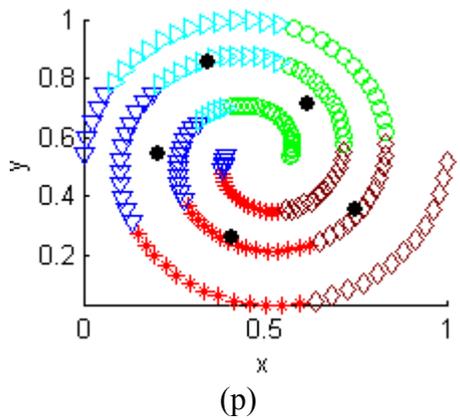
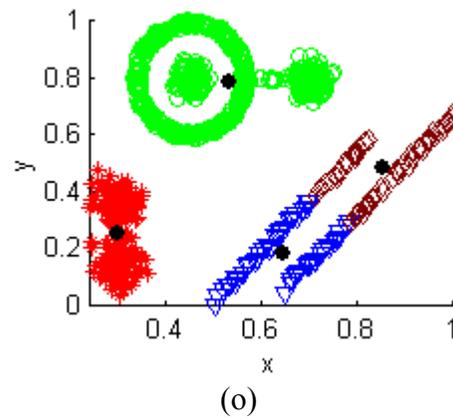
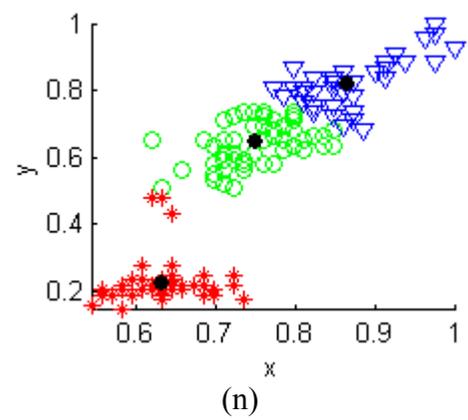
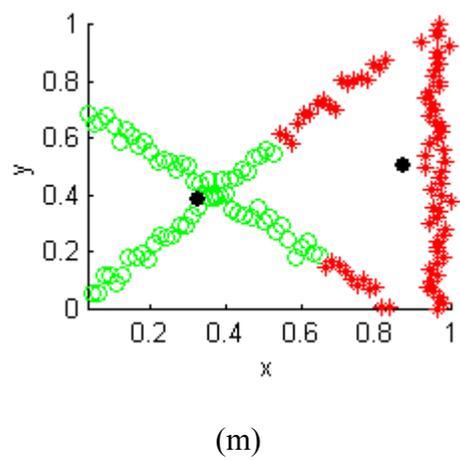
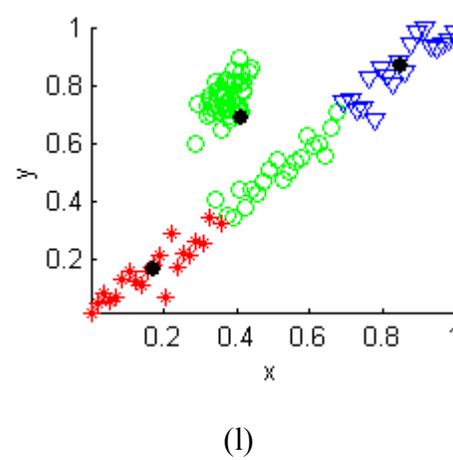
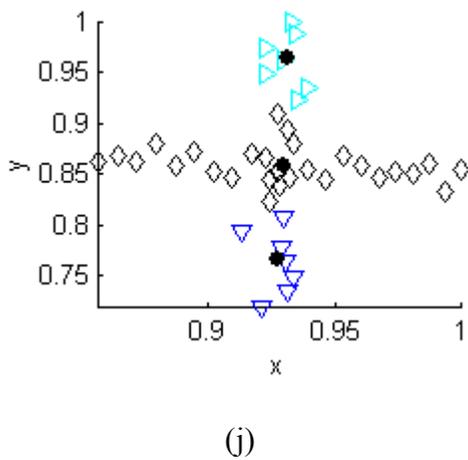
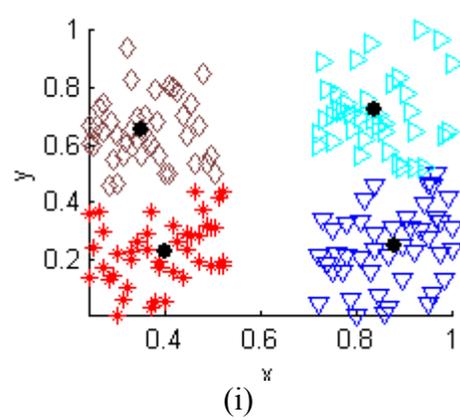
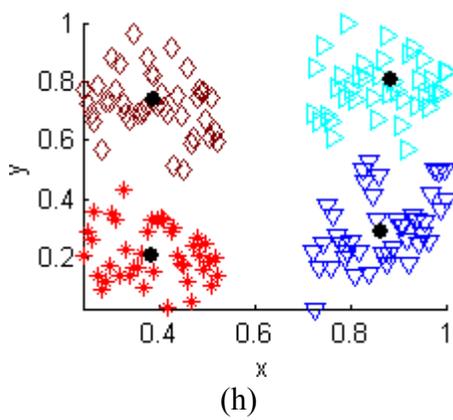
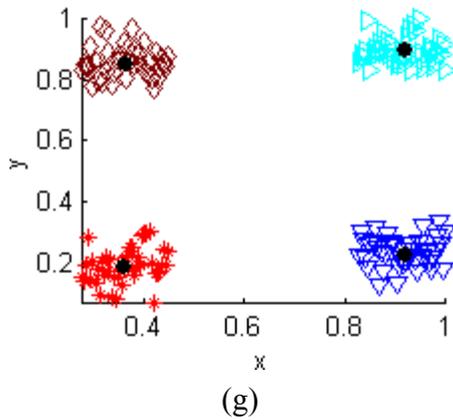
1. Al-Sultan, 1996	33
2. Anderson, 1935	71
3. Araújo, 2001	1
4. Backer and Scheunders, 1999a	1, 26
5. Backer and Scheunders, 1999b	1, 27, 28
6. Backer, 2002	28, 33
7. Ball and Hall, 1965	23
8. Baraldi and Blonda, 1998	8, 9
9. Bezdek and Pal, 1992	1, 17, 65
10. Bezdek and Pal, 1998	39
11. Bezdek, 1974	39
12. Bezdek, 1981	1, 19, 23, 24, 29
13. Castro and Von Zuben, 2001	71
14. Cole, 1998	1, 2
15. Cormem <i>et al.</i> , 1991	32
16. Davé, 1993	33
17. Dave, 1996	29, 36, 39
18. Domingues, 1982	10, 11
19. Duda and Hart, 1973	17, 67, 71
20. Duda, 1997	13
21. Dunn, 1973	23
22. Dunn, 1976	39
23. Frigui and Krishnapuram, 1996	33
24. Fukuyama and Sugeno, 1989	38, 39
25. Gath and Geva, 1989	7, 16, 17, 29, 35, 37, 39
26. Gath <i>et al.</i> , 1997	29, 30, 71
27. Geva <i>et al.</i> , 2000	33, 39
28. Geva, 1989	7
29. Geva, 1999	7, 37, 39
30. Guha <i>et al.</i> , 1998	33
31. Guha <i>et al.</i> , 1999	33
32. Guoyao, 1998	33
33. Gustafson and Kessel, 1979	1, 25, 29, 33, 69, 70
34. Halkidi <i>et al.</i> , 2001	33, 38, 39
35. Han and Kamber, 2001	9, 17, 20, 63, 65
36. Jain and Dubes, 1988	6, 9, 13, 15, 18
37. Jain <i>et al.</i> , 1999	1, 7, 17, 70, 71
38. Karypis <i>et al.</i> , 1999	1, 33
39. Kaymak and Setnes, 2000	29, 33, 61, 71
40. Klir, 1995	68, 71
41. Kohonen, 1995	23, 27
42. Krishnapuram and Frigui, 1992	29
43. Lin and Lee, 1995	36
44. Liu and Xie, 1995	69, 71

45. Pedrycz and Gomide, 1998	42
46. Roubens, 1982.....	36, 69, 71
47. Runkler and Bezdek, 1999	20, 21
48. Ruspini, 1969	23
49. Sintas <i>et al.</i> , 2000	39
50. Von Zuben, 2002.....	5
51. Windham, 1982	69, 71
52. Xie and Beni, 1991.....	38, 39
53. Yager and Filev, 1994.....	33, 67, 71
54. Yager, 1990.....	3, 41
55. Zahid <i>et al.</i> , 1999	7, 33

Apêndice I

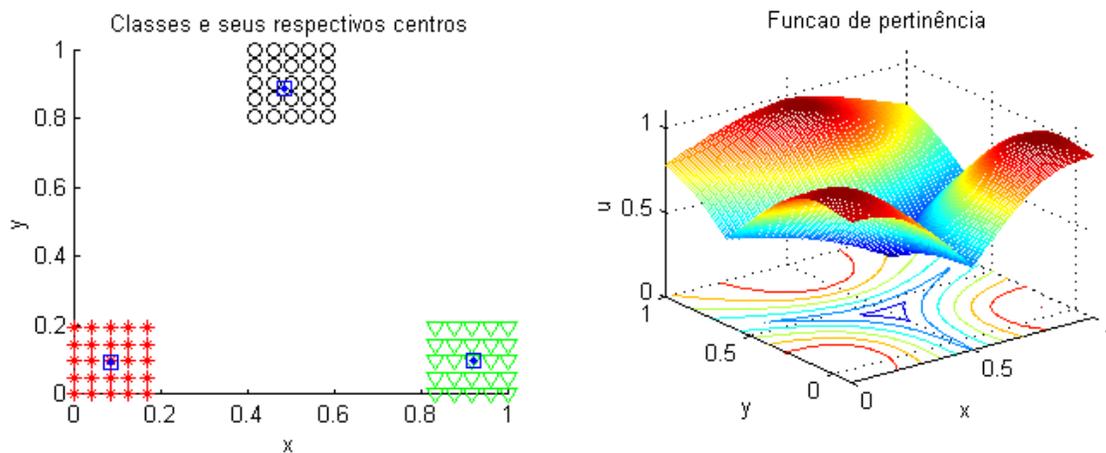
Resultados do algoritmo A – AP para os conjuntos de dados das Figuras 5.1 a 5.5.



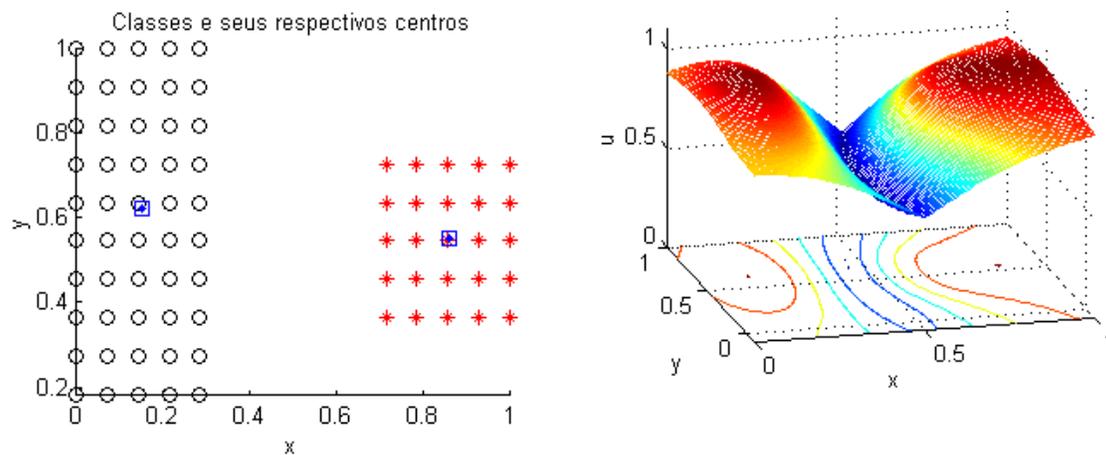


Apêndice II

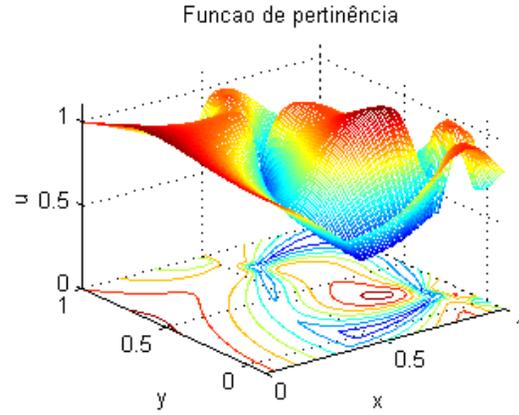
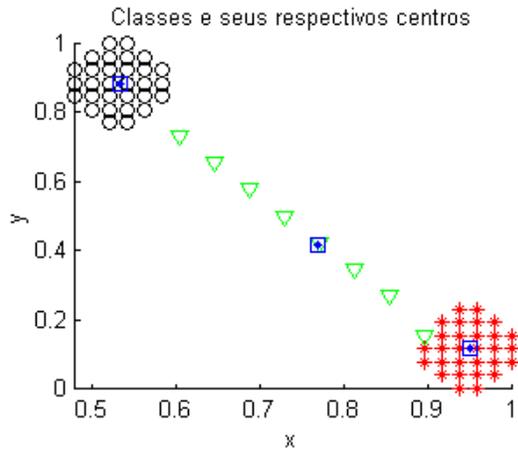
Funções de Pertinência do algoritmo AP-B para as Figuras de “a” a “n”.



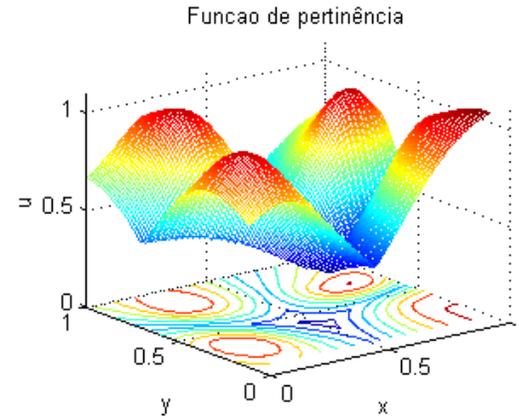
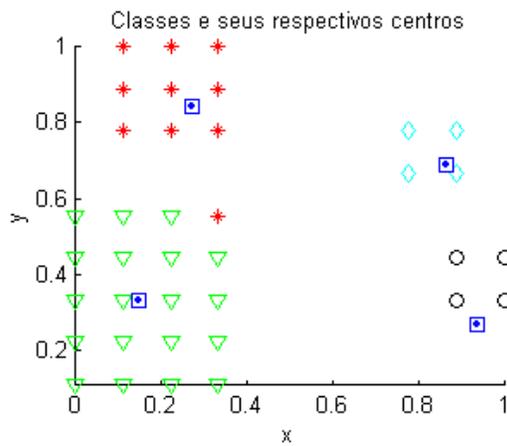
Função de pertinência para o conjunto de dados da Figura 5.1.a.



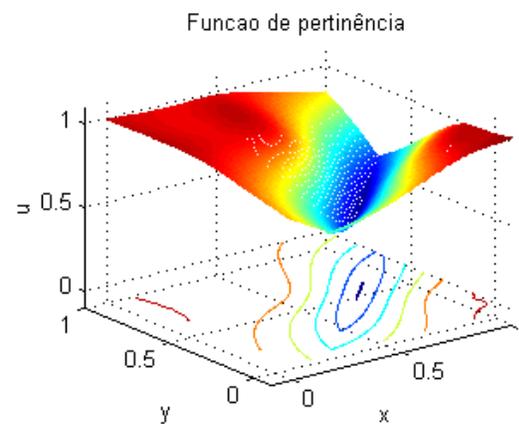
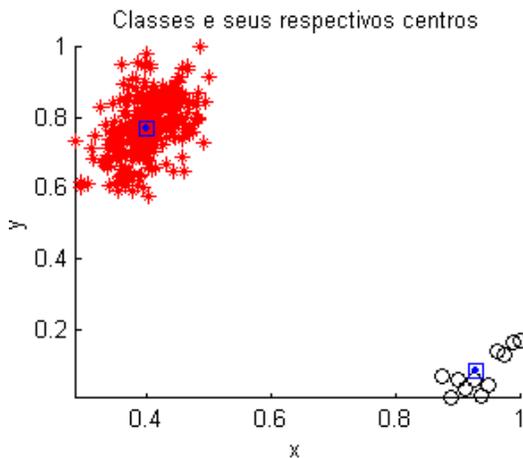
Função de pertinência para o conjunto de dados da Figura 5.1.b.



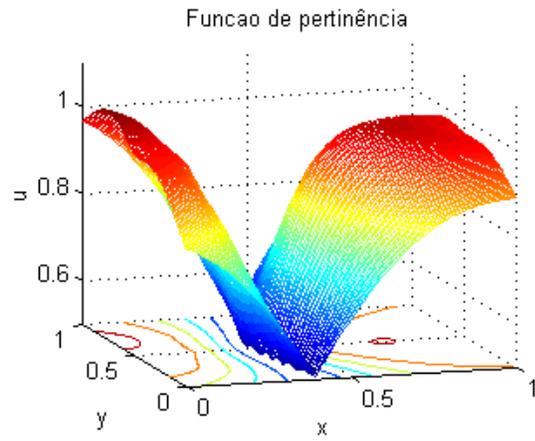
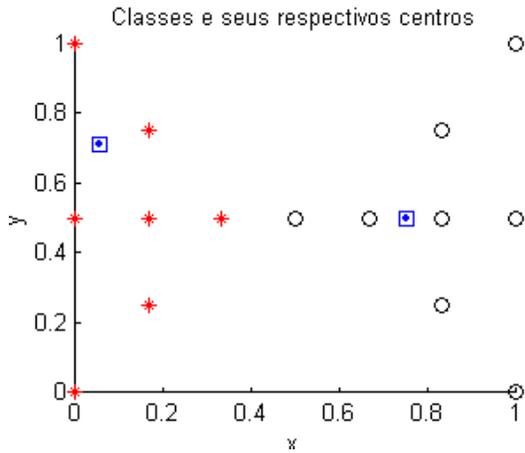
Função de pertinência para o conjunto de dados da Figura 5.1.c.



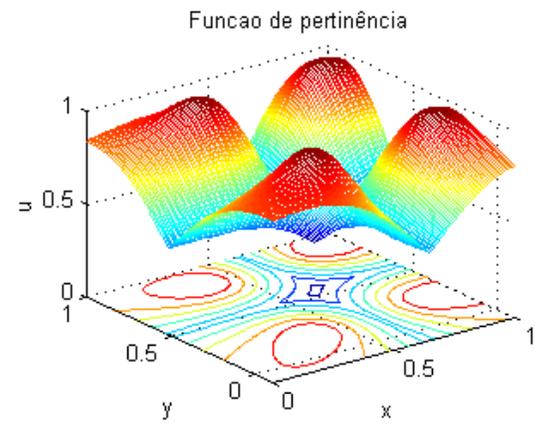
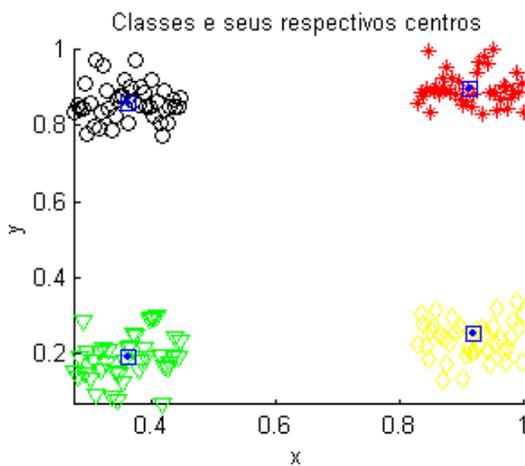
Função de pertinência para o conjunto de dados da Figura 5.2.d.



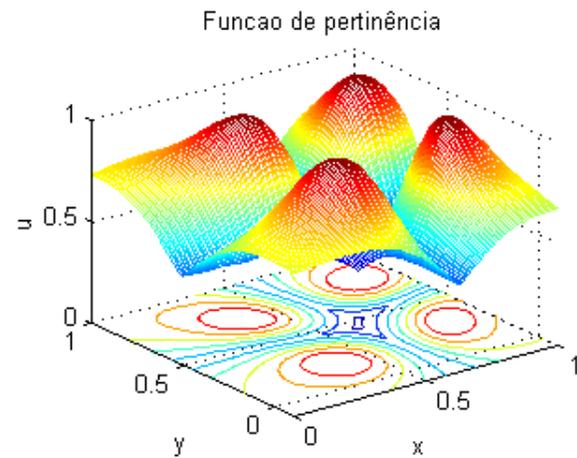
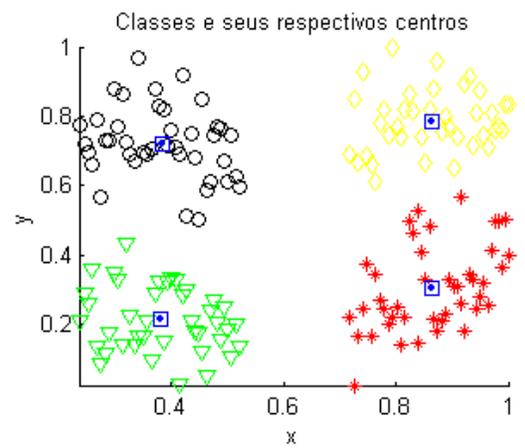
Função de pertinência para o conjunto de dados da Figura 5.2.e.



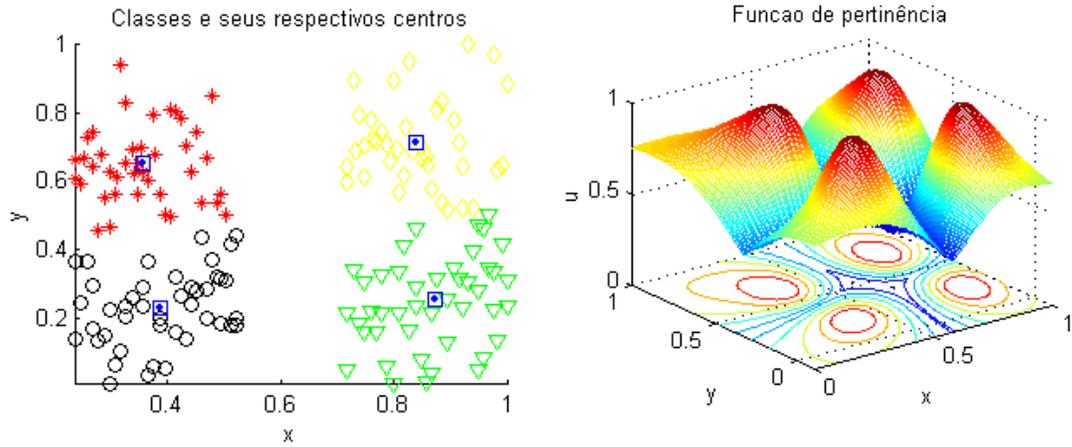
Função de pertinência para o conjunto de dados da Figura 5.2.f.



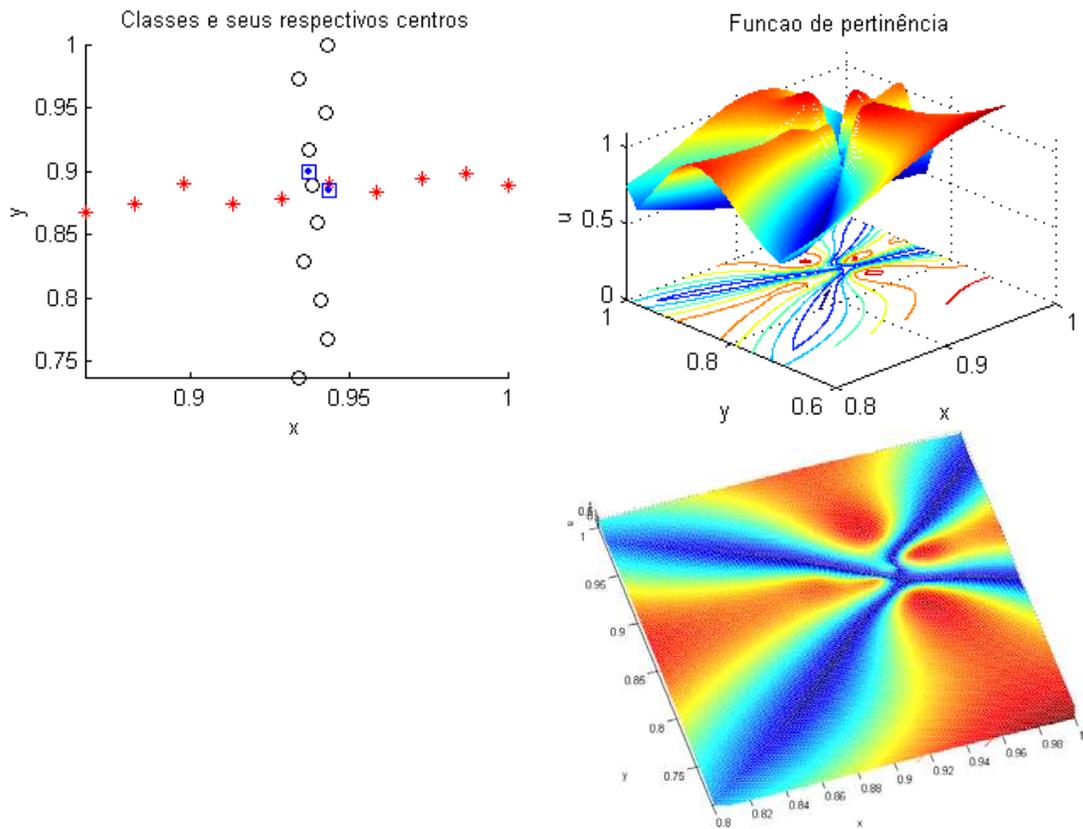
Função de pertinência para o conjunto de dados da Figura 5.3.g.



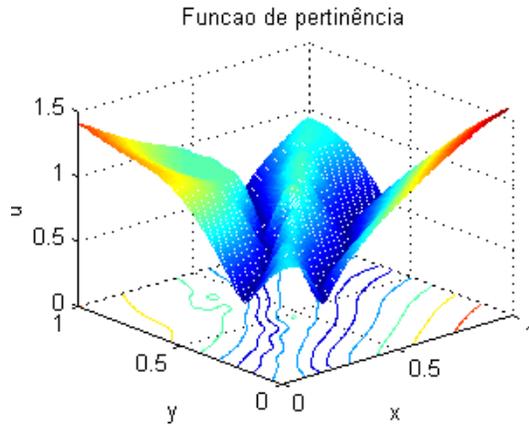
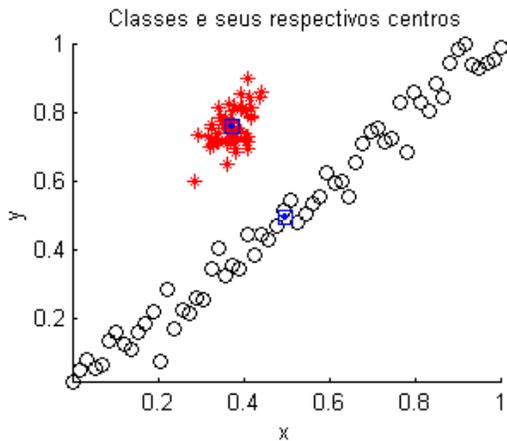
Função de pertinência para o conjunto de dados da Figura 5.3.h.



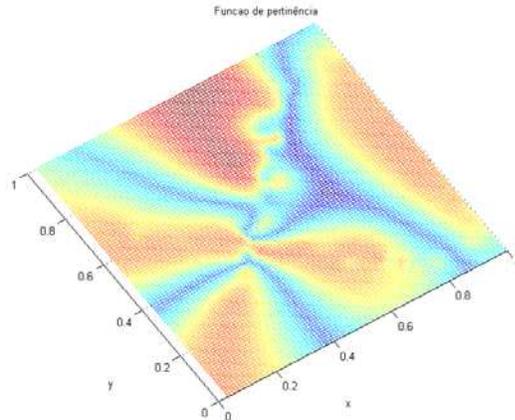
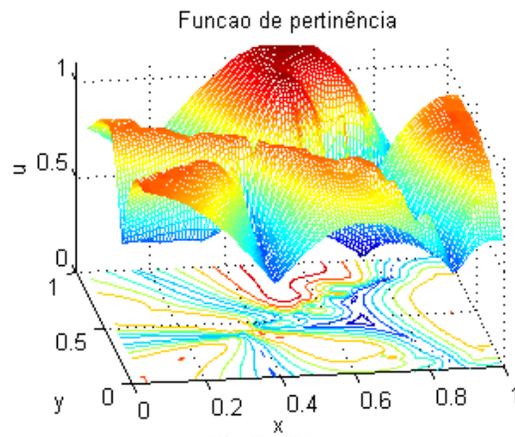
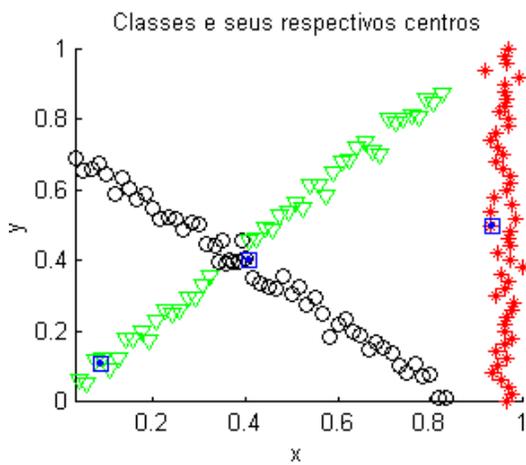
Função de pertinência para o conjunto de dados da Figura 5.3.i.



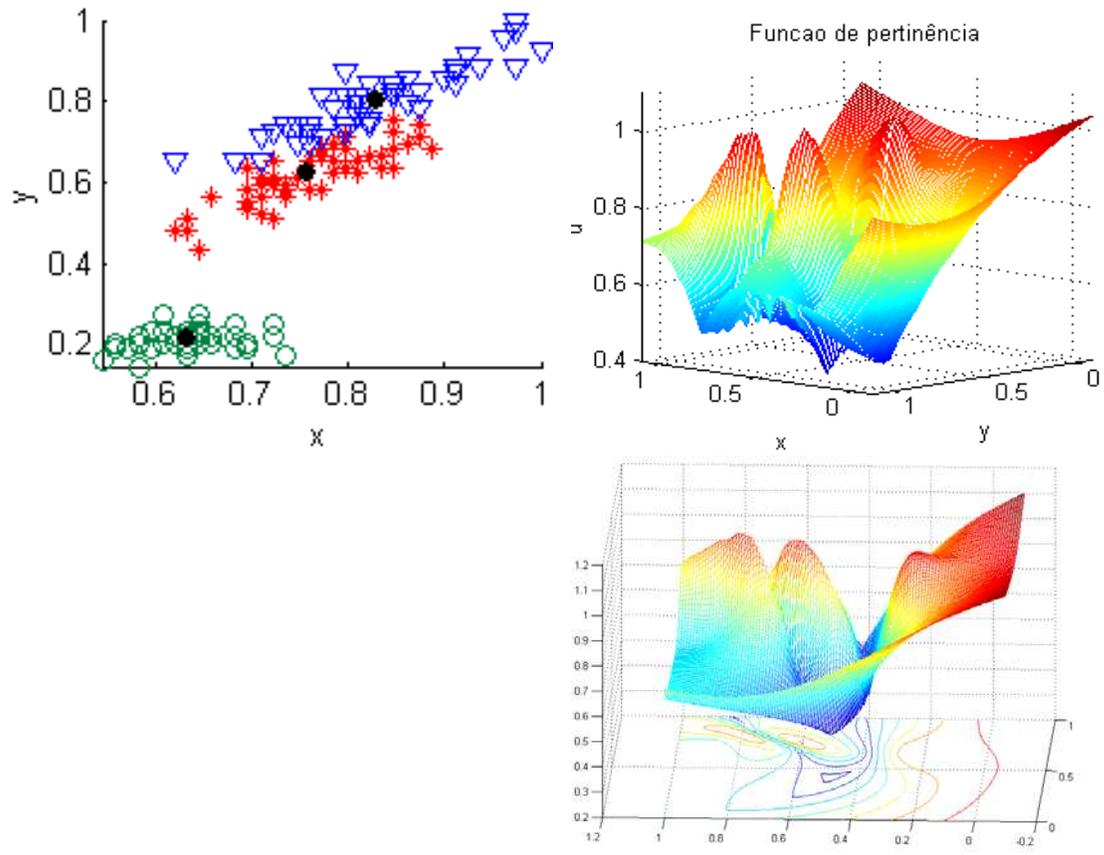
Função de pertinência para o conjunto de dados da Figura 5.4.j. além de uma visão somente com as curvas de níveis.



Função de pertinência para o conjunto de dados da Figura 5.4.l.



Função de pertinência para o conjunto de dados da Figura 5.4.m. além de uma visão somente com as curvas de níveis.



Função de pertinência para o conjunto de dados da Figura 5.5.n., também com um outro ângulo da mesma função de pertinência.

Apêndice III

Este apêndice apresenta os resultados dos testes realizados com as funções de validação e os algoritmos FCM, GK, FCL, FFSECL, MFKM e AP utilizando os conjuntos de dados “a, e, g, i, f, d, n, l” .

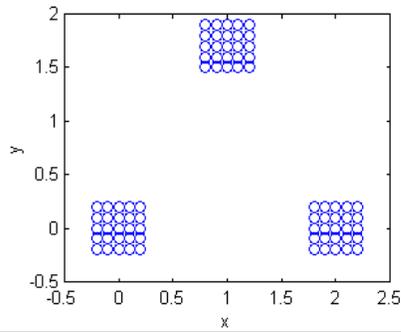
Estes testes foram realizados com todas as funções de validação descritas neste trabalho, exceto a Partição por Hiper volume e Partição por Densidade. Estas duas funções foram excluídas por não conseguirem avaliar corretamente nenhum dos conjuntos de dados “a” e “n”. Estes conjuntos de dados foram utilizados para verificar o desempenho das funções. As funções que não conseguiram avaliar corretamente pelo menos um destes conjuntos foram eliminadas.

Em todos os testes, o número de grupos, parâmetro suposto conhecido para os algoritmos supervisionados, foram inicializados aleatoriamente. Para os algoritmos MFKM e AP, ambos não supervisionados, foi necessário ajustar os parâmetros η e τ para se obter o número de grupos esperado.

Legenda:

PE	: Partição Entrópica;
XB	: Xie-Beni;
FG	: Fukuyama-Sugeno;
PC	: Coeficiente de Partição;
AD	: Partição pela Densidade Média;
min	: minimiza função;
max	: maximiza função;
	: número de grupos desejados;
Valor de τ	: limiar de incompatibilidade entre os grupos.

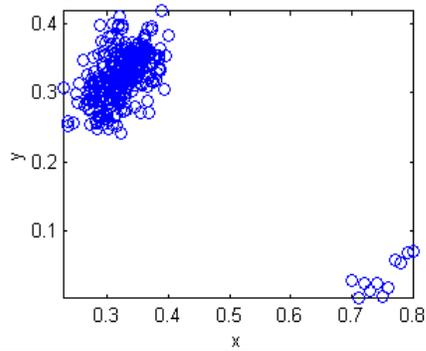
três grupos bem separados



Função	no. Classe	Algoritmos						valor de τ
		FCM	GK	FCL	FFSECL	MFKM	AP	
PE min	2	0,29	0,09	0,30	0,25	0,09	0,09	0.12
	3	0,05	0,05	0,06	0,05	0,02	0,05	0.05
	4	0,12	0,11	0,16	0,20	0,13	0,14	0.015
	5	0,20	0,17	0,46	0,48	0,12	0,18	0.01
XB min	2	11,82	0,23	83,35	185,72	0,04	0,25	
	3	0,01	0,01	0,02	0,06	0,03	0,01	
	4	0,49	0,43	1,27	2,61	1,94	0,20	
	5	0,54	0,43	7,43E+06	74,94	3,10	0,10	
FG min	2	10,34	0,85	10,03	16,07	-9,03	1,38	
	3	-18,26	-18,26	-17,48	-20,34	-12,87	-18,25	
	4	-16,52	-16,85	-14,26	-14,15	-9,13	-11,91	
	5	-14,07	-15,19	-3,68	-4,28	-1,29	-7,67	
PC max	2	0,53	0,89	0,50	0,81	0,63	0,88	
	3	0,96	0,96	0,94	1,28	0,82	0,96	
	4	0,86	0,88	0,81	1,01	0,57	0,64	
	5	0,74	0,79	0,42	0,51	0,12	0,36	
AD Max	2	10,02	790,06	76,79	101,19	1175,56	710,19	
	3	2222,71	2059,65	1316,59	1355,96	1311,47	1587,37	
	4	1532,93	1551,27	1064,54	944,89	825,63	1189,36	
	5	1421,78	1427,82	231,23	274,92	892,30	1324,82	

Resultados das funções de validação de acordo com as funções de pertinências, geradas por seus respectivos algoritmos de agrupamento, para o conjunto de dados da Figura 5.1.a.

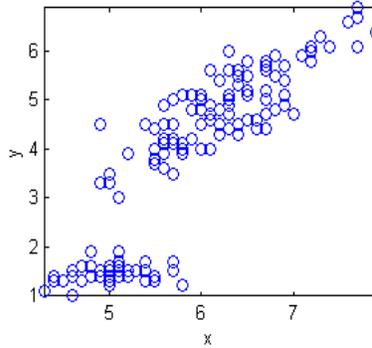
dois grupos bem separados e com diferentes tamanhos



bd conj-e

Função	no. Classe	Algoritmos						valor de τ
		FCM	GK	FCL	FFSECL	MFKM	AP	
PE min1	2	0,09	0,01	0,05	0,00	0,06	0,01	0.05
	3	0,16	0,18	0,16	0,17	0,26	0,19	0.008
	4	0,27	0,28	0,39	0,45	0,51	0,28	0.005
	5	0,39	0,35	0,49	0,57	0,59	0,29	0.004
XB min	2	0,03	0,01	0,02	0,09	0,01	0,01	
	3	0,15	0,19	0,21	1,80	4,88	0,19	
	4	0,29	0,21	2,02	5,63	8,42	0,16	
	5	1,73	0,54	7,58	13,57	41,32	0,10	
FG min	2	-2,36	-5,68	-7,01	-25,57	-1,35	-5,23	
	3	-8,08	-7,91	-6,34	-12,99	-0,83	-7,52	
	4	-8,24	-8,35	1,67	3,99	0,41	-8,25	
	5	-1,24	-8,34	1,48	2,94	0,32	-2,33	
PC max	2	0,89	0,99	0,95	1,56	0,72	0,99	
	3	0,77	0,74	0,78	1,20	0,48	0,73	
	4	0,65	0,64	0,51	0,79	0,41	0,64	
	5	0,53	0,58	0,42	0,59	0,38	0,24	
AD max	2	22137,78	23229,32	10456,55	19025,51	22958,70	19105,48	
	3	23110,85	23838,77	18805,31	21115,50	29602,97	21538,00	
	4	23613,20	24629,09	9076,65	9501,07	19221,74	16950,67	
	5	13308,24	23119,89	7644,06	7758,72	18603,36	22853,22	

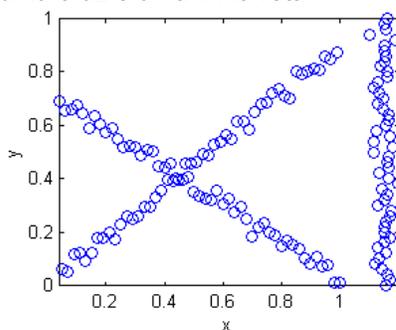
Resultados das funções de validação de acordo com as funções de pertinências, geradas por seus respectivos algoritmos de agrupamento, para o conjunto de dados da Figura 5.2.e.



Função	no. Classe	Algoritmos						valor de τ
		FCM	GK	FCL	FFSECL	MFKM	AP	
PE min	2	0,21	0,09	0,20	0,15	0,17	0,08	0.04
	3	0,15	0,14	0,18	0,19	0,22	0,16	0.01
	4	0,19	0,21	0,42	0,47	0,36	0,22	0.008
	5	0,26	0,28	0,38	0,37	0,30	0,24	0.005
XB min	2	0,65	0,05	0,58	3,50	3,55	0,04	
	3	0,11	0,17	0,26	0,93	10,02	0,44	
	4	0,14	0,17	4,26	6,61	9,62	0,25	
	5	0,59	0,68	32,61	50,91	2,11	0,13	
FG min	2	4,59	-6,68	2,15	8,48	-1,13	-7,31	
	3	-8,32	-7,98	-5,79	-4,51	-4,29	-7,20	
	4	-8,31	-8,21	0,71	2,04	-0,78	-7,34	
	5	-6,98	-6,99	-4,81	-6,67	-4,24	-5,89	
PC max	2	0,67	0,88	0,71	1,11	0,43	0,89	
	3	0,82	0,82	0,77	0,96	0,48	0,79	
	4	0,77	0,75	0,47	0,55	0,27	0,73	
	5	0,68	0,66	0,52	0,75	0,40	0,47	
AD max	2	1892,38	7219,97	2327,89	2405,25	3868,35	7540,45	
	3	8935,39	10081,53	7649,24	7341,80	10700,54	9174,85	
	4	8987,14	10029,97	1910,95	1764,79	2689,46	8753,78	
	5	8509,60	8598,08	2583,20	3191,36	8649,67	7364,61	

Resultados das funções de validação de acordo com as funções de pertinências, geradas por seus respectivos algoritmos de agrupamento, para o conjunto de dados da Figura 5.5.n.

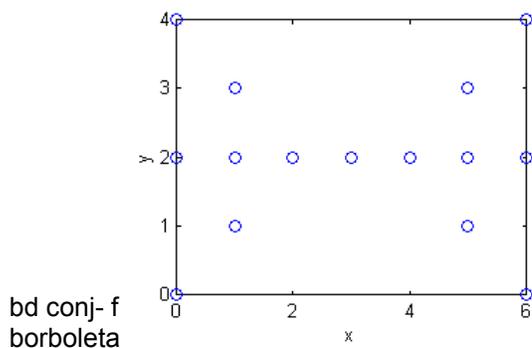
uma cruz e uma linha reta



bd conj-m

Função	no. Classe	Algoritmos						valor de τ
		FCM	GK	FCL	FFSECL	MFKM	AP	
PE min	2	0,18	0,12	0,18	0,10	0,14	0,12	0.15
	3	0,23	0,23	0,35	0,35	0,18	0,17	0.1
	4	0,32	0,16	0,36	0,40	0,29	0,26	0.07
	5	0,34	0,18	0,44	0,51	0,34	0,17	0.04
XB min	2	0,24	0,20	0,16	0,36	0,03	0,22	
	3	0,10	0,10	4,09	7,72	0,11	0,52	
	4	0,26	0,62	0,50	0,81	0,20	0,34	
	5	0,20	0,78	3,97	47,51	1,59	0,73	
FG min	2	3,11	-0,35	-2,50	2,88	-0,14	0,55	
	3	-8,46	-8,57	2,54	5,25	-3,15	-0,21	
	4	-8,83	-3,67	-4,18	-2,60	-3,47	-8,91	
	5	-10,55	-7,64	-9,16	-6,05	-5,13	-6,20	
PC max	2	0,72	0,84	0,73	1,28	0,12	0,84	
	3	0,71	0,72	0,52	0,71	0,48	0,78	
	4	0,61	0,80	0,57	0,66	0,31	0,68	
	5	0,61	0,78	0,49	0,50	0,43	0,80	
AD max	2	521,08	915,64	506,72	486,21	261,18	895,12	
	3	800,98	806,92	358,74	388,54	690,06	1238,64	
	4	694,31	2370,22	536,74	526,24	861,85	1086,80	
	5	961,23	3137,13	395,45	324,16	649,73	3364,42	

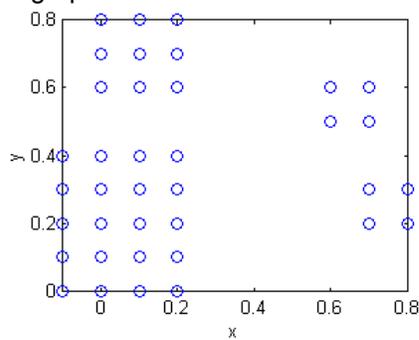
Resultados das funções de validação de acordo com as funções de pertinências, geradas por seus respectivos algoritmos de agrupamento, para o conjunto de dados da Figura 5.4.m.



Função	no. Classe	Algoritmos						valor de τ
		FCM	GK	FCL	FFSECL	MFKM	AP	
PE min	2	0,21	0,10	0,21	0,14	0,16	0,17	0.3
	3	0,29	0,18	0,30	0,31	0,20	0,19	0.22
	4	0,35	0,20	0,35	0,35	0,34	0,17	0.2
	5	0,48	0,24	0,51	0,54	0,34	0,25	0.1
XB min	2	0,38	0,17	0,89	1,81	0,46	0,46	
	3	0,60	0,46	0,30	0,49	0,10	2,03	
	4	0,30	0,23	0,53	0,81	137,56	0,31	
	5	3,97E+08	1,10	2,74 E+08	4,97E+05	31,09	10,05	
FG min	2	0,82	-0,40	1,15	2,14	0,55	0,77	
	3	-0,23	-0,29	-0,01	-0,04	-0,70	-0,85	
	4	-0,90	-1,46	-0,45	-0,41	-0,02	-0,56	
	5	-0,37	-1,29	0,17	0,22	-0,64	-0,81	
PC max	2	0,67	0,87	0,69	1,08	0,54	0,76	
	3	0,62	0,76	0,60	0,76	0,30	0,76	
	4	0,57	0,77	0,57	0,74	0,35	0,56	
	5	0,42	0,73	0,41	0,50	0,29	0,70	
AD max	2	50,47	96,20	48,04	53,90	56,74	59,38	
	3	56,77	72,65	32,31	31,67	650665,06	57,08	
	4	49,55	46,26	46,81	46,74	30,90	123,80	
	5	33,56	168,30	22,40	23,17	381608,38	5791,15	

Resultados das funções de validação de acordo com as funções de pertinências, geradas por seus respectivos algoritmos de agrupamento, para o conjunto de dados da Figura 5.2.f.

4 grupos com tamanhos diferentes

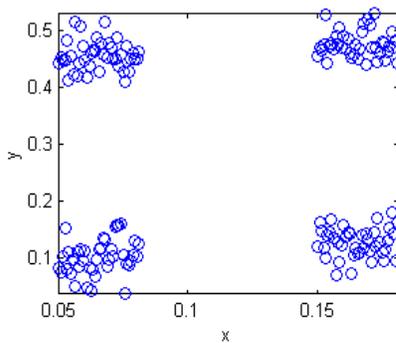


bd conj-d

Função	no. Classe	Algoritmos						valor de τ
		FCM	GK	FCL	FFSECL	MFKM	AP	
PE min	2	0,27	0,12	0,27	0,23	0,13	0,16	1.1
	3	0,35	0,21	0,33	0,33	0,24	0,17	0.07
	4	0,27	0,22	0,41	0,46	0,33	0,24	0.05
	5	0,29	0,27	0,43	0,53	0,31	0,31	0.03
XB min	2	1,48	0,19	5,13	12,28	0,16	0,56	
	3	1,18	0,24	5,95	10,01	0,21	0,11	
	4	0,25	0,24	2,26	3,77	0,46	0,15	
	5	0,15	0,15	3,36	5,58	0,71	0,40	
FG min	2	3,30	-0,10	3,35	5,34	-0,34	2,37	
	3	0,76	-0,79	0,20	0,77	-0,58	-3,40	
	4	-3,82	-4,23	-0,40	0,08	-0,28	-4,20	
	5	-4,35	-4,67	-1,28	-1,26	0,08	-3,23	
PC max	2	0,57	0,83	0,57	0,86	0,53	0,78	
	3	0,53	0,74	0,54	0,76	0,56	0,80	
	4	0,68	0,75	0,49	0,58	0,48	0,73	
	5	0,67	0,70	0,52	0,50	0,39	0,65	
AD max	2	65,91	195,75	82,00	97,44	134,06	109,48	
	3	108,20	115,13	126,35	142,24	143,17	286,21	
	4	200,90	116,10	124,68	118,92	83,28	198,06	
	5	147,21	156,76	84,44	78,91	98,55	161,90	

Resultados das funções de validação de acordo com as funções de pertinências, geradas por seus respectivos algoritmos de agrupamento, para o conjunto de dados da Figura 5.2.d.

4 grupos bem distintos

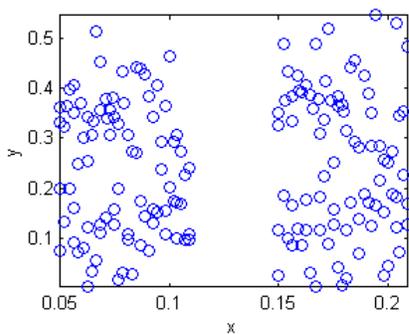


bd conj-g

Função	no. Classe	Algoritmos						valor de τ
		FCM	GK	FCL	FFSECL	MFKM	AP	
PE min	2	0,17	0,04	0,18	0,12	0,04	0,02	0.15
	3	0,18	0,11	0,29	0,33	0,09	0,08	0.08
	4	0,07	0,07	0,11	0,11	0,12	0,07	0.03
	5	0,13	0,12	0,34	0,41	0,15	0,12	0.01
XB min	2	0,51	0,36	0,48	0,98	0,00	0,80	
	3	0,11	0,12	3,59	5,12	1,52	0,02	
	4	0,02	0,02	0,04	0,10	0,00	0,02	
	5	0,42	0,37	2,37E+08	6088,25	1,35	0,33	
FG min	2	9,32	5,99	4,33	11,87	-9,57	-4,41	
	3	-13,47	-15,42	-3,98	3,47	-8,58	-13,25	
	4	-27,91	-27,89	-24,64	-29,70	-12,55	-27,70	
	5	-25,56	-26,34	-10,42	-10,08	-9,45	-26,28	
PC max	2	0,74	0,97	0,73	1,15	0,31	0,98	
	3	0,79	0,87	0,61	0,69	0,28	0,49	
	4	0,94	0,94	0,90	1,28	0,41	0,94	
	5	0,86	0,88	0,62	0,71	0,32	0,88	
AD max	2	654,94	549,78	696,36	766,04	5607,60	1707,90	
	3	2437,60	3068,46	701,11	475,75	4643,52	4037,69	
	4	5410,37	5418,11	4147,93	4497,64	5141,31	5456,77	
	5	4961,56	5222,29	1255,57	1221,31	5378,79	4263,34	

Resultados das funções de validação de acordo com as funções de pertinências, geradas por seus respectivos algoritmos de agrupamento, para o conjunto de dados da Figura 5.3.g.

4 grupos sobrepostos



Função	no. Classe	Algoritmos						valor de τ
		FCM	GK	FCL	FFSECL	MFKM	AP	
PE min	2	0,23	0,15	0,23	0,17	0,16	0,15	0.2
	3	0,26	0,22	0,34	0,30	0,17	0,21	0.1
	4	0,38	0,31	0,41	0,45	0,34	0,19	0.05
	5	0,32	0,30	0,51	0,57	0,41	0,31	0.04
XB min	2	0,69	0,30	1,09	3,23	0,19	0,32	
	3	0,13	0,15	2,69	5,14	0,12	1,00	
	4	0,37	0,29	0,98	1,79	0,30	0,06	
	5	0,14	0,17	19,55	28,09	1,20	0,15	
FG min	2	8,25	3,57	7,00	18,67	1,74	3,67	
	3	-4,08	-5,27	-5,77	-7,86	-3,96	-4,66	
	4	-10,34	-10,28	-1,19	-11,83	-1,99	-7,80	
	5	-10,14	-9,86	-2,26	-0,30	-2,08	-5,87	
PC max	2	0,72	0,80	0,64	1,10	0,45	0,80	
	3	0,66	0,73	0,53	0,87	0,40	0,74	
	4	0,54	0,62	0,50	0,62	0,39	0,53	
	5	0,65	0,67	0,41	0,46	0,44	0,42	
AD max	2	486,46	778,76	540,87	542,41	621,24	782,97	
	3	987,33	1223,81	418,21	648,90	682,44	1240,06	
	4	760,99	1009,09	518,98	572,27	480,37	1605,57	
	5	1326,94	1461,87	440,82	389,68	449,93	955,98	

Resultados das funções de validação de acordo com as funções de pertinências, geradas por seus respectivos algoritmos de agrupamento, para o conjunto de dados da Figura 5.3.i.