Algoritmos Aglomerativos de Agrupamento Baseados em Teoria de Matrizes

Rodrigo Costa Camargos

Julho / 2016

Dissertação de Mestrado em Ciência da Computação

Algoritmos Aglomerativos de Agrupamento Baseados em Teoria de Matrizes

Esse documento corresponde à dissertação de mestrado apresentada à Banca Examinadora da Dissertação no curso de Mestrado em Ciência da Computação da Faculdade Campo Limpo Paulista.

Campo Limpo Paulista, 14 de Julho de 2016.

Rodrigo Costa Camargos

Profa. Dra. Maria do Carmo Nicoletti
Orientadora

Faculdade Campo Limpo Paulista

Programa de Mestrado em Ciência da Computação

"Algoritmos Aglomerativos de Agrupamento Baseados em Teoria de Matrizes"

Rodrigo Costa Camargos

Dissertação de Mestrado apresentado ao Programa de Mestrado em Ciência da Computação da Faculdade Campo Limpo Paulista, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Membros da Banca:

Profa. Dra. Maria do Carmo Nicoletti
(Orientadora –FACCAMP)

Prof. Dr. Osvaldo Luiz de Oliveira
(FACCAMP)

Prof. Dr. Sergio Donizetti Zorzo
(UFSCar)

Campo Limpo Paulista, 14 de julho de 2016.

FICHA CATALOGRÁFICA

Dados Internacionais de Catalogação na Publicação (CIP)

Câmara Brasileira do Livro, São Paulo, Brasil.

Camargos, Rodrigo Costa

Algoritmos aglomerativos de agrupamento baseados em teoria de matrizes / Rodrigo Costa Camargos. Campo Limpo Paulista, SP: FACCAMP, 2016.

Orientadora: Profª. Drª. Maria do Carmo Nicoletti

Dissertação (Programa de Mestrado em Ciência da Computação) – Faculdade Campo Limpo Paulista – FACCAMP.

1. Algoritmos de agrupamento. 2. Aprendizado indutivo de máquina. I. Nicoletti, Maria do Carmo. II. Campo Limpo Paulista. III. Título.

CDD-005.1

Agradecimentos

Agradeço primeiro a minha família, por todo o suporte, esforço e dedicação que me proporcionaram ao longo destes anos.

À professora e orientadora Dra. Maria do Carmo Nicoletti pela orientação, incentivo e, principalmente, pelas inúmeras contribuições feitas durante o período de realização do trabalho.

Ao professor e coordenador do programa de mestrado em ciência da computação da Faccamp Dr. Osvaldo Luiz de Oliveira pelas contribuições e incentivo nesta importante etapa da minha formação.

Ao professor Dr. Sérgio Donizetti Zorzo pela disposição e contribuições ao trabalho examinado.

À CAPES pela bolsa concedida, sem a qual teria sido praticamente inviável a condução desta pesquisa.

Resumo. *Esta dissertação tem como foco principal a investigação de algoritmos de aprendizado de máquina não supervisionados identificados como algoritmos de agrupamento hierárquicos, particularmente aqueles que se enquadram na subcategoria de hierárquicos aglomerativos. Algoritmos de agrupamento hierárquico (AH) produzem uma hierarquia de agrupamentos aninhados, organizados como uma árvore hierárquica. Os chamados algoritmos de agrupamento aglomerativos (AA) podem ser abordados como uma categoria particular de AH, em que o processo de agrupamento é feito de maneira botton-up. De interesse particular neste trabalho são os algoritmos de agrupamento aglomerativos (AA) baseados em conceitos da Teoria de Matrizes, com o objetivo de (1) identificação das principais características de conjunto de padrões que promovem um bom desempenho deste tipo de algoritmo; (2) estudo, entre as várias alternativas existentes, para o cálculo de distância entre grupos. Os algoritmos aglomerativos, o algoritmo particional K-Means, as técnicas de pré-processamento de dados e os métodos de validação foram implementados e disponibilizados no sistema computacional AggloCluster, com o objetivo de oferecer uma plataforma para uso e avaliação de tais algoritmos. O trabalho apresenta e discute os resultados de experimentos realizados em conjuntos de dados sintéticos; tais resultados indicam que os algoritmos aglomerativos mostram bons resultados para a maioria dos conjuntos de dados utilizados neste trabalho.*

Abstract: *This dissertation is mainly focused on the investigation of unsupervised machine learning algorithms identified as hierarchical clustering algorithms, particularly those that fall under the subcategory of agglomerative clustering. Hierarchical clustering algorithms (HC) produce a hierarchy of nested clustering, organized as a hierarchical tree. The so-called agglomerative clustering algorithms (AC) can be approached as a particular category of HC, where the clustering process operates bottom-up. The main focus of this research are the agglomerative clustering algorithms (AC) based on matrix theory, with the objective of (1) identify data characteristics which promote good performance of AC; (2) study among the various alternatives of inter-group distance measure. The agglomerative algorithms, partitional algorithm K-Means, the data preprocessing techniques and validation methods have been implemented and made available in the AggloCluster computer system, aiming to provide a platform for use and evaluation of such algorithms. The work presents and discusses the results of experiments conducted on synthetic data set; results indicate that the agglomerative algorithms performed well for most of the data sets used in this research.*

SUMÁRIO

Lista de Siglas

AIM – Aprendizado Indutivo de Máquina

AH – Agrupamento Hierárquico

AA – Agrupamento Aglomerativo

EAG – Esquema Aglomerativo Generalizado

MUAS – *Matrix Updating Algoritmic Scheme* (Esquema Algorítmico de Atualização de Matrizes)

Lista de Tabelas

Lista de Figuras

Capítulo 1
Introdução

1.1 Contextualização

A área de pesquisa de Aprendizado Indutivo de Máquina (AIM) é uma subárea da Inteligência Artificial que investiga, principalmente, algoritmos computacionais que permitem a implementação da habilidade de realizar aprendizado automático, por computadores. Ao longo das últimas décadas inúmeras ideias de como viabilizar o aprendizado automático têm sido propostas e implementadas. Embora várias propostas, que não se caracterizam como indutivas, tenham sido contempladas, o sucesso e a popularidade de métodos automáticos de aprendizado se devem, sem dúvida alguma, aos chamados algoritmos de aprendizado indutivo de máquina. Revisões de algoritmos de aprendizado de máquina e de aprendizado indutivo de máquina podem ser vistas em várias publicações na literatura, tais como [Mitchell 1997] [Duda *et al*., 2001] [Bishop 2006] [Witten *et al*., 2011].

Um dos primeiros requisitos para a utilização de um sistema computacional que realiza aprendizado indutivo é dispor de um conjunto de dados, conhecido como *conjunto de treinamento*, que representa o conceito a ser aprendido. Cada dado (ou padrão) de um conjunto de treinamento é, geralmente, descrito por um vetor de atributos (i.e., um vetor de valores associados a atributos) e, dependendo da situação, de uma classe associada (que indica qual conceito o dado em questão representa). A classe de cada dado do conjunto de treinamento é, na maioria dos casos, determinada por um especialista humano da área de conhecimento à qual pertencem os dados. O fato de a classe participar da descrição do dado e do algoritmo de aprendizado fazer uso dessa informação caracteriza a técnica de aprendizado automático como de *aprendizado supervisionado* (ver [Mitchell 1997], [Witten *et al*., 2011]).

Em muitas situações do mundo real, entretanto, a classe à qual cada dado pertence é desconhecida e/ou não existe um especialista humano que, com base na descrição dos valores de atributos que descrevem os dados, seja capaz de determiná-la. Técnicas de aprendizado indutivo de máquina que lidam com conjuntos de dados que não têm uma classe associada são conhecidas como técnicas de *aprendizado não-*

supervisionado. Uma das técnicas de aprendizado não-supervisionado mais populares é chamada de agrupamento (*clustering*). O objetivo principal de algoritmos de agrupamento é particionar o conjunto de dados disponível em grupos, de acordo com as similaridades ou dissimilaridades entre tais dados. Como pode ser confirmado em pesquisa bibliográfica associada especificamente à agrupamentos, o número de algoritmos propostos na literatura é considerável (ver, por exemplo, [Theodoridis & Koutroumbas 2009], [Duda *et al.*, 2001], [Jain & Dubes 1988], [Jain 2008]).

1.2 Objetivo

O objetivo deste trabalho de pesquisa é a investigação empírica de algoritmos de agrupamento caracterizados como hierárquicos, particularmente aqueles que se enquadram na subcategoria de *hierárquicos aglomerativos*. Algoritmos hierárquicos produzem uma hierarquia de agrupamentos aninhados. Via de regra esses algoritmos envolvem N passos ou seja, tantos quantos forem os dados disponibilizados. A cada passo t um novo agrupamento é produzido usando, para isso, o agrupamento produzido no passo anterior (i.e., no passo $t-1$).

De interesse particular neste trabalho são os algoritmos de agrupamento aglomerativos (AA) baseados em conceitos da Teoria de Matrizes, com o objetivo de (1) identificação das principais características de conjuntos de padrões que promovem um bom desempenho deste tipo de algoritmo; (2) estudo, entre as várias alternativas existentes, para o cálculo de distâncias entre grupos.

1.3 Organização do Documento

Além deste capítulo inicial de introdução, o trabalho descreve, no Capítulo 2, conceitos relacionados à área de AIM, os principais tipos de aprendizado supervisionado, não-supervisionado e semi-supervisionado, mostra um exemplo concreto de uma situação de aprendizado de máquina, apresenta uma lista de algoritmos de agrupamento e descreve as etapas que envolvem o processo de agrupamento.

O Capítulo 3 apresenta os algoritmos hierárquicos aglomerativos de agrupamento baseados em teoria de matrizes, a notação adotada neste trabalho, a descrição e um *trace* de alto nível do esquema aglomerativo generalizado (EAG),

discussão sobre a complexidade computacional do EAG e um exemplo de agrupamento como método *Complete-Linkage*.

O Capítulo 4 apresenta e caracteriza os índices de validação que são comumente empregados para avaliar a qualidade dos agrupamentos produzidos pelos algoritmos de agrupamento e, também, faz uma breve descrição do algoritmo particionante K-Means, utilizado como *baseline* para comparação de resultados.

O Capítulo 5 apresenta uma descrição das funcionalidades do sistema computacional AggloCluster, ambiente utilizado para experimentação com os algoritmos aglomerativos que inclui, também, a implementação do algoritmo K-Means.

O Capítulo 6 descreve os experimentos feitos com conjuntos de padrões sintéticos *Sizes*, *Square* e *Aggregation*, e discute os resultados obtidos de tais experimentos.

O Capítulo 7 descreve e discute os experimentos realizados com conjuntos de padrões sintéticos com presença de *outliers*.

O Capítulo 8 descreve experimentos feitos com o conjunto de padrões sintéticos *Gestalt* e faz uma análise dos resultados obtidos.

O Capítulo 9, finalmente, conclui o trabalho com um resumo das principais atividades realizadas na pesquisa, comenta os resultados obtidos dos experimentos e sugere pontos a serem investigados como continuidade do trabalho desenvolvido e descrito nesta dissertação.

Capítulo 2

Aprendizado Indutivo de Máquina e Algoritmos de Agrupamento

Neste capítulo é apresentada a área de pesquisa de Aprendizado Indutivo de Máquina (AIM), com foco nas principais características gerais, objetivos, conceitos e modelos. O capítulo caracteriza as subáreas de (1) *aprendizado supervisionado*, (2) *aprendizado não-supervisionado* e (3) *aprendizado semi-supervisionado*. Apresenta conceitos e taxonomias dos algoritmos de agrupamento, faz uma breve descrição das etapas do processo de agrupamento e, ao final, lista alguns dos algoritmos de agrupamento hierárquico mais populares.

2.1 Aprendizado Indutivo de Máquina

Segundo [Mitchell 1997], os principais objetivos associados às pesquisas desenvolvidas na área de AIM são o desenvolvimento de técnicas computacionais que permitem simular o processo de aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de maneira automática. Como apontado em [Nicoletti 1994], existem inúmeras características que, de certa forma, diferenciam entre si os muitos algoritmos de AIM. Algumas das principais características são brevemente apresentadas a seguir:

- *Incrementabilidade*: diz respeito à maneira como os dados são apresentados ao algoritmo de AIM. Algoritmos considerados incrementais constroem a expressão do conceito dado a dado; tal expressão necessita de constante revisão por parte do algoritmo, uma vez que um novo dado pode causar, eventualmente, um rearranjo da expressão do conceito induzido até então. A expressão do conceito se modifica à medida que os dados se tornam disponíveis.
- *Não-Incrementabilidade*: algoritmos de AIM considerados não incrementais esperam que o conjunto de treinamento deva estar disponível desde o início do processo de aprendizado. A expressão do conceito induzida por tais algoritmos é feita considerando todos os dados de uma vez.

- *Um Conceito × Vários Conceitos*: o algoritmo de AIM é construído de tal forma a poder aprender uma ou várias expressões de conceitos de uma vez, na dependência dos dados do conjunto de treinamento.
- *Uso de Teoria de Domínio:* quando um algoritmo de AIM não possuir qualquer informação extra sobre o problema de aprendizado abordado, a indução da expressão do conceito é feita pelo algoritmo com base apenas no conjunto de treinamento disponibilizado. Em muitas situações do mundo real que requerem a solução de problemas de aprendizado complexos, é fundamental que conhecimento extra sobre o problema esteja disponível ao algoritmo, para auxiliar a indução do conceito. Esse conhecimento prévio é, via de regra, conhecido como *teoria de domínio*.
- *Linguagens de Descrição:* são, geralmente, linguagens formais que descrevem os dados, a teoria do domínio e as hipóteses formuladas. Os algoritmos mais populares de AIM empregam linguagens lógicas proposicionais que, se por um lado, simplificam o aprendizado, por outro, limitam a expressão do conceito às representações proposicionais. As soluções de problemas de aprendizado mais complexos, que exigem uma linguagem de representação mais elaborada, não podem ser encontradas por algoritmos proposicionais. Algoritmos mais sofisticados, que empregam representações baseadas em lógica de primeira ordem, por exemplo, devem ser usados em tais problemas.
- *Critérios de Avaliação do Conceito Induzido*: são critérios usados para medir a qualidade do conceito induzido por um algoritmo. Entre os critérios mais populares estão a precisão de classificação, a transparência da descrição induzida e a complexidade computacional. A precisão de classificação é medida, geralmente, com o percentual de exemplos classificados corretamente pela expressão induzida do conceito, a transparência da descrição é medida pelo número de descritores e operadores usados na descrição do conceito e, por fim, a complexidade computacional é relacionada aos recursos computacionais necessários para o aprendizado.

2.2 Aprendizado Supervisionado, Semi-supervisionado e Não-supervisionado

O processo de aquisição de conhecimento em AIM parte do específico para o geral, ou seja, induz generalizações que descrevem conceito(s) a partir de um conjunto de dados do(s) conceito(s). Neste contexto, os dados disponíveis para treinamento podem ter um impacto significativo no sucesso ou falha do mecanismo que realiza o aprendizado [Mitchell 1997].

Como mencionado no Capítulo 1, cada dado ou instância de um conjunto de treinamento é descrito por um vetor de atributos e, dependendo da situação, de uma classe associada. O fato de a classe, no conjunto de treinamento, participar da descrição do dado e do algoritmo de aprendizado fazer uso dessa informação caracteriza tal algoritmo como um algoritmo de *aprendizado supervisionado*. A classe, no caso, é o que se convenciona chamar de supervisão, uma vez que a informação da classe, associada a cada dado, não é coletada, como usualmente são os valores de atributos que descrevem os dados. A classe associada a cada dado é, geralmente, fornecida por especialistas humanos na área do domínio de dados.

Um algoritmo que implementa o aprendizado supervisionado generaliza o conjunto de dados que recebe como *input*. Dependendo do tipo do algoritmo e das técnicas empregadas, a generalização por ele realizada (que pode ser caracterizada como o resultado do aprendizado), é representada utilizando uma determinada linguagem de representação, tal como regras no padrão *if-then*, árvores de decisão, expressões em lógica proposicional, expressões em lógica de primeira ordem, redes neurais, etc. Via de regra o conceito induzido por algoritmos de AIM é conhecido como classificador; entretanto, existem algoritmos que não constroem classificadores – são os chamados algoritmos de regressão.

Muitas vezes o usuário necessita interpretar e compreender o classificador induzido por um sistema de aprendizado supervisionado. De acordo com [Michalski *et al.,* 1998], os sistemas de aprendizado podem ser classificados em duas categorias, considerando o grau com que tais sistemas são 'compreensíveis' ao ser humano:

- Sistemas caracterizados como *caixa-preta*, que desenvolvem sua própria representação do conceito e não fornecem explicações do processo de classificação (por exemplo, aqueles implementados por redes neurais); e

- Sistemas orientados a conhecimento, que objetivam a criação de estruturas simbólicas que sejam compreensíveis por humanos (por exemplo, aqueles que induzem regras de decisão).

Algoritmos que implementam o aprendizado supervisionado utilizam o conjunto de treinamento (em que cada dado é também descrito pela classe associada) para induzir uma expressão geral do conceito (representada via conjunto de regras, árvore de decisão, etc.). Tal expressão é, então, utilizada para determinar a classe associada a dados do mesmo domínio de conhecimento que, entretanto, têm classe desconhecida, como mostra o esquema geral da Figura 2.1.

Como apontado em [Nicoletti *et al*., 1998], em um sistema de aprendizado que faz uso de uma linguagem baseada em atributos para a representação dos dados e conceitos, uma tarefa de aprendizado supervisionado pode ser descrita como:

> *Dado um conjunto de exemplos de treinamento expressos como vetores de pares atributo-valor, cujas classes são conhecidas, encontrar uma regra que prediga a classe de um novo exemplo em função de seus atributos e valores.*

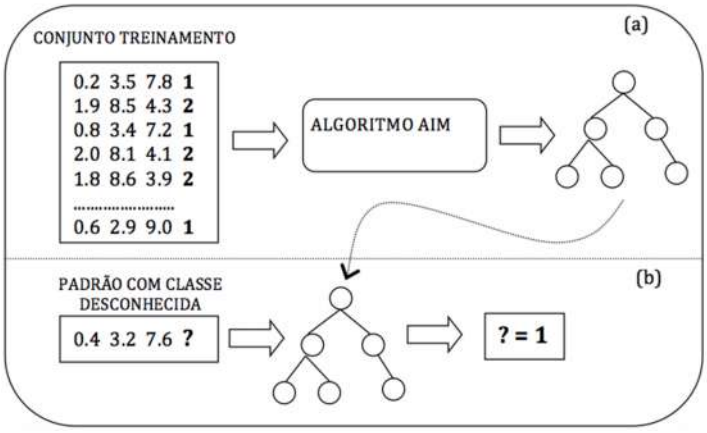


Figura 2.1 Esquema geral simplificado de aprendizado supervisionado. (a) processo de treinamento de um classificador (no caso uma árvore de decisão) e (b) seu uso na classificação de novos dados.

Em muitas situações do mundo real, entretanto, a classe à qual cada dado pertence é desconhecida e/ou não existe um especialista humano que, com base na descrição dos valores de atributos que descrevem os dados, seja capaz de determiná-la. Técnicas de aprendizado indutivo de máquina que lidam com conjunto de dados que não têm uma classe associada são conhecidas como técnicas de *aprendizado não-supervisionado*. Uma das técnicas de aprendizado não-supervisionado mais populares

é chamada de agrupamento (*clustering*). Algoritmos de agrupamento são discutidos brevemente na Seção 2.5.

Uma abordagem mais recente, conhecida como aprendizado *semi-supervisionado*, combina técnicas do aprendizado supervisionado e não-supervisionado que consistem em utilizar algoritmos de aprendizado para aprender a partir de dados classificados ou não. Em outras palavras, este modelo implementa a estratégia de continuamente tentar aprender mais dos dados não rotulados disponíveis, baseando-se inicialmente em um pequeno volume de dados rotulados [Chapelle *et al*., 2006].

2.3 Os Papéis dos Conjunto de Treinamento, de Teste e de Validação no AIM

Como apresentado na seção anterior, o *conjunto de treinamento* representa um conjunto de dados concretos (isto é, dados descritos por valores de seus atributos) do conceito a ser aprendido. Cada dado (ou *padrão*, termo este que será adotado na sequência deste trabalho, como referência a instância, dado, ponto, exemplo, etc.) de um conjunto de treinamento é, geralmente, descrito por um vetor de atributos (i.e., um vetor de valores associados a atributos) e, dependendo da situação, de uma classe associada (que indica qual conceito o padrão em questão representa). Dependendo da classe participar ou não da descrição dos padrões disponíveis, os algoritmos empregados para tratá-los têm características distintas.

O *conjunto de teste* tem um papel relevante para a avaliação da representatividade da expressão do conceito (e.g., árvore de decisão, conjunto de regras, etc.) que foi induzida por um algoritmo de AIM. O conjunto de teste é, geralmente, um subconjunto de padrões do conjunto original de padrões que não é utilizado para a indução do conceito, mas sim para avaliar o conceito induzido a partir dos outros padrões do conjunto de treinamento. Situações em que o classificador induzido esteja demasiadamente ajustado aos padrões de treinamento (i.e., expressa muito bem o conjunto de treinamento, mas falha na classificação de padrões do conjunto de teste) podem caracterizar um problema conhecido como *overfitting* [Han *et al*., 2012] [Zaki & Meira Jr 2014]. Uma das técnicas utilizadas para evitar o *overfitting* é conhecida como validação cruzada, que é apresentada na Seção 2.4 [Hastie *et al*. 2009], [Witten *et al*., 2011].

Uma forma de avaliar o classificador induzido pelo algoritmo de AIM é utilizar um subconjunto que não foi utilizado para o treinamento do classificador. Este subconjunto é conhecido como *conjunto de validação*, e é geralmente utilizado para inferir características de desempenho, tais como precisão, sensitividade e especificidade. A diferença entre o conjunto de testes e o conjunto de validação é que o conjunto de validação não é utilizado em nenhum momento da fase de treinamento, mas apenas para nortear as escolhas no processo de aprendizado [Abu-Mostafa *et al*., 2012].

2.4 Exemplo Didático de Aprendizado Indutivo Supervisionado

Considere o conjunto original de padrões (COP) contendo quarenta (40) padrões, artificialmente gerados, definidos por dois atributos numéricos (x e y) e uma classe (c) associada, mostrados na Tabela 2.1. Os 40 padrões estão distribuídos em três classes, a saber: 16 padrões são da classe 1, 13 da classe 2 e 11 da classe 3. Para facilitar a visualização, a representação dos padrões, no espaço bidimensional, está mostrada na Figura 2.2.

Tabela 2.1 Conjunto original de padrões (COP) artificialmente gerados e agrupados por classe.

x	y	c		x	y	c		x	y	c
2,6	4,5	1		1,3	2,2	2		5,3	2,9	3
3,0	4,2	1		1,6	2,5	2		5,4	2,5	3
3,2	4,6	1		1,5	2,1	2		5,3	3,4	3
3,1	4,8	1		1,8	1,8	2		5,6	3,3	3
3,4	4,2	1		1,6	2,4	2		5,7	3,0	3
3,4	4,6	1		1,7	2,1	2		5,6	2,6	3
3,5	5,0	1		1,9	2,0	2		6,0	3,5	3
3,7	4,7	1		2,3	2,7	2		6,3	3,3	3
3,6	4,3	1		2,3	2,3	2		6,1	2,9	3
3,8	4,1	1		2,4	1,8	2		6,2	2,5	3
4,0	4,1	1		2,6	2,6	2		5,6	3,2	3
3,9	4,6	1		2,5	1,8	2				
4,2	5,0	1		2,7	2,2	2				
4,4	4,6	1								
4,3	4,2	1								
4,7	4,6	1								

2.4.1 Usando o algoritmo J48 (um conjunto de treinamento e um de teste)

Para o exemplo de aprendizado indutivo supervisionado, serão utilizados o conjunto COP (cujos padrões se encontram misturados, com relação à classe) e o algoritmo J48 (que é uma versão do algoritmo C4.5 [Quinlan, 1993]), disponibilizado pelo ambiente Weka (versão 3.6.12). Inicialmente um subconjunto contendo 8 padrões foi retirado do conjunto de 40 padrões e reservado, para ser usado posteriormente no processo de teste da árvore induzida pelo J48; o conjunto restante, de 32 padrões, foi então utilizado como conjunto de treinamento. O conjunto de teste está apresentado na Figura 2.3, descrito no padrão adotado pelo Weka, ou seja, o arquivo deve estar no formato ARFF (*Attribute-Relation File Format*), que é um arquivo em caracteres ASCII.

O formato do arquivo ARFF segue o seguinte padrão:

- Nome da relação seguido do marcador @relation
- Nome dos atributos, do tipo numérico, seguidos do marcador @attribute. Note que os valores do atributo (c) estão restritos ao conjunto {1,2,3}.
- Os padrões ficam abaixo do marcador @data, e os valores dos atributos de cada padrão são separados por vírgula. O Weka, dada a sua configuração padrão, assume que o último valor que descreve um padrão de treinamento representa a classe do padrão em questão.

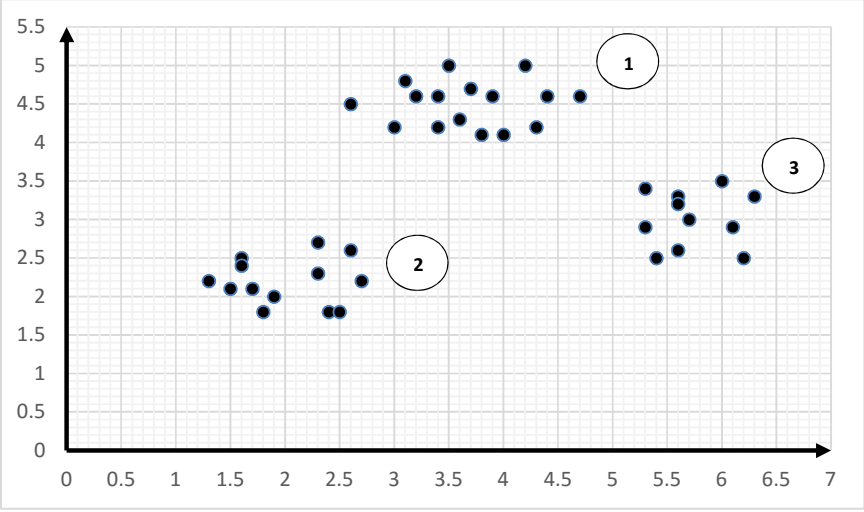


Figura 2.2 Representação dos 40 padrões do conjunto COP no plano cartesiano.

```
1  @relation COPTeste
2  @attribute x numeric
3  @attribute y numeric
4  @attribute c {1,2,3}
5  @data
6  2.6,2.6,2
7  5.6,3.3,3
8  4.4,4.6,1
9  5.7,3.0,3
10 1.8,1.8,2
11 3.0,4.2,1
12 6.1,2.9,3
13 3.8,4.1,1
```

Figura 2.3 Conjunto de teste com 8 padrões, extraído do COP, previamente ao treinamento.

Na sequência são descritos os passos para induzir uma árvore de decisão a partir do conjunto de treinamento. Os passos 1 e 2 correspondem às etapas de pré-processamento dos padrões, seleção do mecanismo indutor e indução do classificador; o passo 3 a etapa de avaliação do classificador.

Passo 1: o arquivo contendo os dados de treinamento é aberto no painel *Preprocess* do Weka Explorer. Na Figura 2.4 é apresentada a tela de pré-processamento na qual é possível visualizar, entre outras informações, a distribuição dos padrões do conjunto de treinamento inicial, contendo 32 padrões.

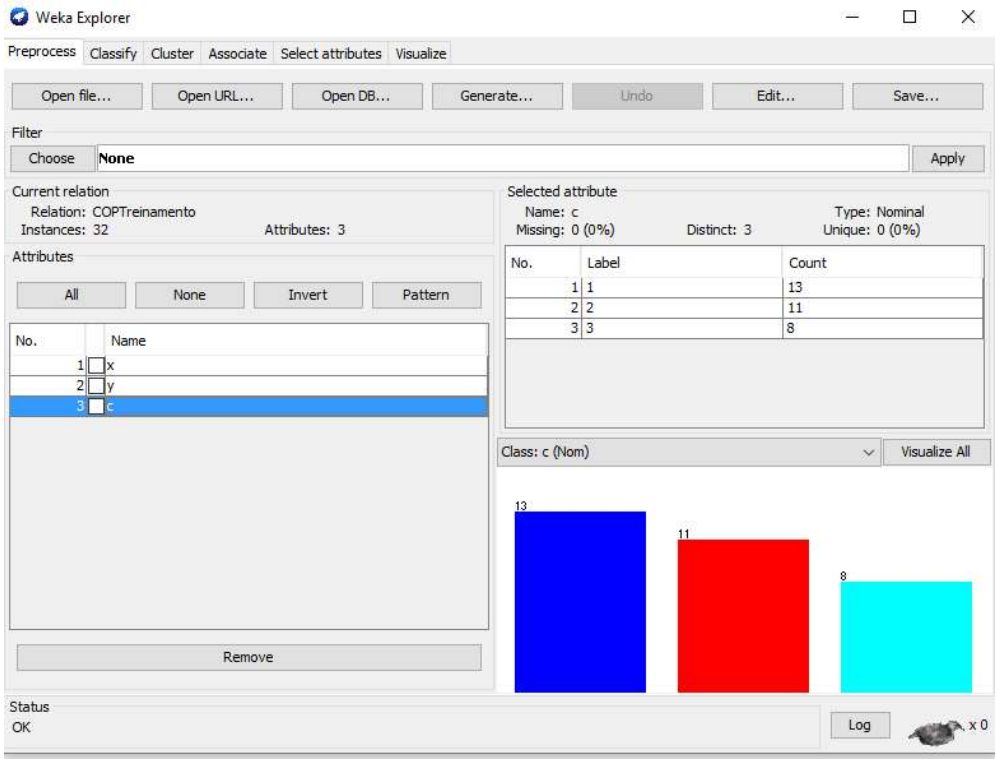


Figura 2.4 Tela de Pré-processamento do Weka Explorer.

No histograma (canto inferior à direita) da Figura 2.4 são apresentados, da esquerda para a direita, 13 padrões com rótulo 1, 11 padrões com rótulo 2 e 8 padrões com rótulo 3, que constituem o conjunto de treinamento informado ao Weka.

Passo 2: no painel *Classify*, conforme mostrado na Figura 2.6, o classificador (*classifier*) J48 é selecionado. Em *Test options*, é marcado *Use training set* e depois o botão *Start* é acionado. Após a execução, a árvore de decisão induzida é apresentada, como na Figura 2.5, e o sistema também fornece informações adicionais via a matriz de confusão (sumário da classificação dos padrões de treinamento, feita pela árvore induzida). As saídas produzidas pelo Weka e apresentadas no trabalho, mantiveram a nomenclatura adotada pelo software.

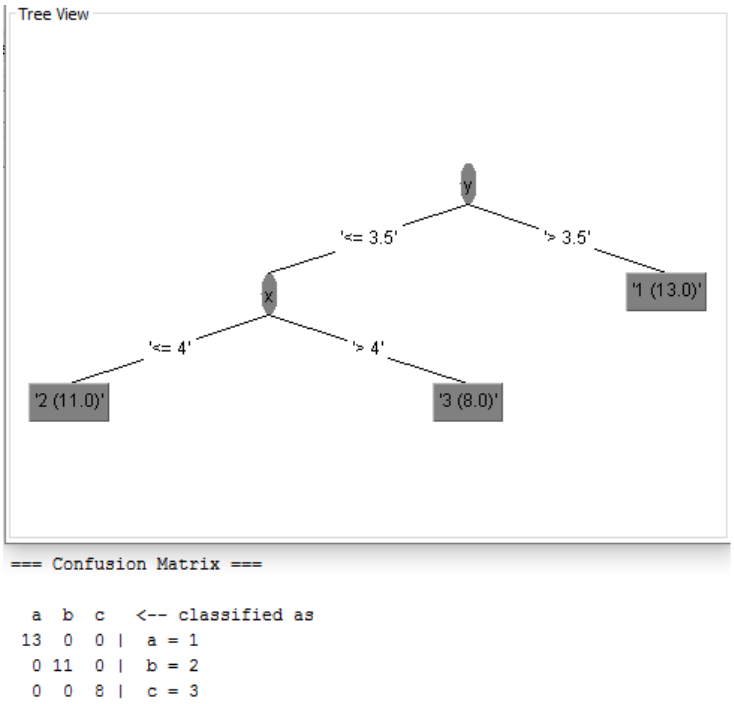


Figura 2.5 Árvore de decisão e Matriz de Confusão.

Uma breve análise da matriz de confusão informada pelo sistema evidencia que a árvore gerada a partir dos 32 padrões está perfeitamente ajustada a eles, considerando que a matriz informa que os 13 padrões de classe 1 foram classificados como de classe 1, os 11 padrões de classe 2 foram classificados como de classe 2 e os 8 de classe 3 foram classificados pela árvore como de classe 3.

No Passo 3 é apresentado o processo de classificação em um conjunto de teste (no caso, aquele descrito na Figura 2.3). O propósito de utilizar o conjunto de teste é

verificar como os padrões são classificados e se o classificador induzido expressa, de forma precisa, o conjunto de treinamento.

Passo 3: no painel *Classify*, com o classificador J48 previamente selecionado no passo anterior, o botão *Supplied test set* é acionado para selecionar o arquivo contendo o conjunto de testes. Na sequência, o botão *Start* é novamente acionado.

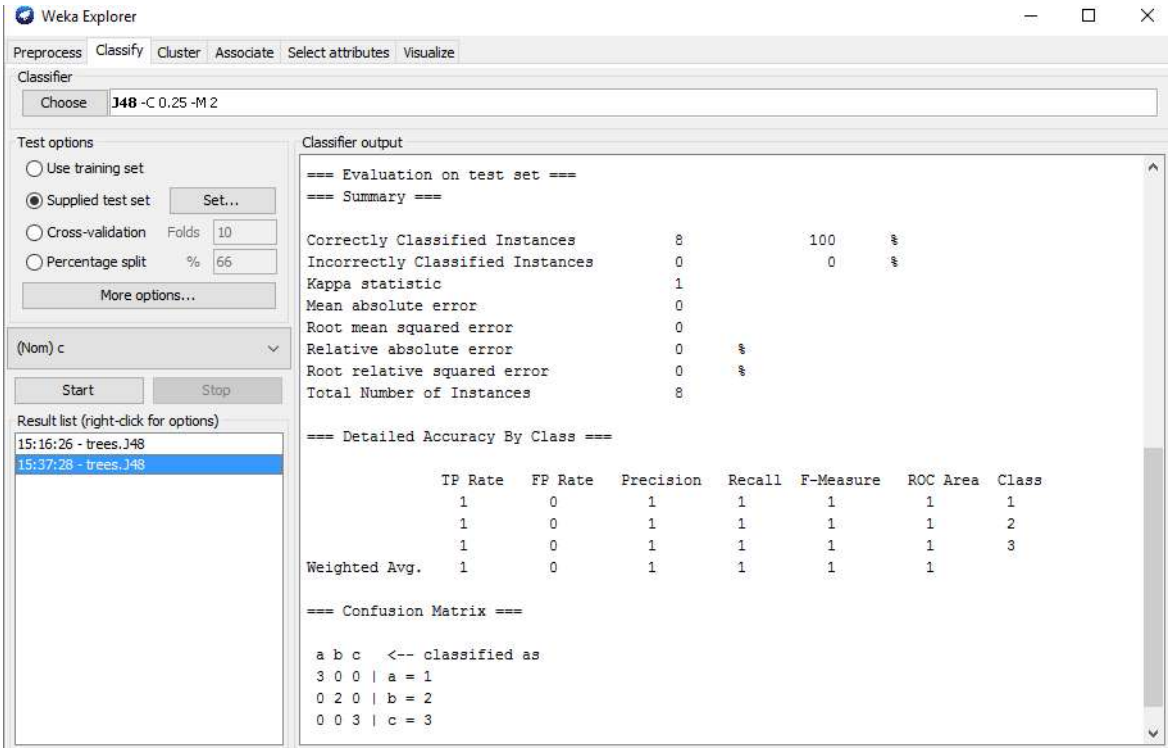


Figura 2.6 Relatório de avaliação do conjunto de teste e Matriz de Confusão.

Como pode ser visto na Figura 2.6 os 8 padrões do conjunto de teste foram classificados corretamente, comprovando que a árvore induzida expressa corretamente o conjunto de treinamento.

2.4.2 Simulando um Procedimento de K-Validação Cruzada

Uma k-validação cruzada (*k cross-validation*) é uma técnica utilizada em AIM que divide o conjunto de padrões em k partes, com o objetivo de evitar ou reduzir o risco de *overfitting*.

O Weka disponibiliza automaticamente a indução de classificadores (no caso via o J48) utilizando um processo de k-validação cruzada, sendo que o usuário do sistema informa o valor de k. Por exemplo, para um valor de k igual a 4, o sistema automaticamente divide o conjunto inicial de padrões em 4 partes e executa 4 processos indutivos, em cada um utilizando uma combinação de 3 partes diferentes e,

a parte que não foi utilizada para a indução do conceito, é assumida como o conjunto de teste. No painel *Classify* do Weka existe a opção de teste com validação cruzada, no qual deve ser informado o número de *folds* (grupos de padrões a serem formados a partir do conjunto inicial de padrões fornecido). Para o exemplo que segue, entretanto, para tornar o processo mais claro, uma 4-validação cruzada foi simulada manualmente. Para a simulação de um processo de 4-validação cruzada, os 40 padrões do COP foram divididos em 4 partes iguais resultando 4 subconjuntos com 10 padrões cada, como evidenciado na Tabela 2.2 (conjuntos A, B, C e D).

Tabela 2.2 Conjuntos de Treinamento e Teste {A, B, C, D}.

A			B			C			D		
x	y	c	x	y	c	x	y	c	x	y	c
3,7	4,7	1	3,9	4,6	1	4,3	4,2	1	3,2	4,6	1
6,1	2,9	3	5,6	2,6	3	2,6	2,6	2	6,0	3,5	3
3,4	4,6	1	4,4	4,6	1	3,4	4,2	1	1,6	2,5	2
2,3	2,3	2	1,3	2,2	2	2,7	2,2	2	4,7	4,6	1
3,5	5,0	1	2,6	4,5	1	5,3	2,9	3	1,5	2,1	2
1,9	2,0	2	1,8	1,8	2	3,0	4,2	1	6,2	2,5	3
3,6	4,3	1	5,4	2,5	3	5,3	3,4	3	3,1	4,8	1
2,3	2,7	2	1,6	2,4	2	5,6	3,3	3	5,6	3,2	3
4,0	4,1	1	3,8	4,1	1	2,4	1,8	2	4,2	5,0	1
6,3	3,3	3	1,7	2,1	2	5,7	3,0	3	2,5	1,8	2

O processo de validação cruzada conduzido foi realizado como mostra a Tabela 2.3, em que quatro processos de indução de conceito foram realizados. Na Indução I_1, por exemplo, a árvore de decisão foi construída com os conjuntos de padrões A+B+C e a árvore induzida foi avaliada usando o conjunto de teste D.

Tabela 2.3 Simulação de um processo de 4-validação cruzada.

Indução	Conjunto de Treinamento	Conjunto de Teste	Número de classificações corretas	Número de classificações incorretas
I_1	A+B+C	D	9	1
I_2	A+B+D	C	10	0
I_3	A+C+D	B	10	0
I_4	B+C+D	A	10	0

A Figura 2.7 mostra o resultado do processo de indução e teste da árvore induzida relativa ao processo indutivo I_1.

```
J48 pruned tree
------------------

y <= 3.4
|   x <= 4: 2 (10.0)
|   x > 4: 3 (8.0)
y > 3.4: 1 (12.0)

Number of Leaves  :     3

Size of the tree :      5


Time taken to build model: 0.02 seconds

=== Predictions on test split ===

inst#,     actual, predicted, error, probability distribution
     1        1:1        1:1          *1      0      0
     2        3:3        1:1        +  *1      0      0
     3        2:2        2:2           0     *1      0
     4        1:1        1:1          *1      0      0
     5        2:2        2:2           0     *1      0
     6        3:3        3:3           0      0     *1
     7        1:1        1:1          *1      0      0
     8        3:3        3:3           0      0     *1
     9        1:1        1:1          *1      0      0
    10        2:2        2:2           0     *1      0
```

Figura 2.7 Relatório do J48 relativo ao processo indutivo I_1: treinamento: A+B+C e Teste: D.

Na Figura 2.7 é mostrado o relatório do Weka contendo a árvore (J48 *pruned tree*) e a classificação (*Predictions on test split*) dos 10 padrões do conjunto de teste D. A coluna *actual* lista as classes informadas no conjunto de teste e a coluna *predicted* os valores classificados de acordo com as regras da árvore induzida pelo J48. Observe que o padrão 2, em destaque, foi classificado incorretamente; de acordo com a árvore induzida pelo processo de indução I_1, o padrão cujo valor do atributo y for maior que 3,4 (caso do padrão em questão, sendo x = 6,0 e y = 3,5) é classificado como de classe 1, diferente do que foi informado no conjunto de teste (i.e., classe 3).

Na sequência, os conjuntos A, B e D são utilizados para treinamento e o conjunto C para teste, no processo indutivo I_2. A Figura 2.8 mostra o resultado obtido quando da avaliação da árvore de decisão gerada pelo J48, usando como teste o conjunto C (como especifica I_2).

```
J48 pruned tree
------------------

y <= 3.5
|   x <= 3.9: 2 (10.0)
|   x > 3.9: 3 (7.0)
y > 3.5: 1 (13.0)

Number of Leaves  :     3

Size of the tree :      5


Time taken to build model: 0 seconds

=== Predictions on test split ===

inst#,     actual, predicted, error, probability distribution
     1        1:1        1:1         *1      0      0
     2        2:2        2:2          0     *1      0
     3        1:1        1:1         *1      0      0
     4        2:2        2:2          0     *1      0
     5        3:3        3:3          0      0     *1
     6        1:1        1:1         *1      0      0
     7        3:3        3:3          0      0     *1
     8        3:3        3:3          0      0     *1
     9        2:2        2:2          0     *1      0
    10        3:3        3:3          0      0     *1
```

Figura 2.8 Relatório do J48 relativo ao processo indutivo I_2: treinamento: A+B+D e Teste: C.

Conforme relatório mostrado na Figura 2.8, todos os padrões do conjunto de teste C foram classificados corretamente, ou seja, os valores listados na coluna *actual* coincidem com os valores listados na coluna *predicted*.

Continuando, no processo indutivo I_3 os conjuntos A, C e D são utilizados para treinamento e o conjunto B para teste; a árvore induzida pelo J48 teve o desempenho mostrado na Figura 2.9.

```
J48 pruned tree
------------------

y <= 3.5
|   x <= 4: 2 (9.0)
|   x > 4: 3 (9.0)
y > 3.5: 1 (12.0)

Number of Leaves  :     3

Size of the tree :      5


Time taken to build model: 0 seconds

=== Predictions on test split ===

inst#,     actual, predicted, error, probability distribution
     1        1:1        1:1          *1      0      0
     2        3:3        3:3           0      0     *1
     3        1:1        1:1          *1      0      0
     4        2:2        2:2           0     *1      0
     5        1:1        1:1          *1      0      0
     6        2:2        2:2           0     *1      0
     7        3:3        3:3           0      0     *1
     8        2:2        2:2           0     *1      0
     9        1:1        1:1          *1      0      0
    10        2:2        2:2           0     *1      0
```

Figura 2.9 Relatório do J48 relativo ao processo indutivo I_3: treinamento: A+C+D e Teste: B.

Conforme relatório mostrado na Figura 2.9, todo os padrões do conjunto de teste B foram classificados corretamente. Entretanto, note que a árvore induzida neste processo (I_3) é diferente da que foi construída no processo indutivo I_2.

Finalmente, de acordo com as especificações do processo indutivo I_4, o J48 induz a árvore com os padrões em B+C+D e testa a árvore obtida usando os padrões em A; o relatório do J48 com relação a esse processo indutivo está mostrado na Figura 2.10.

```
J48 pruned tree
------------------

y <= 3.5
|   x <= 3.9: 2 (10.0)
|   x > 3.9: 3 (9.0)
y > 3.5: 1 (11.0)

Number of Leaves  :     3

Size of the tree :      5


Time taken to build model: 0 seconds

=== Predictions on test split ===

inst#,     actual, predicted, error, probability distribution
     1        1:1        1:1          *1      0      0
     2        3:3        3:3           0      0     *1
     3        1:1        1:1          *1      0      0
     4        2:2        2:2           0     *1      0
     5        1:1        1:1          *1      0      0
     6        2:2        2:2           0     *1      0
     7        1:1        1:1          *1      0      0
     8        2:2        2:2           0     *1      0
     9        1:1        1:1          *1      0      0
    10        3:3        3:3           0      0     *1
```

Figura 2.10 Relatório do J48 relativo ao processo indutivo I_4: treinamento: A+B+D e Teste: C.

O relatório apresentado na Figura 2.10 relativo ao processo indutivo I_4, mostra que os 10 padrões do conjunto de teste C foram classificados corretamente. Note que a árvore induzida neste processo é equivalente a que foi induzida no processo I_2.

A etapa final do processo de k-validação consiste em analisar o desempenho médio do classificador nos k testes. A Tabela 2.4 mostra o sumário da 4-validação realizada com os 40 padrões do COP com informações relativas ao classificador induzido no processo I_3.

Tabela 2.4 Sumário do processo de 4-validação.

Número de classificações corretas	39	97,5%
Número de classificações incorretas	1	2,5%
Erro médio absoluto	0,0167	
Número total de padrões	40	

2.5 Algoritmos de Agrupamento – Conceitos e Taxonomias

Algoritmos de agrupamento (*clustering*), no contexto de aprendizado de máquina e mineração de dados, são utilizados principalmente em tarefas de exploração e extração de padrões. O aprendizado de máquina pode ser visto, neste sentido, como uma caixa de ferramentas (algoritmos de agrupamento) utilizada nas tarefas de mineração de dados. A aplicação de técnicas de agrupamentos para segmentação de mercados (por exemplo, identificar grupos de pessoas que sejam mais receptivas a uma forma específica de propaganda) é muito comum. Como evidenciado em [Gan *et al*., 2007] e [James *et al*., 2013], entretanto, a área de bioinformática (por exemplo, a detecção e extração de características em expressões gênicas) está se tornando, cada vez mais, uma área típica para o uso de algoritmos de agrupamento.

O agrupamento pode ser entendido como um tipo de classificação imposta em um conjunto finito de padrões, em que o relacionamento entre eles é representado em uma matriz de proximidade [Jain & Dubes 1988]. A proximidade, neste contexto, pode ser a distância euclidiana entre pares de padrões em um espaço l-dimensional.

Na taxonomia apresentada em [Jain & Dubes 1988], o agrupamento é um tipo especial de classificação intrínseca (não-supervisionada) pois utiliza apenas a matriz de proximidade para realizar a classificação. Esta taxonomia está organizada da seguinte forma:

- Exclusiva *versus* não-exclusiva: Uma classificação exclusiva representa uma partição do conjunto de padrões, em que cada padrão pertence exclusivamente a um subconjunto, ou grupo. Já a não-exclusiva aceita sobreposição, uma vez que um mesmo padrão pode pertencer a diferentes grupos.
- Intrínseca *versus* extrínseca: Como mencionado anteriormente, uma classificação intrínseca utiliza somente a matriz de proximidade para realizar a classificação. Por outro lado, a classificação extrínseca utiliza-se de informações dos padrões, tais como rótulos ou classes, bem como a matriz de proximidade.
- Hierárquico *versus* particional: A classificação exclusiva e intrínseca é subdividida em classificação hierárquica e particional de acordo com o

tipo da estrutura dos padrões. Uma classificação hierárquica é uma sequência aninhada de partições, enquanto que a particional faz a classificação em uma única partição. Algoritmos particionantes podem ser abordados formalmente como: dados N padrões em um espaço métrico *d*-dimensional, determinar uma partição dos padrões em K grupos de maneira que padrões em um mesmo grupo sejam mais semelhantes entre si do que quando comparados com padrões pertencentes a outros grupos. No contexto desta dissertação, o termo *agrupamento hierárquico* é utilizado no lugar de classificação hierárquica. A apresentação formal do método de agrupamento hierárquico é feita no Capítulo 3.

Vários foram os algoritmos propostos para expressar os métodos de classificação apresentados anteriormente. Os principais métodos de agrupamento, ou algoritmos, mais utilizados são [Jain & Dubes 1988]:

- Aglomerativo *versus* divisivo. Ambos realizam agrupamento hierárquico. Um algoritmo aglomerativo inicia o agrupamento com cada padrão formando um grupo unitário, e gradualmente, faz a fusão de grupos em grupos maiores. Os algoritmos divisivos fazem o processo reverso (i.e., iniciam com todos os padrões em um único grupo e o subdivide em grupos menores).
- Sequencial *versus* simultâneo: No método sequencial, cada padrão é tratado um-a-um e não todos de uma vez como ocorre no método simultâneo.
- Monotético *versus* politético: O termo monotético significa que apenas um critério diferenciador é utilizado no processo de agrupamento, o politético, mais de um critério. Esta abordagem tem mais aplicações em problemas de taxonomia, onde os objetos a serem agrupados são representados como padrões, ou pontos em um espaço. O algoritmo de agrupamento monotético utiliza das características (ou atributos) dos padrões uma-a-uma, diferentemente do politético que as utiliza todas de uma vez.
- Baseados na Teoria dos Grafos *versus* Álgebra de Matrizes: Alguns algoritmos são expressados em termos de teoria dos grafos, utilizando propriedades tais como conectividade para definir os grupos de um

agrupamento. Outros algoritmos expressam em termos de construções algébricas, tal como erro quadrático médio.

Na taxonomia proposta por [Theodoridis & Koutroumbas 2009], algumas das categorias apresentadas anteriormente estão reorganizadas, mas de forma mais ampla. Nela os algoritmos são divididos nas seguintes categorias:

- Algoritmos sequenciais: Produzem um único agrupamento, mas o resultado final depende da ordem na qual os padrões são passados para o algoritmo. Este esquema de agrupamento tende a produzir grupos compactos, hiperesféricos ou hiperelipsoidais.
- Algoritmos de agrupamento hierárquico: Constroem uma hierarquia de agrupamentos (i.e., uma árvore de agrupamentos conhecida como dendrograma). Nessa estrutura, cada agrupamento pode conter outros agrupamentos, denominados filhos; se um agrupamento não tiver nenhum filho, ele é denominado uma folha do dendrograma. Esta categoria de algoritmos divide-se em:
    - Algoritmos de agrupamento aglomerativos – Produzem uma sequência de agrupamentos de um número decrescente de grupos (*clusters*), *m*, em cada passo. Os principais algoritmos aglomerativos que representam esta categoria são o *single linkage*, *complete linkage* e o *average linkage*. Os algoritmos aglomerativos podem ser divididos nas seguintes subcategorias:
        - Algoritmos baseados em conceitos da Teoria de Matrizes (ênfase desta dissertação);
        - Algoritmos baseados em conceitos da Teoria dos Grafos;
    - Algoritmos de agrupamento divisivos – Atuam na direção oposta à dos algoritmos de agrupamento aglomerativos, i.e., produzem uma sequência de agrupamentos com um número crescente de grupos a cada passo.
- Algoritmos de agrupamento baseados na otimização da função de custo – Os algoritmos desta categoria são chamados também de *esquemas de otimização de função iterativa* e são consideradas as seguintes subcategorias:

- Algoritmos de agrupamento *hard* ou *crisp* – um padrão pertence exclusivamente a um grupo específico.
- Algoritmos de agrupamento probabilístico – baseados na regra de Bayes.
- Algoritmos de agrupamento *Fuzzy* – um padrão pode pertencer a um grupo específico com um determinado grau, o que permite a pertinência de um mesmo padrão a vários grupos de um agrupamento.
- Algoritmos de detecção de limites – Ao invés de determinar os agrupamentos pelo vetor de características, estes algoritmos ajustam, de forma iterativa, os limites das regiões que delimitam os agrupamentos.

Uma das razões que justifica investigar, nesta dissertação, os métodos aglomerativos no lugar dos divisivos baseia-se no fato de que esta segunda abordagem encontra alguns desafios quando do particionamento de um grupo grande em subgrupos menores. Por exemplo, existem 2^{N-1} -1 maneiras possíveis de particionar um conjunto de N padrões em dois subconjuntos (disjuntos). Quando o valor de N é grande (algumas centenas), torna-se computacionalmente proibitivo examinar todas as possibilidades. Por consequência o método divisivo normalmente faz uso de heurísticas de particionamento, o que pode levar a resultados imprecisos. Quanto à eficiência, o método divisivo não disponibiliza uma forma de rastrear as decisões de particionamento realizadas nos primeiros passos do processo de agrupamento. Uma vez particionado, qualquer possibilidade alternativa de agrupamento não é considerada novamente. No geral, a complexidade dos algoritmos divisivos cresce exponencialmente em relação à N. Por outro lado os algoritmos aglomerativos são, no pior cenário, cúbicos em relação à N e por isto, na prática, são factíveis computacionalmente [Han *et al*. 2011]. Em razão dos desafios associados aos métodos divisivos, existe uma quantidade substancialmente maior de métodos aglomerativos disponíveis na literatura.

2.6 Etapas do Processo de Agrupamento

O processo de agrupamento pode ser dividido, geralmente, em quatro etapas: (1) pré-processamento dos padrões; (2) escolha da medida de similaridade e execução do algoritmo de agrupamento; (3) validação dos resultados; (4) interpretação dos grupos identificados.

O pré-processamento dos dados é uma etapa fundamental no processo de agrupamento. A preparação e transformação dos dados devem ser realizadas a fim de proporcionar um bom desempenho dos algoritmos de agrupamento e evitar que características de padrões que tenham valores não normalizados, i.e., em escalas diferentes, dominem os resultados de alguma função de dissimilaridade (e.g., distância Euclidiana). Apesar do termo pré-processamento se aplicar à várias possíveis técnicas que permitem tratar dados inconsistentes, dados com ruído, dados com valores de atributos ausentes, etc., no contexto desse trabalho, o pré-processamento está voltado essencialmente à normalização dos dados, assumindo que os problemas anteriores já foram tratados. Os algoritmos de agrupamento são influenciados pela escala dos atributos dos padrões e muitas vezes requer algum tipo de normalização, considerando a medida de dissimilaridade a ser utilizada. A técnica conhecida como *z-score* permite realizar a transformação dos dados por meio da padronização [Jain & Dubes 1988] ou padronização *z-score* [Berthold *et al*., 2010]. Cada atributo é padronizado conforme a Eq. (2.1).

$$zA = \frac{A - \mu}{\sigma} \qquad \text{Eq. (2.1).}$$

em que A representa o valor original do atributo do padrão a ser normalizado, μ a média dos valores associados ao atributo em questão, σ o desvio padrão do atributo e zA o correspondente valor normalizado. Outra técnica disponível para transformação dos dados que realiza a normalização dos valores de atributos trazendo-os para uma determinada faixa (*feature scaling*) é conhecida como *Min-Max* [Berthold *et al.* 2010]. Esta técnica faz o escalonamento dos atributos trazendo seus valores para um intervalo [0, 1], em que o menor valor é alterado para 0 e o maior para 1. O cálculo da distância Euclidiana, função de dissimilaridade utilizada pelos algoritmos de agrupamento investigados nesta pesquisa, é sensível a diferenças de escalas nos atributos. A transformação dos valores por meio das técnicas citadas anteriormente

tende a diminuir tal sensibilidade da função de dissimilaridade e aumentar o desempenho (precisão) do algoritmo. O método de normalização a ser utilizado dependerá das características do conjunto de padrões e da presença ou não de ruídos (veja [Milligan & Cooper 1988]).

A escolha da medida de similaridade utilizada no processo de agrupamento deve ser feita considerando as características do conjunto de padrões, como, por exemplo, o tipo e escala dos atributos. Várias medidas para o cálculo da similaridade foram propostas, entre as quais estão as medidas de distância, correlação e associação. Entre as diversas medidas de distância podemos destacar a distância Manhattan, Euclidiana, Chebyshew, Mahalanobis e Minkowsky. No escopo dos algoritmos de agrupamento investigados nesta pesquisa, apenas a distância Euclidiana é utilizada como função de dissimilaridade. Uma descrição formal da distância Euclidiana é apresentada no Capítulo 3. Uma revisão dos vários tipos de medidas de similaridade pode ser encontrada em [Jain & Dubes 1988] e [Everitt *et. al*., 2011].

Assim como a escolha da medida de similaridade, a escolha do algoritmo de agrupamento deve ser realizada considerando características ou o domínio do conjunto de padrões. Na escolha do algoritmo de agrupamento é preciso levar em consideração qual tipo de agrupamento é esperado como resultado. Como discutido na Seção 2.5, diferentes algoritmos produzem diferentes tipos de agrupamento não existindo, portanto, um algoritmo ótimo para todos os casos. Segundo [Berkhin 2002], algumas características devem ser levadas em consideração na escolha do algoritmo de agrupamento tais como: tipos de entrada (matriz de dissimilaridade ou matriz de padrões); tipos de atributos; escalabilidade (para conjuntos de padrões grandes); grau de dimensionalidade dos conjuntos de padrões; habilidade para encontrar grupos com diferentes formas; tolerância a *outliers*; complexidade computacional de tempo e espaço; dependência da ordem de entrada dos dados; tipos de agrupamento (e.g., partição ou hierarquia); dificuldade em determinar os parâmetros de entrada do algoritmo e forma de apresentação dos resultados.

A etapa de avaliação ou validação dos grupos identificados pelo algoritmo de agrupamento consiste em determinar o grau de significância dos resultados obtidos. A validação de agrupamentos, normalmente, é realizada com base em índices estatísticos que, de forma quantitativa, apontam a qualidade dos grupos encontrados no conjunto de padrões. Estes índices de validação são discutidos no Capítulo 4.

A interpretação dos grupos identificados pelo algoritmo de agrupamento é uma etapa em que é necessária, em muitos casos, a participação do especialista do domínio. Os agrupamento resultantes da execução dos algoritmos de agrupamento não apresentam descrições conceituais simples, mas sim uma descrição por meio de valores estatísticos e índices de qualidade que limitam ou dificultam a extração dos conceitos descritos pelos grupos identificados pelo algoritmo.

2.7 Algoritmos Hierárquicos

No que segue, alguns dos algoritmos de agrupamento hierárquico aglomerativo e divisivo mais populares, encontrados na literatura, são brevemente abordados.

- AGNES (*AGglomerative NESting*) – É um algoritmo aglomerativo proposto por [Kaufman & Rousseeuw 2005]. Foi projetado de modo que não seja necessário informar o número de grupos. A construção de um dendrograma, representação hierárquica de agrupamento, permite derivar todos os possíveis agrupamentos gerados pelo algoritmo.

  A matriz de dissimilaridade ou o conjunto de treinamento são os dois *inputs* do AGNES cuja implementação contempla duas métricas de similaridade (distância Euclidiana e a distância Manhattan).

  O método de agrupamento disponível no AGNES é o *average linkage* (também conhecido como *UPGMA*). Muitos do algoritmos aglomerativos mais atuais, como os descritos em [Kang & Landry, 2015] e [Tamura & Miyamoto, 2014], são extensões do AGNES. O conceito de agrupamento aglomerativo, métodos para cálculo da dissimilaridade e as diferentes técnicas de agrupamentos são discutidos no Capítulo 3.
- BIRCH (*Balanced Iterative Reducing and Clustering Using Hierarchies*) – Foi proposto tendo em vista a necessidade de agrupar grandes volumes de padrões de forma eficiente (i.e., com acesso mínimo ao disco e maior uso da memória principal). Para atingir este objetivo o algoritmo faz uma compactação do conjunto de treinamento em subconjuntos, permitindo assim que o processo de agrupamento ocorra na memória principal com uma única leitura dos dados. Mesmo com uma proposta de uso eficiente do hardware, algumas deficiências foram constatadas como: desempenho reduzido quando

os grupos ou agrupamentos não são uniformes no que diz respeito a tamanhos e formatos; adequado apenas para conjuntos numéricos e pontos no espaço Euclidiano. Para uma análise mais aprofundada do algoritmo ver [Zhang *et al*., 1996].

- CURE (*Clustering Using Representatives*) – É um algoritmo indicado também para grandes volumes de padrões. O algoritmo inicia selecionando aleatoriamente um subconjunto dos padrões (amostra) e o particiona em novos subconjuntos resultando em um agrupamento parcial de cada partição. Na sequência, utiliza o representante de cada grupo da partição para realizar o agrupamento dos dados restantes.

  De forma simplificada, o CURE executa os seguintes passos: (1) seleção aleatória; (2) particionamento; (3) agrupamento parcial das partições; (4) eliminação dos *outliers*; (4) agrupamento parcial dos grupos, (5) agrupamento dos padrões restantes. O tempo de execução do algoritmo assim como sua complexidade computacional foram relatados em [Guha *et al*., 1998].
- CHAMELON (*A Hierarchical Clustering Algorithm Using Dynamic Modeling*) – Este algoritmo, diferente do BIRCH e CURE, não segue um modelo estático em seu processo de agrupamento, ao invés disso mede a similaridade entre dois grupos baseando-se em modelos dinâmicos. Para quantificar a similaridade entre dois grupos, os conceitos de *interconectividade relativa* e *proximidade relativa* devem estar definidos. Em outras palavras, dois grupos são agrupados se, e somente se, a interconectividade e a proximidade entre dois grupos são comparáveis à interconectividade interna dos grupos e a proximidade entre os padrões dentro desses grupos (ver [Karypis *et al*., 1999], [Theodoridis & Koutroumbas 2009]).
- ROCK (*RObust Clustering using LinKs*) – A proposta deste algoritmo é sugerir uma alternativa para agrupamentos de padrões com atributos categóricos (escala nominal). No lugar da métrica de distância Euclidiana, são utilizadas alternativas como a distância de *Hamming* ou a VDM (*Value Difference Metric*). Neste contexto, o conjunto de padrões é analisado como um grafo esparso, onde padrões são os vértices do grafo e a aresta entre dois padrões indica que fazem parte de um mesmo subgrafo (grupo). Partindo do

princípio que o número de ligações entre um par de padrões é o número de vizinhos comuns destes padrões, o algoritmo analisa o grau de conectividade dos grupos e seleciona o par que maximiza a soma das ligações [Guha *et al*., 2000].

- MONA (*Monothetic Analysis*) – Muitas são as formas de agrupar padrões com valores de atributos binários. A ideia básica deste algoritmo é selecionar um destes atributos e dividir o conjunto de padrões a partir de um deles, de forma que cada parte esteja relacionada a cada atributo. Neste sentido, dois subconjuntos são formados e, em cada um deles, um dos atributos restantes é selecionado e utilizado da mesma forma para dividir o subconjunto em dois grupos menores [Kaufman & Rousseeuw 2005]. Este algoritmo implementa uma estratégia de agrupamento hierárquico divisivo, na mesma linha do que será apresentado a seguir.
- DIANA (*DIvisive ANAlysis*) – O algoritmo DIANA, proposto por [Kaufman & Rousseeuw 2005], pode operar sobre conjuntos de padrões que seguem exatamente a mesma estrutura daqueles utilizados pelo algoritmo AGNES. Entretanto, o algoritmo considera que inicialmente todos os padrões fazem parte de um único grupo e, então, divide esse grupo em dois grupos menores. Nessa divisão não são consideradas todas as possibilidades, mas é utilizado um procedimento iterativo que otimiza a escolha dos padrões participantes de cada novo grupo. Nesse procedimento, o padrão menos semelhante a todos os outros é selecionado e utilizado para a criação de um novo grupo. A seleção aleatória é utilizada caso mais de um padrão esteja apto para ser selecionado. Na sequência são selecionados outros padrões que são mais semelhantes ao novo grupo que ao grupo inicial e, então, esses padrões são transferidos para o novo grupo. Esse processo se repete dividindo, a cada iteração, o grupo que contém o maior valor de diâmetro (distância intra-grupo) até que restem apenas grupos unitários (*singletons*).

No Capítulo 3 são apresentados as principais estratégias de agrupamento aglomerativo (*Single Linkage*, *Complete Linkage* e *Average Linkage*), bem como um *trace* da execução do Esquema Aglomerativo Generalizado, que recebe como *input* o mesmo conjunto de padrões (COP) utilizado na Seção 2.4.

Capítulo 3

Algoritmos Hieráquicos Aglomerativos de Agrupamento Baseados em Teoria de Matrizes

Este capítulo está organizado da seguinte forma: Na Seção 3.1 é apresentada uma formalização da notação dos principais conceitos envolvidos em agrupamentos e na Seção 3.2 o Esquema Aglomerativo Generalizado (EAG) é apresentado e discutido em detalhes.

3.1 Notação Adotada

Considere que o conjunto de N padrões a serem agrupados, seja $X = \{P_1, P_2, ..., P_N\}$ em que cada padrão P_i, $1 \leq i \leq N$ é descrito por M atributos, $A_1, A_2, ..., A_M$. Um K-agrupamento de X é uma partição de X em K conjuntos (grupos), $G_1, G_2, ..., G_K$, ou seja, K-Agrupamento = $\{G_1, G_2, ..., G_K\}$. As três condições a seguir devem ser verificadas:

(1) $G_i \neq \emptyset$, i = 1, ..., K (cada um dos grupos é não-vazio)

(2) $\bigcup_{i=1}^{K} G_i = X$ (a união de todos os grupos recompõe o conjunto X original)

(3) $G_i \cap G_j = \emptyset$, $i \neq j$ e i, j = 1, ..., K (os grupos são dois-a-dois disjuntos)

Assume-se que os padrões que pertencem a cada um dos grupos G_i ($1 \leq i \leq K$), quando comparados entre si, são “mais semelhantes” do que quando comparados com padrões que pertencem a um outro grupo, que não o G_i [Jain *et al.*, 1999].

O conceito de similaridade adotado para a implementação de algoritmos de agrupamento desempenha um papel altamente relevante no resultado obtido. Uma maneira de implementar o conceito de similaridade é por meio do uso de uma medida de distância definida no espaço de atributos (que descrevem os padrões); dois padrões P_i e P_j são considerados similares se estiverem 'perto' um do outro, em que 'estar perto' precisa também ser quantificado. Nesta dissertação o 'estar perto' é quantificado por meio de uma medida de distância, no caso, a distância Euclidiana.

Seja um espaço M-dimensional definido por M atributos (de dados) e dois padrões desse espaço, P_i e P_j, representados por (P_{i_1}, P_{i_2}, ..., P_{i_M}) e P_j = (P_{j_1}, P_{j_2}, ..., P_{j_M}) respectivamente. A distância euclidiana entre os dois padrões, P_i e P_j, é calculada pela Eq. (3.1).

$$d(P_i, P_j) = \sqrt{\sum_{k=1}^{M} \left(P_{i_k} - P_{j_k}\right)^2} \qquad (3.1)$$

Considere novamente o conjunto de N padrões X = {P_1, P_2, ..., P_N} e dois agrupamentos dos padrões de X, identificados por AG_1 e AG_2, respectivamente. O agrupamento AG_1, contendo K grupos, está aninhado no agrupamento AG_2 que contém R (< K) grupos, (notado por $AG_1 \langle AG_2$) se cada grupo em AG_1 for subconjunto de um conjunto de AG_2 e, pelo menos um grupo de AG_1 for um subconjunto próprio de um grupo de AG_2.

Considerando X = {P_1, P_2, P_3, P_4, P_5}, o agrupamento A_1 = {{P_1,P_3}, {P_4}, {P_2,P_5}} está aninhado em A_2 = {{P_1,P_3,P_4}, {P_2,P_5}}. Entretanto, A_1 não está aninhado nem em A_3 = {{P_1,P_4}, {P_3}, {P_2,P_5}} ou tampouco em A_4 = {{P_1,P_2,P_4}, {P_3,P_5}}. A Figura 3.1 apresenta visualmente essas três situações.

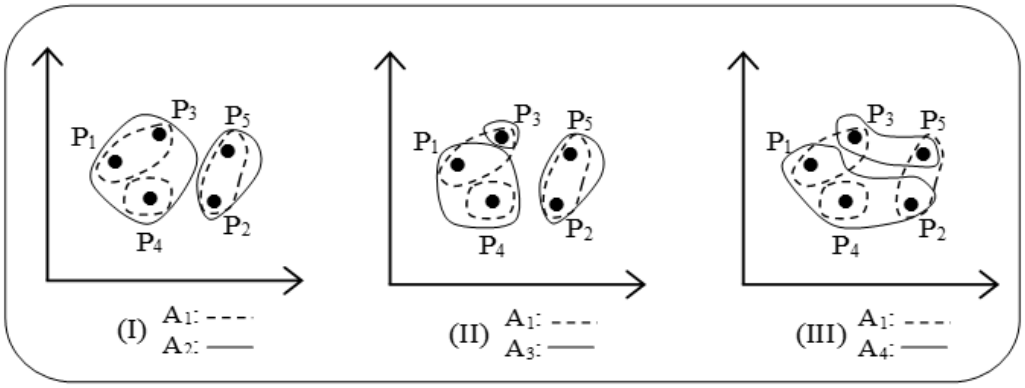


Figura 3.1 (I) agrupamento A_1 = {{P_1,P_3},{P_4},{P_2,P_5}} aninhado em A_2 = {{P_1,P_3,P_4},{P_2,P_5}}; (II) agrupamento A_1 não está aninhado em A_3 = {{P_1, P_4}, {P_3},{P_2,P_5}}, (III) agrupamento A_1 não está aninhado em A_4 = {{P_1, P_2, P_4},{P_3,P_5}}.

3.2 Esquema Aglomerativo Generalizado (EAG)

A seguir, na Seção 3.2.1 são feitas considerações iniciais sobre agrupamentos hierárquicos e, na Seção 3.2.2, o Esquema Aglomerativo Generalizado (EAG) é abordado em detalhes.

3.2.1 Considerações Iniciais

Os algoritmos hierárquicos produzem uma hierarquia de agrupamentos aninhados; via de regra esses algoritmos envolvem N passos, ou seja, tantos quantos for o número de padrões disponibilizados. A cada passo t um novo agrupamento é produzido usando, para isso, o agrupamento produzido no passo anterior i.e., no passo $t-1$. Particularmente, os algoritmos hierárquicos aglomerativos produzem uma sequência de agrupamentos com um número decrescente de grupos; assim o agrupamento produzido em um passo t é baseado no agrupamento produzido no passo $t-1$, no qual dois grupos são unidos, diminuindo assim o número de grupos a cada passo [Theodoridis & Koutroumbas 2009]. O resultado do algoritmo é uma árvore de agrupamentos, conhecida como dendrograma, que permite visualizar o agrupamento construído, a cada iteração.

Considerando que o conjunto fornecido ao algoritmo hierárquico aglomerativo tenha N padrões, os seguintes passos são executados:

1. Criar um agrupamento inicial com N grupos, em que cada grupo (*singleton)* contém apenas um padrão do conjunto inicial fornecido;

2. Selecionar dois grupos do agrupamento corrente que exibem maior semelhança entre si;

3. Formar um novo grupo unindo os grupos selecionados no passo 2 formando, desta forma, um novo agrupamento, com um grupo a menos;

4. Decrementar o número de grupos do novo agrupamento obtido;

5. Avaliar a condição de parada: volta ao passo 2 enquanto o número de grupos for maior que um, caso contrário finaliza o processo.

A ideia básica do algoritmo é unir, a cada passo, os dois grupos de padrões que são mais semelhantes entre si. Considerando que a semelhança seja avaliada como distância; a cada iteração os dois grupos que estão separados pela menor distância, são unidos em um único grupo. Via de regra a definição adotada para

'menor distância' é o que diferencia os vários algoritmos hierárquicos aglomerativos. Neste contexto, estratégias foram propostas e, entre as mais conhecidas estão: (a) *Single Linkage*, (b) *Complete Linkage*, (c) *Average Linkage*, (d) *Centroid-based* e (e) *Ward* (ver, por exemplo, [Theodoridis & Koutroumbas 2009], [Everitt *et al.*, 2011], [Duda *et al*, 2001], [Kaufman & Rousseeuw 2005]). Na sequência uma breve descrição de (a), (b) e (c) é apresentada.

(a) *Single Linkage*: Em algoritmos que adotam a estratégia *single linkage* a distância entre dois grupos G_i e G_j é determinada por meio de um par de padrões. Tal par de padrões P_i e P_j deve ser tal que: (1) $P_i \in G_i$ e $P_j \in G_j$; (2) a distância entre P_i e P_j é a menor dentre todas as distâncias entre todos os pares de padrões que satisfazem a condição (1), como estabelece Eq. (3.2).

$$d_G(G_i, G_j) = \min\{d(P_i, P_j) \mid P_i \in G_i \,\&\, P_j \in G_j, i \neq j\} \tag{3.2}$$

A Figura 3.2 ilustra a determinação da distância entre os grupos G_1 e G_2, em agrupamento *single linkage*.

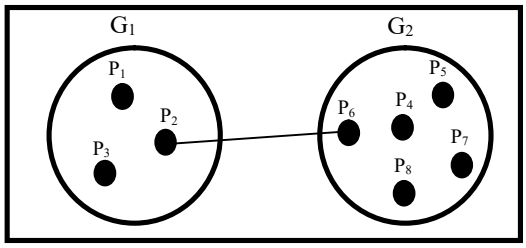


Figura 3.2 (*Single linkage*) Distância entre os grupos G_1 e G_2 é definida pelo par de padrões P_2 e P_6, uma vez que tal par é o que exibe a menor distância, considerando todos os outros pares (em que padrões dos pares pertencem a grupos distintos) que podem ser formados.

Para melhor entendimento do agrupamento *single linkage*, na Seção 3.2.3 é apresentado um *trace* completo que mostra em detalhes o uso desta estratégia.

(b) *Complete Linkage*: Essa abordagem para o estabelecimento de um valor de distância entre dois grupos, G_i e G_j, compara todos os pares de padrões P_i e P_j tal que: (1) $P_i \in G_i$ e $P_j \in G_j$; (2) a distância entre P_i e P_j é a maior dentre todas as distâncias entre todos os pares de padrões que satisfazem condição (1), como estabelece Eq. (3.3).

$$d_G(G_i, G_j) = \max\{d(P_i, P_j) \mid P_i \in G_i \,\&\, P_j \in G_j, i \neq j\} \tag{3.3}$$

Algoritmos que adotam a estratégia *complete linkage* para a determinação da distância entre dois grupos evitam uma desvantagem característica daqueles algoritmos que adotam a estratégia *single linkage*, chamada *fenômeno de encadeamento*. Pode facilmente acontecer, quando do uso do *single linkage*, que a distância entre dois grupos específicos acaba por ser a menor, devido apenas à presença de um par de padrões que têm uma distância pequena, enquanto todos os demais pares formados têm uma distância grande ou seja, estão bem distantes uns dos outros (que representaria uma situação em que tais grupos não estão próximos, a menos de um par de padrões).

A Figura 3.3 ilustra o processo em que a distância entre os grupos G_1 e G_2 é determinada pelo par de padrões P_3 e P_7, que têm entre si a maior distância dentre todas as distâncias entre possíveis pares.

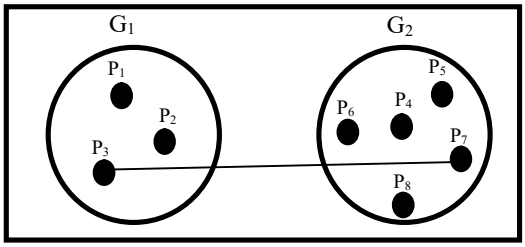


Figura 3.3 (*Complete linkage*) Distância entre os grupos G_1 e G_2 é definida pelo par de padrões P_3 e P_7, uma vez que tal par é o que exibe a maior distância, considerando todos os outros pares (em que padrões dos pares pertencem a grupos distintos) que podem ser formados.

A estratégia *complete linkage* normalmente, produz grupos compactos ou menos alongados do que aqueles gerados pelo *single linkage*. Uma característica do *complete linkage* é a tendência de não formar grupos grandes. Entretanto, esta abordagem é menos suscetível a ruídos e *outliers* (exceções).

(c) *Average Linkage*: Essa estratégia (também conhecida como UPGMA [Everitt *et al*., 2011]), diferente das duas anteriores, calcula a distância média entre pares de padrões pertencentes a grupos distintos, conforme ilustrado na Figura 3.4, tendendo a combinar grupos com pequenas variações. Por ser uma abordagem intermediária entre o *single linkage* e *complete linkage,* sua sensibilidade à presença de *outliers* é reduzida [Duda *et al*., 2001].

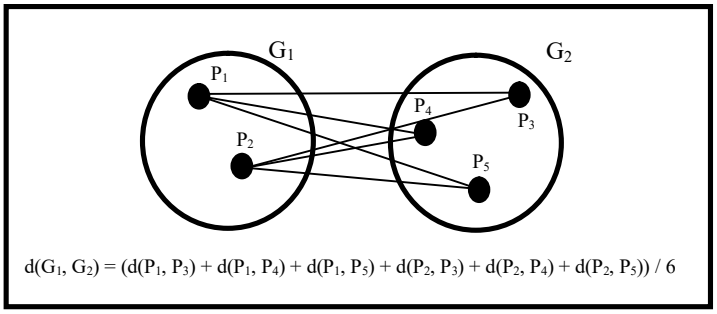


Figura 3.4 (*Average Linkage*) Distância entre os grupos G_1 e G_2 é a média das distâncias entre todos os pares de padrões de grupos diferentes.

A estratégia WPGMA (*Weighted Pair Group Method with Arithmetic Mean*) é uma variação *ponderada* da estratégia *Average Linkage*, em que a distância entre dois grupos é a média ponderada entre todos os pares de padrões de grupos diferentes. Em outras palavras, a distância entre G_1 e G_2, por exemplo, é a soma das distâncias entre todos os pares de padrões de grupos diferentes dividido por 2.

3.2.2 Descrição do EAG

Seja o conjunto de N padrões M-dimensionais $X = \{P_1, P_2, ..., P_N\}$ e considere todos os possíveis subconjuntos dois-a-dois disjuntos de X, $SC(X) = \{G_1, G_2, ..., G_h\}$. Seja $g(G_i,G_j)$ (i,j = 1, ..., h) uma função definida em $SC(X)\times SC(X)$, com valores reais, que mede a proximidade entre os subconjuntos G_i e G_j (i,j = 1, ..., h) e seja *t* o nível corrente do processo de obtenção de agrupamentos.

Descrevendo de forma breve, o agrupamento inicial AG_0 consiste de N grupos, cada um contendo um único elemento (padrão) de X. Na primeira iteração, o agrupamento AG_1 é produzido contendo N – 1 grupos, tal que $AG_0 \langle AG_1$. O procedimento continua até que o agrupamento final, AG_{N-1}, que possui um único grupo, seja obtido.

A Figura 3.5 apresenta o pseudocódigo em alto nível do algoritmo EAG (em inglês *GAS – Generalized Agglomerative Scheme*), que cria uma hierarquia de N agrupamentos, de maneira que cada um está aninhado em todos os agrupamentos sucessivos ou seja, $AG_t \langle AG_s$, para $t < s$, $s = 1, ..., N–1$.

procedure EAG (X, AG_t)
Input: X = {P_1, P_2,...,P_N}
Output: AG_t
1. **begin**
2. $AG_0 \leftarrow$ {{P_1}, {P_2},...,{P_N}} % agrupamento inicial
3. Nro_G $\leftarrow$ N
4. t $\leftarrow$ 0
5. **repeat**
6. t $\leftarrow$ t + 1
7. entre todos os possíveis pares de grupos (G_r,G_s) em AG_{t-1}, encontrar (G_i,G_j) tal que:

$$g(Gi,Gj) = \begin{cases} \min_{r,s}\, g(G_r,G_s) \text{ se g for função de dissimilaridade} \\ \max_{r,s}\, g(G_r,G_s) \text{ se g for função de similaridade} \end{cases}$$

8. New_G $\leftarrow$ $G_i \cup G_j$
9. Nro_G $\leftarrow$ Nro_G – 1
10. $G_{Nro\_G} \leftarrow$ New_G
11. AGt $\leftarrow$ $(AG_{t-1} - \{G_i,G_j\}) \cup \{G_{Nro\_G}\}$
12. **until** todos os padrões pertençam a um único grupo.
13. **end**
return AG_t % $AG_t \langle AG_s$, para t < s, s = 1, ..., N–1
end procedure

Figura 3.5 Pseudocódigo do Esquema Aglomerativo Generalizado (EAG) de Agrupamento (adaptado de Theodoridis & Koutroumbas (2009)).

Note que se dois padrões são agrupados em um único grupo no nível *t* da hierarquia, estes tendem a permanecer no mesmo grupo para todos os agrupamentos subsequentes. Esta *propriedade de agrupamento* apresenta uma desvantagem; não dispõe meios para recuperar de um agrupamento "ruim", que pode ter ocorrido em um nível anterior da hierarquia [Gower 1967].

A cada nível *t*, há N – *t* grupos e, para determinar o par de grupos que serão unidos no nível *t* + 1, $\binom{N-t}{2} \equiv \frac{(N-t)(N-t-1)}{2}$ pares de grupos devem ser considerados. O número de operações necessárias por um esquema aglomerativo genérico é proporcional a N^3. Entretanto, a complexidade do algoritmo depende da definição da função de dissimilaridade cuja formalização é apresentada na Seção 3.3.

Na próxima seção é apresentado um exemplo passo-a-passo do EAG no agrupamento de um conjunto de padrões artificialmente gerados.

3.2.3 *Trace* Alto Nível do EAG no Conjunto Original de Padrões (COP)

Nesta seção são descritos os passos realizados pelo procedimento EAG para a geração de um agrupamento, considerando como conjunto inicial de padrões o conjunto X =

{(2,6 4,5), (1,5 2,1), (5,3 3,4), (1,9 2,0), (3,7 4,7)}. A cada grupo criado (com exceção dos grupos do agrupamento AG_0), é apresentado um dendrograma para facilitar a visualização dos grupos e da hierarquia de agrupamentos produzidas pelo algoritmo. Na sequência são descritos os passos realizados pelo algoritmo:

- Inicia o índice para o número de grupos com valor 5. Nro_G = 5.
- Cria o agrupamento inicial $AG_0 = \{G_0, G_1, G_2, G_3, G_4\}$ sendo $G_0 = \{(2,6\ 4,5)\}$, $G_1 = \{(1,5\ 2,1)\}$ $G_2 = \{(5,3\ 3,4)\}$, $G_3 = \{(1,9\ 2,0)\}$, $G_4 = \{(3,7\ 4,7)\}$, em que cada padrão pertence a um único grupo.
- Contador do número de iterações inicia com zero. $t = 0$.
- Incrementa o contador t em 1. $t = 1$.
- Calcula a dissimilaridade (distância) entre todos os grupos tomados dois a dois (Tabelas 3.1 e 3.2) e as inclui na matriz de dissimilaridade (MD).
- Ordena os elementos de MD, em ordem crescente, e recupera o valor 0.41 que corresponde a menor distância encontrada (calculada entre os grupos G_1 e G_3).

Tabela 3.1 Valores das distâncias entre grupos do agrupamento AG_0.

$d(G_0, G_1)$	$\sqrt{(2,6-1,5)^2+(4,5-2,1)^2} = 2,64$
$d(G_0, G_2)$	$\sqrt{(2,6-5,3)^2+(4,5-3,4)^2} = 2,91$
$d(G_0, G_3)$	$\sqrt{(2,6-1,9)^2+(4,5-2,0)^2} = 2,59$
$d(G_0, G_4)$	$\sqrt{(2,6-3,7)^2+(4,5-4,7)^2} = 1,11$
$d(G_1, G_2)$	$\sqrt{(1,5-5,3)^2+(2,1-3,4)^2} = 4,01$
$d(G_1, G_3)$	$\sqrt{(1,5-1,9)^2+(2,1-2,0)^2} = 0,41$
$d(G_1, G_4)$	$\sqrt{(1,5-3,7)^2+(2,1-4,7)^2} = 3,40$
$d(G_2, G_3)$	$\sqrt{(5,3-1,9)^2+(3,4-2,0)^2} = 3,67$
$d(G_2, G_4)$	$\sqrt{(5,3-3,7)^2+(3,4-4,7)^2} = 2,06$
$d(G_3, G_4)$	$\sqrt{(1,9-3,7)^2+(2,0-4,7)^2} = 3,24$

Tabela 3.2 Matriz de Dissimilaridade (MD) gerada a partir da Tabela 3.1.

Grupo	**G_0**	**G_1**	**G_2**	**G_3**	**G_4**
G_0	0				
G_1	2,64	0			
G_2	2,91	4,01	0		
G_3	2,59	0,41	3,67	0	
G_4	1,11	3,40	2,06	3,24	0

- Cria um novo grupo que contém a união de G_1 e G_3 ($G_5 = G_1 \cup G_3$).
- $G_5 = \{(1,5\ 2,1), (1,9\ 2,0)\}$

- Remove G_1 e G_3 de AG_0 e faz a união de AG_0 com G_5, resultando no agrupamento AG_1.
- $AG_1 = (AG_0 - \{G_1, G_3\}) \cup \{G_5\}$.
- $AG_1 = \{\{(2{,}6\ 4{,}5)\}, \{(5{,}3\ 3{,}4)\}, \{(3{,}7\ 4{,}7)\}, \mathbf{\{(1{,}5\ 2{,}1), (1{,}9\ 2{,}0)\}}\}$
- Agrupamento atual, $AG_1 = \{G_0, G_2, G_4, \mathbf{G_5}\}$. Nro_G = 4.

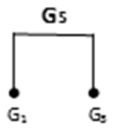


Figura 3.6 Dendrograma ilustrando o grupo G_5, formado no agrupamento AG_1.

- Incrementa o contador *t* em 1. *t* = 2.
- Calcula a dissimilaridade entre o novo grupo G_5 e todos os grupos restantes G_0, G_2 e G_4. No cálculo da dissimilaridade de G_5 para os outros grupos, são utilizadas as distâncias de G_1 e G_3 previamente calculadas (o reaproveitamento dos cálculos proporciona melhor desempenho do algoritmo).

Tabela 3.3 Valores das distâncias entre grupos do agrupamento AG_1.

$d(G_0, G_5)$	$f(d(G_0, G_1), d(G_0, G_3)) = 2{,}59$
$d(G_2, G_5)$	$f(d(G_2, G_1), d(G_2, G_3)) = 3{,}67$
$d(G_4, G_5)$	$f(d(G_4, G_1), d(G_4, G_3)) = 3{,}24$

Observação: Note que a função *f* retorna o menor valor entre os dois argumentos, i.e., utiliza a estratégia *single linkage* sendo os valores 2,59 = $d(G_0, G_3)$, 3,67 = $d(G_2, G_3)$ e 3,24 = $d(G_4, G_3)$.

- Atualiza a MD removendo as informações de G_1 e G_3 e incluindo as de G_5. Os valores das distâncias entre G_5 e os demais grupos correspondem a menor distância entre G_i e G_1 e G_i e G_3, sendo i = {0, 2, 4}.

Tabela 3.4 Matriz de Dissimilaridade (MD) gerada a partir da Tabela 3.3

Grupo	**G_0**	**G_2**	**G_4**	**G_5**
G_0	0			
G_2	2,91	0		
G_4	1,11	2,06	0	
G_5	2,59	3,67	3,24	0

- Ordena os elementos de MD, em ordem crescente, e recupera o valor 1,11 que corresponde a menor distância encontrada (calculada entre os grupos G_0 e G_4).
- Cria um novo grupo e faz a união dos grupos G_0 e G_4 (New_G = $G_4 \cup G_0$).

$G_6 = \{(3{,}7\ 4{,}7), (2{,}6\ 4{,}5)\}$

- Remove G_0 e G_4 de AG_1 e faz a união de AG_1 com G_6, resultando no novo agrupamento AG_2. $AG_2 = (AG_1 - \{G_4, G_0\}) \cup G_6$. AG_2 = {{(5,3 3,4)}, {(1,5 2,1), (1,9 2,0)}, **{(3,7 4,7), (2,6 4,5)}**.
- Forma o agrupamento $AG_2 = \{G_2, G_5, \mathbf{G_6}\}$. Nro_G = 3.

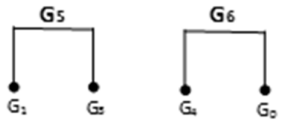


Figura 3.7 Dendrograma ilustrando os grupos G_5 e G_6, formados no agrupamento AG_2.

- Calcula a dissimilaridade entre o novo grupo G_6 e os grupo G_2 e G_5. No cálculo da dissimilaridade de G_6 para os grupos restantes, são utilizadas as distancias de G_0 e G_4 previamente calculadas.

Tabela 3.5 Valores das distâncias entre grupos do agrupamento AG_2.

$d(G_2, G_6)$	$f(d(G_2, G_4), d(G_2, G_0)) = 2{,}06$
$d(G_5, G_6)$	$f(d(G_5, G_4), d(G_5, G_0)) = 2{,}59$

Observação: Note que os valores 2,06 = $d(G_2, G_4)$ e 2,59 = $d(G_5, G_0)$.

- Atualiza a MD, removendo as informações de G_0 e G_4 e incluindo as de G_6. Os valores das distâncias entre G_6 e os demais grupos correspondem a menor distância entre G_i e G_4 e G_i e G_0, sendo i = {2,5}.

Tabela 3.6 Matriz de Dissimilaridade (MD) gerada a partir da Tabela 3.5.

Grupo	**G_2**	**G_5**	**G_6**
G_2	0		
G_5	3,67	0	
G_6	2,06	2,59	0

- Ordena os elementos de MD, em ordem crescente, e recupera o valor 2,06 que corresponde a menor distância (calculada entre os grupos G_2 e G_6).
- Cria um grupo e faz a união dos grupos G_2 e G_6 (New_G = $G_2 \cup G_6$). G_7 = {(5,3 3,4), (3,7 4,7), (2,6 4,5)}
- Remove G_2 e G_6 de AG_2 e faz a união de AG_2 com G_7, resultando no agrupamento AG_3. $AG_3 = (AG_2 - \{G_6, G_2\}) \cup G_7$. AG_3 = {{(1,5 2,1), (1,9 2,0)}, **{(5,3 3,4), (3,7 4,7), (2,6 4,5)}**}
- Forma o agrupamento $AG_3 = \{G_5, \mathbf{G_7}\}$. Nro_G = 2.

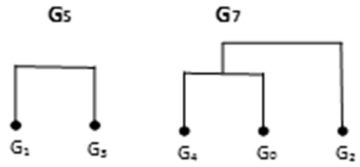


Figura 3.8 Dendrograma ilustrando os grupos G_5 e G_7, formados no agrupamento AG3.

- Continua o agrupamento fazendo a união de G_5 e G_7 resultando no novo grupo G_8. O ponto de parada do algoritmo ocorre quando o agrupamento final, AG_4 é composto de um único grupo. $G_8 = \{\{G_1, G_3\}, \{\{G_0, G_4\}, G_2\}\}$
- Forma o agrupamento final $AG_4 = \{\mathbf{G_8}\}$. Nro_G = 1.

O dendrograma a seguir ilustra o grupo G_8, formado no agrupamento final AG_4.

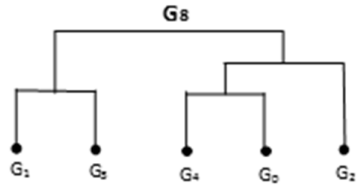


Figura 3.9 Dendrograma ilustrando o grupo G_8, formado no agrupamento final AG_4.

3.3 Algoritmos Aglomerativos Baseados em Teoria de Matrizes

De acordo com Theodoridis & Koutroumbas (2009), existem duas principais categorias de algoritmos aglomerativos: (1) aqueles baseados em conceitos de Teoria de Matrizes; e (2) aqueles baseados em conceitos de Teoria dos Grafos. Esta seção discute em detalhes a primeira categoria, mas antes algumas definições são necessárias para melhor entendimento desta abordagem.

3.3.1 Definições Relevantes

Definição 3.1 Um conjunto de N padrões M-dimensionais $X = \{P_1, P_2, ..., P_N\}$, pode ser associado a uma matriz de dimensão N × M, chamada *matriz de padrões*, notada por *MP(X)*, cuja i-ésima linha representa o i-ésimo padrão de X.

Como exemplo, considere X = {(2,6 4,5), (1,5 2,1), (5,3 3,4), (1,9 2,0), (3,7 4,7)}. A matriz de padrões de X é:

$$MP(X) = \begin{bmatrix} 2,6 & 4,5 \\ 1,5 & 2,1 \\ 5,3 & 3,4 \\ 1,9 & 2,0 \\ 3,7 & 4,7 \end{bmatrix}$$

Definição 3.2 Considere um conjunto de N padrões X = {P_1, P_2, ..., P_N}. Ao conjunto X pode ser associada uma matriz chamada *matriz de dissimilaridade*, notada por *MD(X)*, que é uma matriz N × N em que seu elemento md_{ij} representa a dissimilaridade entre os padrões P_i e P_j, para i, j = 1, ..., N. A matriz de dissimilaridade correspondente a X é:

$$MD(X) = \begin{bmatrix} 0 & 2,64 & 2,91 & 2,59 & 1,11 \\ 2,64 & 0 & 4,01 & 0,41 & 3,40 \\ 2,91 & 4,01 & 0 & 3,67 & 2,06 \\ 2,59 & 0,41 & 3,67 & 0 & 3,24 \\ 1,11 & 3,40 & 2,06 & 3,24 & 0 \end{bmatrix}$$

3.3.2 Dendrograma

Como pode ser observado no exemplo da Seção 3.2.3, um dendrograma é um tipo de grafo que descreve a sequência de agrupamentos produzidos por um algoritmo aglomerativo; cada novo grupo formado no EAG corresponde a um nível nesta estrutura. Um *corte* horizontal no dendrograma (como o mostrado pela linha tracejada na Figura 3.10), em determinada altura, resultaria em três agrupamentos: AG_1 = {G_1, G_3}, AG_2 = {G_0, G_4} e AG_3 = {G_2}.

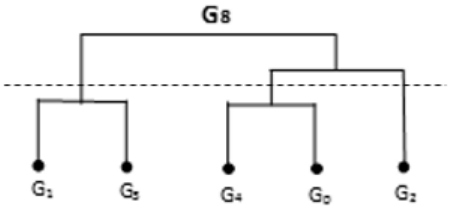


Figura 3.10 Linha tracejada representando um corte horizontal no dendrograma.

Como discutido em [Theodoridis & Koutroumbas 2009], um *dendrograma de proximidade* é um dendrograma que leva em consideração o nível de proximidade e o momento em que ocorreu, pela primeira vez, a fusão de dois grupos. Quando uma medida de dissimilaridade é utilizada, o dendrograma de proximidade é chamado de *dendrograma de dissimilaridade*. Esta forma de representar o nível de proximidade pode ser utilizada na identificação da formação de grupos em qualquer nível, servindo

como uma ferramenta para a escolha de qual método de agrupamento é mais adequado aos padrões.

Como mencionado na Seção 3.2.1, o método de agrupamento ou o tipo de algoritmo determina toda a hierarquia de agrupamento. Neste aspecto, o dendrograma como um todo pode ser útil em algumas aplicações como, por exemplo, na área da biologia (e a sua taxonomia). Entretanto, em outras aplicações, há interesse somente em um tipo específico de agrupamento, aquele que melhor se adéqua aos padrões. De forma equivalente, um corte horizontal em determinado nível (*threshold*) no dendrograma pode resultar no agrupamento com um número pré-determinado de grupos.

3.3.3 *Matrix Updating Algoritmic Scheme* (MUAS)

Algoritmos de agrupamento aglomerativo baseados em Teoria de Matrizes podem ser vistos como casos particulares do EAG, tendo como *input* a matriz de padrões, *MP*, ou a matriz de dissimilaridade, $MD_0 = MD(X)$, construída a partir de X. O elemento $md_{ij} \in MD(X)$ é o valor da dissimilaridade (distância euclidiana, por exemplo) entre os padrões P_i e P_j do conjunto X, para i, j = 1, ..., N.

Como discutido em Theodoridis & Koutroumbas (2009), a cada nível *t*, quando dois grupos são unidos em apenas um, o tamanho da matriz de dissimilaridade MD_t se torna $(N - t) \times (N - t)$. MD_t é construída a partir de MD_{t-1} por meio da (1) exclusão das duas linhas e das duas colunas que correspondem aos grupos que foram unidos e (2) adição de uma nova linha e uma nova coluna, contendo as distâncias do novo grupo formado e os outros grupos do agrupamento. A distância entre o novo grupo formado pela união de dois grupos G_i e G_j (i.e., $G_q = G_i \cup G_j$) e um grupo antigo G_s é uma função (f) como representada em Eq. (3.4), ou seja, depende das três distâncias entre os grupos: G_i e G_s, G_j e G_s e entre G_i e G_j. Para sua implementação é preciso primeiro fazer uma escolha de qual função utilizar para calcular a distância entre dois grupos de padrões (ver as mais populares em [Theodoridis & Koutroumbas 2009]).

$$d(G_q,G_s) = f(d(G_i,G_s), d(G_j,G_s), d(G_i,G_j)) \quad (3.4)$$

A Figura 3.11 apresenta o pseudocódigo do algoritmo MUAS. Como mencionado anteriormente o *input* do algoritmo pode ser de duas formas (matriz de dissimilaridades ou matriz de padrões), entretanto foi definido que o MUAS aceitará apenas o conjunto de padrões como entrada. Na linha 13 é chamado um subalgorítmo

que atualiza a matriz de dissimilaridade conforme descrito no primeiro e segundo parágrafo desta seção.

procedure MUAS (X, AG_t)
Input: X = {P_1, P_2,...,P_N}
Output: AG_t
1. **begin**
2. $AG_0 \leftarrow$ {{P_1}, {P_2},...,{P_N}} % agrupamento inicial
3. Nro_G $\leftarrow$ N
4. t $\leftarrow$ 0
5. Cria a matriz de dissimilaridades MD_t
6. **repeat**
7. t $\leftarrow$ t + 1
8. encontre G_i,G_j, tal que g(Gi,Gj) = $\min_{r,s}$ g(G_r,G_s)
9. New_G $\leftarrow$ $G_i \cup G_j$
10. Nro_G $\leftarrow$ Nro_G – 1
11. GNro_G $\leftarrow$ New_G
12. AGt $\leftarrow$ (AG_{t-1} – {G_i,G_j}) $\cup$ {G_{Nro_G}}
13. Atualiza a matriz de dissimilaridades MD_t a partir de MD_{t-1}
14. **until** todos os padrões pertençam a um único grupo.
15. **end**
return AG_t % $AG_t \langle AG_s$, para t < s, s = 1, ..., N–1
end procedure

Figura 3.11 Pseudocódigo do *Matrix Updating Algorithmic Scheme* (MUAS) adaptado de Theodoridis & Koutroumbas (2009).

3.3.4 Sobre a Complexidade do EAG

Como discutido na Seção 3.2.2, a complexidade do algoritmo que implementa o EAG é $O(N^3)$. Entretanto, várias implementações destes esquemas tem sido propostas na literatura na qual se propõe reduzir o tempo computacional para O(N). Em [Kurita 1991], por exemplo, uma implementação é discutida, no qual o tempo computacional é reduzido para $O(N^2 \log N)$. Em [Murtagh 1983, 1984, 1985] foram discutidas implementações dos vários algoritmos aglomerativos que reduziram para $O(N^2)$ o tempo computacional. Implementações com suporte a paralelismo são discutidas em [Willett 1989], [Li 1990] e [Olson 1993]. Mais recentemente, experimentos utilizando métodos ativos de agrupamento em [Eriksson *et al*. 2001] e agrupamento espectral em [Krishnamurthy *et al*., 2012] também evidenciaram ganhos substanciais em desempenho e precisão. Para uma discussão aprofundada da complexidade computacional dos algoritmos propostos neste projeto, veja [Müller 2011].

No presente trabalho foi implementado um algoritmo de agrupamento aglomerativo parametrizável para os métodos de agrupamentos (*Single Linkage*, *Complete Linkage* e *Average Linkage*) discutidos na Seção 3.2.

3.4 Um Exemplo de Aprendizado Indutivo Não-supervisionado Usando o Método de Agrupamento *Complete Linkage*

Nesta seção é apresentado um exemplo do uso de um algoritmo de agrupamento hierárquico, tendo como entrada o mesmo conjunto de padrões (COP) apresentado na Seção 2.4. O exemplo tem como ambiente computacional o software Weka que, dentre as várias estratégias de agrupamento, disponibiliza o método aglomerativo *Complete Linkage*. A escolha deste método de agrupamento tem como base a tentativa de atender alguns dos critérios para validação dos resultados que foram discutidos na seção anterior.

A intenção, neste exemplo, é permitir uma análise comparativa entre os resultados do processo de classificação (supervisionado) e agrupamento (não-supervisionado). O exemplo de aprendizado supervisionado foi apresentado no Capítulo 2 (ver Seção 2.4 Um Exemplo Didático de Aprendizado Indutivo Supervisionado Usando Validação Cruzada). Nesta seção é discutido apenas o resultado desta comparação, não sendo descritos os passos da realização do processo de agrupamento no ambiente Weka.

Na execução do processo de agrupamento do COP no Weka, foi escolhido o método *Complete Linkage* (estratégia de agrupamento hierárquico) passando como parâmetros a função da distância (Euclidiana) e o número de grupos (3).

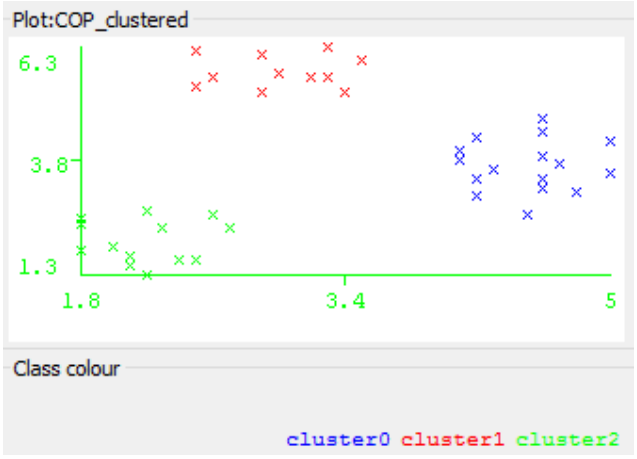


Figura 3.12 Agrupamento do COP, pelo método *Complete Linkage* disponível no Weka.

Para a validação (externa) do agrupamento mostrado na Figura 3.12, o mesmo conjunto COP pode ser utilizado como gabarito; as informações sobre a quantidade de classes e padrões por classe podem indicar as quantidades de grupos e padrões por grupo. Na sequência, é mostrado um novo processo de agrupamento no Weka utilizando, desta vez, o atributo classe (c) para validação dos grupos.

A Figura 3.13 mostra o resultado do agrupamento em que os 16 padrões foram alocados ao grupo 0 (cluster0), 11 padrões no grupo 1 (cluster1) e 13 padrões no grupo 2 (cluster2). No processo de agrupamento, o atributo classe (c) do COP foi ignorado para evitar sua influência no cálculo da distância entre os padrões.

```
=== Model and evaluation on training set ===

Clustered Instances

0      16 ( 40%)
1      11 ( 28%)
2      13 ( 33%)


Class attribute: c
Classes to Clusters:

  0  1  2  <-- assigned to cluster
 16  0  0 | 1
  0  0 13 | 2
  0 11  0 | 3

Cluster 0 <-- 1
Cluster 1 <-- 3
Cluster 2 <-- 2

Incorrectly clustered instances :       0.0       0      %
```

Figura 3.13 Relatório, gerado pelo Weka, do processo de agrupamento (método *Complete Linkage*) utilizando o atributo classe (c) para validação dos grupos.

A Figura 3.13 mostra o relatório do Weka com o resultado do processo de agrupamento, considerando o atributo classe (c) para validação dos grupos. Note que as 3 classes (1, 3 e 2) correspondem, respectivamente, aos 3 grupos (0, 1 e 2) identificados no processo de agrupamento. A quantidade de padrões por grupo também corresponde a quantidade de padrões por classe.

Para finalizar, a Figura 3.14 apresenta o dendrograma que mostra a sequência de agrupamentos dos 40 padrões do COP.

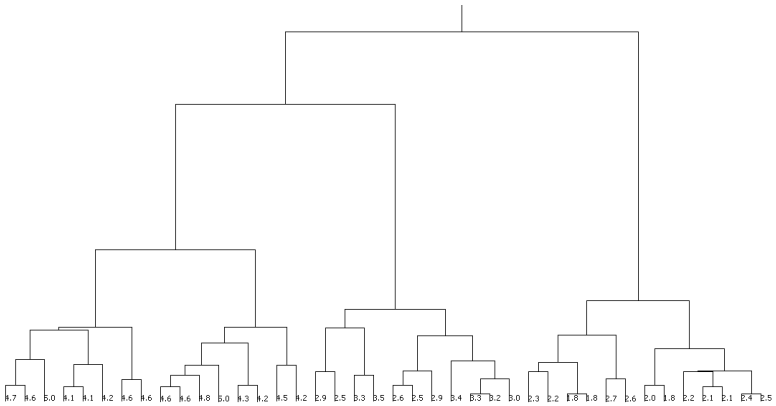


Figura 3.14 Dendrograma mostrando a sequência de agrupamentos dos 40 padrões do COP.

Capítulo 4
Validação de Agrupamentos

No contexto desta pesquisa, a validação de agrupamentos produzidos pelos algoritmos aglomerativos é feita de duas formas: (1) por meio da avaliação dos valores dos índices de validação (internos e externos); (2) comparação (*benchmarking*) com agrupamentos produzidos pelo algoritmo particional K-Means.

A etapa de validação dos resultados diz respeito a procedimentos que avaliam, de forma objetiva e quantitativa, os resultados da análise dos agrupamentos. Uma das formas de realizar a validação de um agrupamento induzido por um algoritmo de agrupamento é por meio de índices estatísticos.

Na sequência são listados alguns dos aspectos relevantes relacionados ao processo de validação de agrupamentos [Tan *et al*., 2005].

1. Determinar a *tendência do agrupamento* em um conjunto de padrões (i.e., distinguir se há ou não uma estrutura aleatória dos dados);
2. Determinar o número correto de grupos;
3. Avaliar o quanto que o resultado da análise está adequado aos padrões sem ter que referenciar a informações externas;
4. Comparar os resultados da análise com resultados externos conhecidos, tais como rótulos de classes;
5. Comparar dois agrupamentos e determinar qual é o melhor.

Note que os itens 1, 2 e 3 não fazem uso de informações externas (são técnicas não-supervisionadas), ao contrário do item 4. O item 5 pode ser realizado de forma supervisionada ou não-supervisionada.

Como discutido em [Halkidi *et al*., 2001], embora seja possível desenvolver vários tipos de medições numéricas para implementar os diferentes aspectos de validação de agrupamentos mencionados anteriormente, muitos são os desafios envolvidos, considerando que: (1) uma medida de validação de agrupamento pode ser limitada no escopo de sua aplicabilidade. Por exemplo, a maioria dos trabalhos de medição de tendências de agrupamento tem sido feitos levando em consideração somente padrões bi ou tridimensionais; (2) é necessário um *framework* para

interpretar qualquer uma dessas medidas. Por exemplo, se for obtido o valor 10 de uma medição que avaliou quão bem os rótulos dos grupos condizem com os rótulos de classes fornecidos externamente, este valor representa uma correspondência boa, normal ou fraca entre os rótulos? O grau de correspondência pode ser medido analisando a distribuição estatística destes valores, ou seja, qual a probabilidade desta correspondência entre esses valores ocorrer.

Conforme sugerem [Theodoridis & Koutroumbas 2009], existem três abordagens para investigar a validade dos agrupamentos:

(1) baseada em *critérios externos*, i.e., avalia os resultados dos algoritmos de agrupamento baseado em uma estrutura pré-estabelecida. É uma abordagem supervisionada pois mede a extensão na qual a estrutura do agrupamento descoberta por um algoritmo de agrupamento corresponde a uma estrutura externa. Um exemplo de um índice supervisionado é a entropia (i.e., correspondência entre rótulos induzidos e classes obtidas de uma fonte externa);

(2) baseada em *critérios internos*, em que os resultados são avaliados em termos de quantidades relacionadas aos vetores do conjunto de padrões como, por exemplo, a matriz de proximidade. Pode ser vista como uma estratégia não-supervisionada pois mede a estrutura do agrupamento (em termos de coesão e isolamento) sem que seja necessário recorrer a informações externas, utilizando, por exemplo, a soma dos erros quadrados;

(3) baseada em *critérios relativos*, que compara a estrutura do agrupamento com outras geradas por diferentes esquemas de agrupamento. É uma medição que pode ser feita de forma supervisionada ou não-supervisionada dependendo do propósito da comparação.

Em [Berry & Linoff 1996] é proposto dois critérios para validação de agrupamentos e para a escolha de um esquema de agrupamento ótimo:

- Grau de compactação – os padrões de cada grupo devem estar o mais próximo possível entre si. Uma forma de medir o grau de compactação é a variância, a qual deve ser minimizada.
- Separação – a distância entre os grupos formados deve ser a maior possível. Como visto anteriormente, existem algumas abordagens para medir a distância entre dois grupos, como *Single Linkage*, *Complete Linkage* e *Average Linkage*.

4.1 Índice de Dunn e Índice de Davies-Bouldin

O índice de [Dunn 1973] é baseado em medidas geométricas dos grupos em que são consideradas as características de densidade e separação. A distância mínima entre grupos mede o quanto estão separados e o diâmetro máximo do grupo mede o seu grau de compactação. O índice de Dunn é definido pela Eq. (4.1):

$$D = \frac{d_{min}}{d_{max}}, d_{min} = min_{i,j\in\{1,\dots,m\},i\neq j}\ d_{p_i p_j}\ e\ d_{max} = max_{i,j\in\{1,..,m\},i=j} d_{p_i p_j} \qquad \text{Eq. (4.1)}$$

em que d_{min} denota a menor distância entre dois padrões de grupos distintos (i.e., menor distância inter-grupos), e d_{max} a maior distância entre dois padrões do mesmo grupo (i.e., maior distância intra-grupo). O valor de D é limitado ao intervalo [0, ∞] e deve ser maximizado.

O índice de [Davies & Bouldin 1979] se baseia nas medidas de similaridades e dispersão dos grupos e é defininido pela Eq. (4.2):

$$DB = \frac{1}{L}\sum_{i=1}^{l} max(d_{ij})_{j=1\dots l;j\neq i'}, d_{ij} = \frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \qquad \text{Eq. (4.2)}$$

em que L é o número de grupos, σ_i é a distância média de todos os padrões em um grupo i para o seu centroide c_i, σ_j é a distância média de todos os padrões no grupo j para o seu centroide c_j, e $d(c_i, c_j)$ é a distância entre os centróides c_i e c_j. Valores menores de DB correspondem a grupos que são compactos, seus centroides distantes um do outro. Ao contrário do índice de Dunn, o valor do índice de DB deve ser minimizado.

4.3 Índice de Rand e Índice de Jaccard

O índice de Dunn e o índice de Davies-Bouldin são métricas estatísticas para realizar validação interna em agrupamentos. Nesta seção são apresentados duas métricas para validação externa: índice de Rand e índice de Jaccard.

O índice de [Rand 1971] mede a similaridade entre dois agrupamentos. Este índice trabalha com duas partições V e U de um mesmo conjunto de padrões, sendo V correspondente a divisão conhecida a *priori* desse conjunto de padrões em K classes, e U a matriz obtida por meio do resultado da execução de um algoritmo de agrupamento. Tal índice pode ser calculado de acordo com a seguinte equação:

$$R = \frac{a + d}{a + b + c + d} \qquad \text{Eq. (4.3)}$$

em que a é o número de pares de padrões contidos nos mesmos grupos tanto em V quanto em U; b é o número de pares de padrões contidos em grupos diferentes V mas nos mesmos grupos em U; c é o número de pares de padrões contidos nos mesmos grupos em V mas em grupos diferentes em U; d é o número de pares de padrões contidos em grupos diferentes tanto em V como em U. O índice de Rand varia no intervalo [0.0, 1.0], em que valores próximos de 1 indicam um total grau de semelhança entre a partição a *priori* e a obtida, enquanto valores próximos de 0 indicam uma distribuição obtida por acaso [Milligan 1996].

O índice de Jaccard, também conhecido como coeficiente de similaridade de Jaccard [Jaccard 1908], elimina o termo d da Eq. (4.3) pois enfatiza a similaridade em termos de pares de padrões que pertençam juntos em ambas as partições [Zaki & Meira 2014].

$$R = \frac{a}{a + b + c} \qquad \text{Eq. (4.4)}$$

Desta forma, o índice de Jaccard tem valores no intervalo [0.0, 1.0], e assim como o índice de Rand, quanto mais próximo a 1 melhor a qualidade do resultado obtido. Em resumo, a diferença do índice de Jaccard em relação ao índice de Rand é que o índice de Jaccard não penaliza as classificações indicadas pela letra d.

4.4 Algorítimo K-Means

O algoritmo K-Means implementa uma técnica de agrupamento baseada em protótipos (centros) [Tan *et al.*, 2005]. O K-Means define um protótipo em termos de um centróide, que normalmente corresponde a média dos padrões em um grupo. K-Means inicia escolhendo K centróides iniciais, em que K é um parâmetro definido pelo usuário, que representa o número de grupos desejados. Cada padrão é então atribuído ao centróide mais próximo, e cada coleção de padrões atribuída ao centróide forma um grupo. O centróide de cada grupo é então atualizado baseado nos padrões atribuídos ao grupo. O processo de atribuição dos padrões e a atualização dos centróides se repete até que os centróides permaneçam inalterados.

```
procedure K-Means (X, K, AG)
Input: X = {P1, P2,...,PN}, K
Output: AG
1. begin
2. C ← escolheCentroides(X, K)
3. novos_centroides ← true
4. while novos_centroides do
5.   begin
6.     for i=1 to N do
7.       begin
8.         centroide_mais_proximo ← encontreCentroideMaisProximo(Pi, C)
9.         Gcentroide_mais_proximo ← Gcentroide_mais_proximo ∪ {Pi}
10.      end
11.     AG ← {G1, G2, ..., Gk}
12.     novoC ← recalcularCentroides(C, AG)
13.     novos_centroides ← verificaCentroides(C, novoC)
14.  end
15  end
return AG
end procedure
```

Figura 4.1 Pseudocódigo em alto nível do algoritmo K-Means adaptado de Jain *et al*. (1999).

O algoritmo K-Means é formalmente descrito pelo pseudocódigo mostrado na Figura 4.2. No subalgoritmo *escolheCentroides*, K padrões são aleatoriamente escolhidos do conjunto de padrões X como centróides iniciais. Na sequência, padrões são atribuidos aos centróides iniciais mais próximos a partir do resutlado do subalgoritmo *encontreCentroideMaisProximo*. Após os padrões terem sido atribuidos a um centróide, o centróide (i.e., média dos padrões) é então atualizado por meio da execução do subalgoritmo *recalcularCentroides*. O subalgoritmo *verificaCentroides* é chamado para checar se houve alteração do centróide; caso o centróides permaneçam inalterados o algoritmo encerra sua execução retornando o agrupamento resultante AG, caso contrário continua o laço (*while*) até que a condição de parada (i.e., novos_centroides = falso) seja satisfeita.

Uma das limitações do algoritmo K-Means é que ele atribui o mesmo peso a todos os atributos que descrevem o conjunto de padrões, entretanto, foram propostas extensões ao seu esquema genérico quanto à ponderação (*weighting*) de tais características [Amorim 2011]. Outra preocupação é que seus resultados são altamente dependentes da escolha dos centróides iniciais e do valor K informado pelo usuário.

Capítulo 5
Sistema Computacional AggloCluster – Principais Funcionalidades

Neste capítulo é apresentada uma descrição das funcionalidades do sistema computacional AggloCluster. O sistema disponibiliza a implementação do esquema *Matrix Updating Algorithmic Scheme* (MUAS), parametrizável para 4 estratégias de agrupamento (*Single Linkage*, *Complete Linkage*, UPGMA e WPGMA), bem como uma implementação do algoritmo AGNES, adaptada para induzir um número k de grupos. A versão clássica do AGNES [Kaufman & Rousseeuw 2005] implementa apenas a estratégia *Average Linkage* UPGMA (*Unweighted Pair Group Method with Arithmetic Mean*), que induz um agrupamento hierárquico com o maior número possível de grupos. A versão do AGNES disponibilizada no AggloCluster, entretanto, contempla as quatro estratégias mencionadas anteriormente.

O AggloCluster é composto pelos módulos de pré-processamento (painel *Preprocess*), agrupamento (painel *Cluster*) e validação (painel *Cluster Validity*). O desenvolvimento do sistema foi fundamental para o melhor entendimento de como funcionam os algoritmos hierárquicos aglomerativos, além de oferecer um ambiente computacional para a investigação e realização de experimentos. O sistema foi projetado para ser executado como um aplicativo *Windows Form* na plataforma Microsoft Windows; a linguagem C# em conjunto com o .NET Framework 4.0 foram utilizados para sua implementação. A seguir são apresentadas a sua arquitetura funcional e a descrição de cada um dos módulos.

5.1 Módulo de Pré-processamento (painel *Preprocess*)

O painel de pré-processamento (*Preprocess*) do AggloCluster permite a importação do conjunto de padrões por meio da leitura de arquivos texto em formato ARFF (*Attribute-Relation File Format*) bem como a visualização dos padrões em duas dimensões no plano cartesiano, como mostrado na Figura 5.1. Se os padrões

possuírem mais de duas dimensões, então são plotados os dois primeiros atributos de cada padrão (com a possibilidade da seleção dos atributos ser modificada posteriormente pelo usuário).

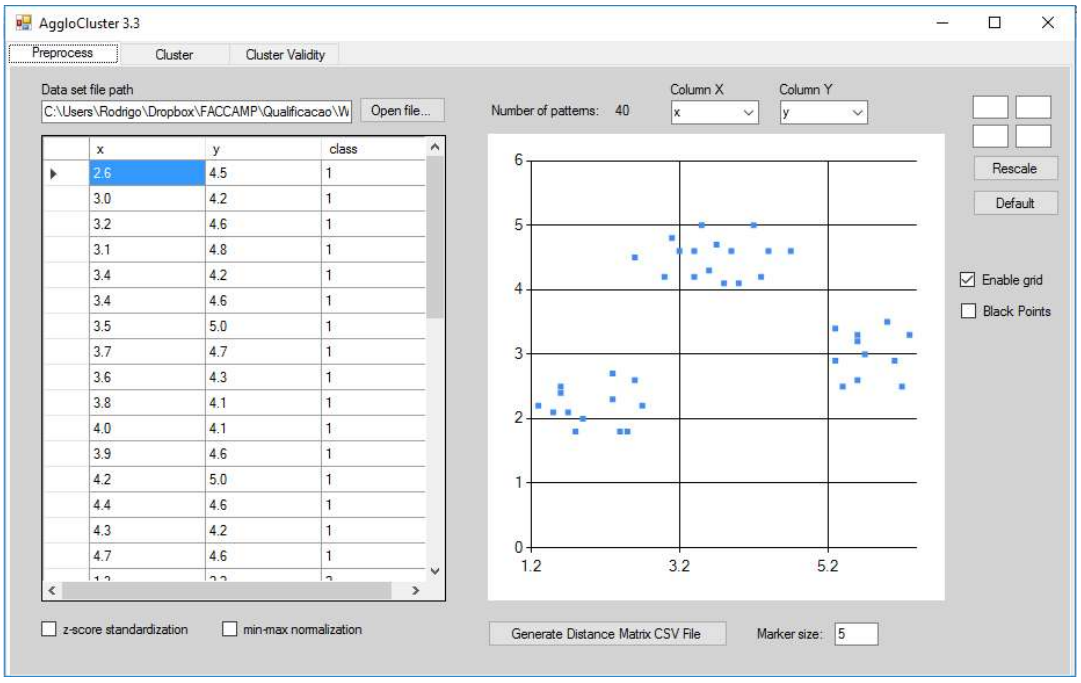


Figura 5.1 Tela de Pré-processamento do AggloCluster.

Como pode ser observado no canto inferior esquerdo da Figura 5.1, existem duas opções para normalização de padrões. O *checkbox z-score standardization* permite realizar a transformação dos dados por meio de uma técnica conhecida como padronização. Outra técnica disponível para transformação dos dados, que realiza a normalização dos valores de atributos trazendo-os para uma determinada faixa (*feature scaling*), é conhecida como Min-Max e pode ser acessada pelo *checkbox min-max normalization*. Tais técnicas de padronização e normalização foram apresentadas no Capítulo 2.

5.2 Módulo de Agrupamento (painel *Cluster*)

Este módulo disponibiliza a implementação do MUAS e, também, as estratégias de agrupamento citadas anteriormente. Uma vez selecionada a estratégia de agrupamento e acionado o botão *Start Clustering*, o sistema executa o algoritmo e plota o seu resultado em uma árvore (*treeview*) com a hierarquia de grupos como mostra a Figura 5.2. A representação visual do agrupamento hierárquico na forma de

um dendrograma pode ser obtida acionando o botão *Show Dendrogram*, como mostra a Figura 5.3.

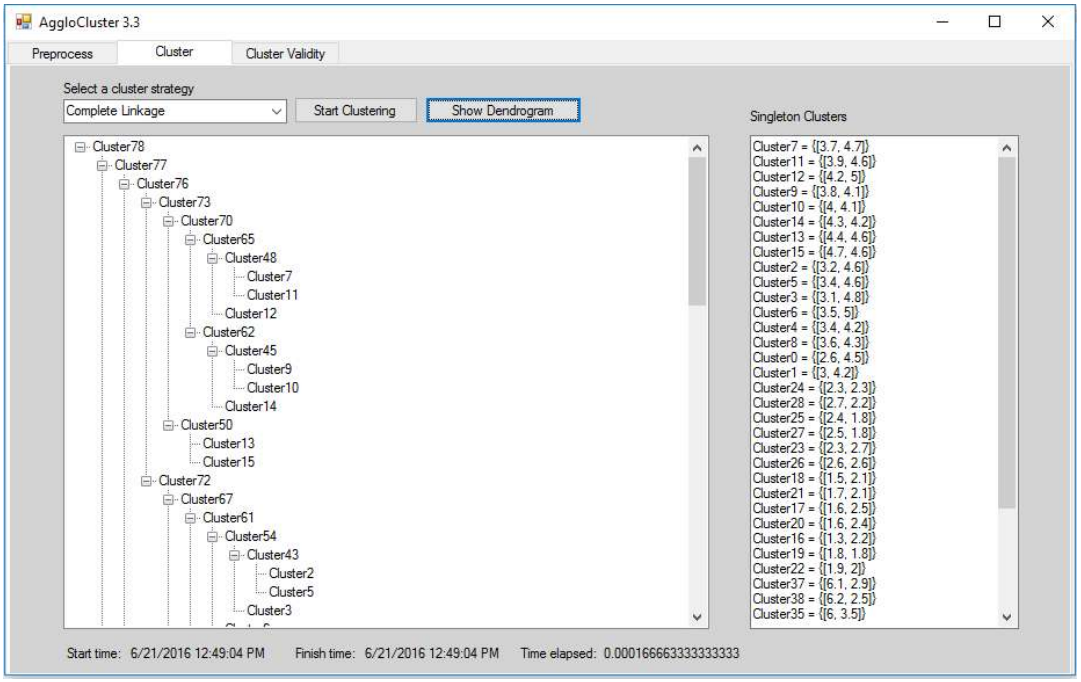


Figura 5.2 Resultado da execução do MUAS utilizando como estratégia de agrupamento o *Complete Linkage*.

É importante observar que o AggloCluster somente habilita o painel *Cluster* se um conjunto de padrões foi previamente importado para o sistema por meio do painel *Preprocess*. O mesmo comportamento ocorre com o painel *Cluster Validity* apresentado na Seção 5.3.

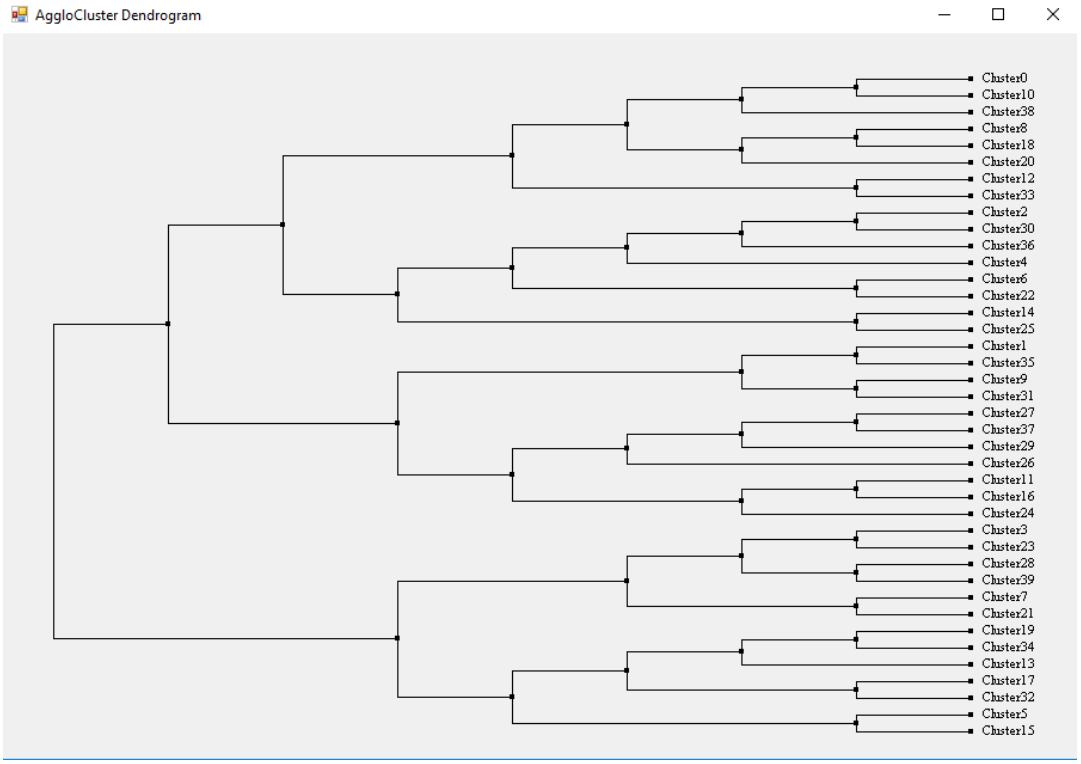


Figura 5.3 Dendrograma resultante da execução do MUAS utilizando o método *Complete Linkage* como estratégia de agrupamento.

5.3 Módulo de Validação (Painel *Cluster Validity*)

O painel *Cluster Validity* disponibiliza recursos para realizar a validação interna de agrupamentos que já foram induzidos. Neste módulo estão disponíveis os índices de validação interna (índice de Dunn e índice Davies-Bouldin) e de validação externa (índice de Rand e índice de Jaccard), como mostra a Figura 5.4.

O módulo de validação disponibiliza, também, o algoritmo particional K-Means, apresentado no Capítulo 4, cujos agrupamentos por ele produzido servirão de *baseline* na comparação dos resultados produzidos pelo algoritmos aglomerativos. A Figura 5.5 mostra o resultado do K-Means com a formação de 3 grupos (*Cluster0*, *Cluster1* e *Cluster2*) para o mesmo conjunto de padrões mostrado na Figura 5.1. Note que este módulo não se restringe apenas à validação, mas também permite que os agrupamentos resultantes dos algoritmos (AGNES e K-Means) sejam visualmente comparados.

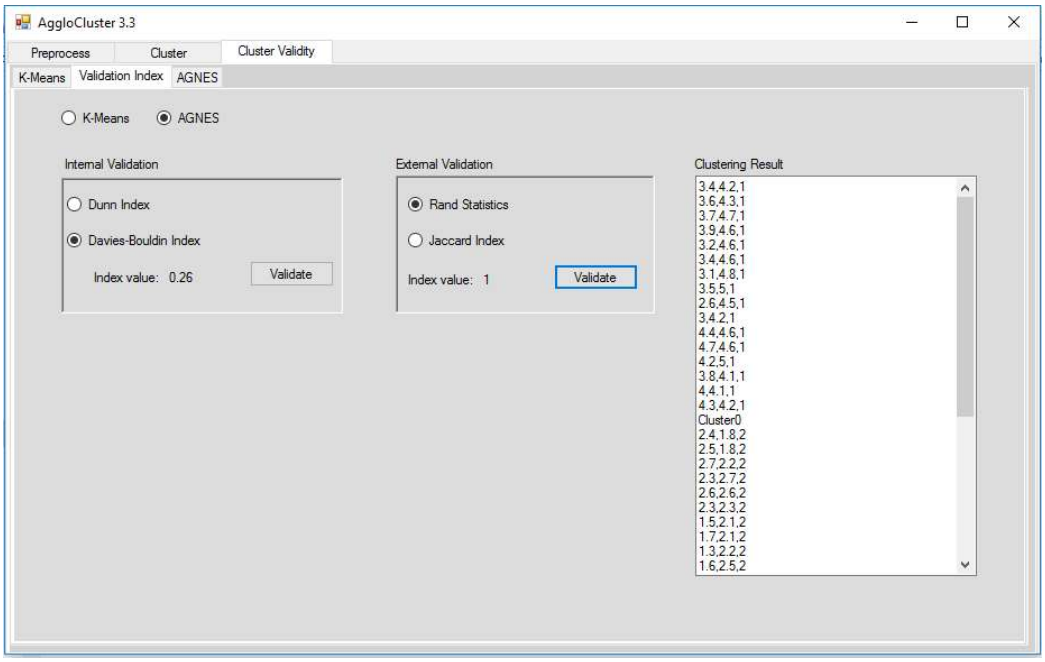


Figura 5.4 Resultado do cálculo do índice de Davies-Bouldin e índice de Rand no agrupamento resultante do AGNES.

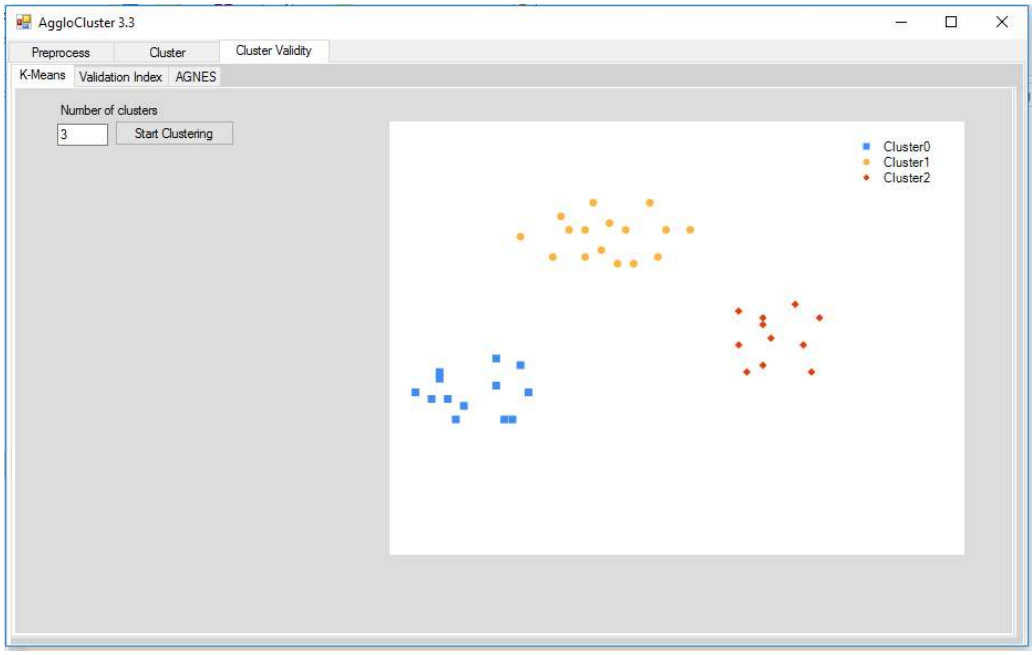


Figura 5.5 Resultado do algoritmo K-Means com a formação de 3 grupos (*Cluster0*, *Cluster1* e *Cluster2*).

O algoritmo AGNES [Kaufman & Rousseeuw 2005], algoritmo aglomerativo baseado no esquema MUAS, foi implementado, mas não em sua forma clássica, ou seja, o ponto de parada do algoritmo é uma quantidade k de grupos informada e não

um único grupo contendo todos os demais subgrupos, como na forma clássica. A Figura 5.6 mostra o resultado do agrupamento produzido pelo AGNES em que foi informada a quantidade 3 de grupos e a estratégia *Complete Linkage*.

Figura 5.6 Execução do algoritmo AGNES resultando na formação de 3 grupos (*Cluster0*, *Cluster1* e *Cluster2*).

É importante notar que tanto o esquema MUAS quanto o algoritmo AGNES realizam a tarefa de agrupamento criando uma estrutura hierárquica de grupos aninhados. Entretanto, para que os grupos produzidos pelo AGNES possam ser comparados com os grupos produzidos pelo K-Means, foi necessário adaptar a saída do algoritmo AGNES para transformar a hierarquia de grupos (e.g., como mostra a Figura 5.3) em grupos planos (*flat clusters*), ou seja, sem sobreposição de grupos (e.g., como mostra a Figura 5.6).

Capítulo 6

Experimentos e Análise dos Resultados nos Conjuntos de Padrões Sintéticos *Sizes, Square* e *Aggregation*

Este capítulo descreve os experimentos realizados com o algoritmo hierárquico aglomerativo disponível no sistema computacional AggloCluster. Os resultados obtidos dos experimentos são analisados e discutidos, com destaque às estratégias de agrupamento que apresentaram melhor desempenho. Os conjuntos de padrões bidimensionais utilizados nos experimentos são conjuntos artificialmente criados de forma que os grupos sejam visualmente identificáveis por seres humanos.

6.1 Descrição dos Conjuntos de Padrões Utilizados nos Experimentos

Para a realização dos experimentos foram utilizados sete conjuntos de padrões sintéticos, baixados do link: https://github.com/deric/clustering-benchmark. Seis deles (Sizes1, Sizes3, Sizes5, Square1, Square3 e Square5) são inspirados no artigo de [Handl & Knowles 2004] e o sétimo conjunto, nomeado Aggregation, foi inspirado no artigo de [Gionis *et al* 2005].

Os conjuntos de padrões Sizes e Square são descritos por dois atributos com distribuição normal. O número de grupos, os tamanhos dos grupos (i.e., número de padrões por grupo), o vetor de médias e o vetor do desvio padrão, para cada distribuição normal, foram fixados manualmente nos grupos Sizes e Square. Nos conjuntos Square todos os grupos têm o mesmo número de padrões (250); entretanto, os três conjuntos diferem entre si com relação às distâncias entre os respectivos grupos. Os conjuntos de padrões Square foram empregados com o objetivo de

investigar a sensibilidade dos algoritmos aglomerativos com relação à distância e ao incremento da sobreposição inter-grupos.

Nos conjuntos de padrões Sizes, a distância inter-grupos e o desvio padrão são mantidos, enquanto que o tamanho (número de padrões) de cada grupo varia em cada conjunto. Os conjuntos de padrões Sizes foram utilizados para investigar a sensibilidade do algoritmo à grupos de tamanhos diferentes.

O conjunto Aggregation é formado por sete grupos de padrões, com formas visualmente distintas. As características particulares desse conjunto de padrões são reconhecidas, na literatura, por dificultarem a abordagem aglomerativa de agrupamento, tais como existência de 'pontes' entre grupos, grupos de diferentes diâmetros, com diferentes formas, etc. A Tabela 6.1 apresenta uma breve descrição dos sete conjuntos utilizados nos experimentos.

Tabela 6.1 Resumo dos 7 conjuntos de padrões sintéticos utilizados nos experimentos. #NP: número de padrões, #NG: número de grupos. (*) Número de padrões considerando a numeração dos grupos.

Conjunto de Padrões	#NP	#NG	Tamanhos dos Grupos
Sizes1	1000	4	400-200-200-200
Sizes3	1000	4	667-111-111-111
Sizes5	1000	4	769-77-77-77
Square1	1000	4	250-250-250-250
Square3	1000	4	250-250-250-250
Square5	1000	4	250-250-250-250
Aggregation	788	7	45-170-102-273-34-130-34(*)

A Figura 6.1 mostra os três conjuntos de padrões Sizes plotados no plano cartesiano. Na Figura 6.2 são apresentados os três conjuntos de padrões Squares, e na Figura 6.3 o conjunto de padrões Aggregation. A plotagem foi realizada por meio do módulo de pré-processamento do sistema computacional AggloCluster.

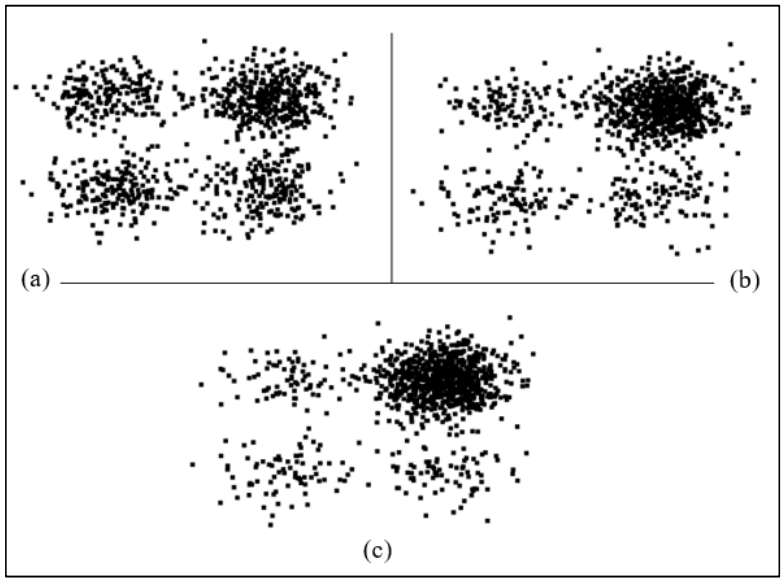


Figura 6.1 Conjuntos de padrões sintéticos (a) Sizes1, (b) Sizes3 e (c) Sizes5.

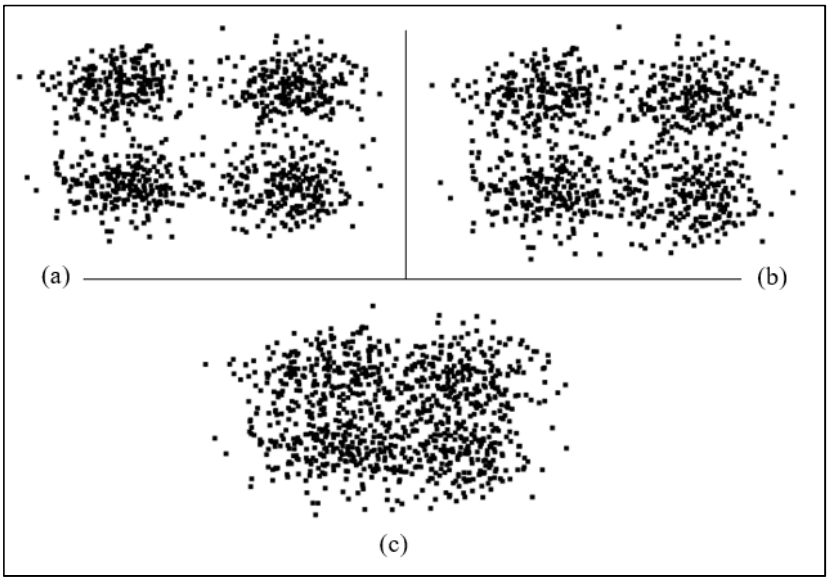


Figura 6.2 Conjuntos de padrões sintéticos (a) Square1, (b) Square3 e (c) Square5.

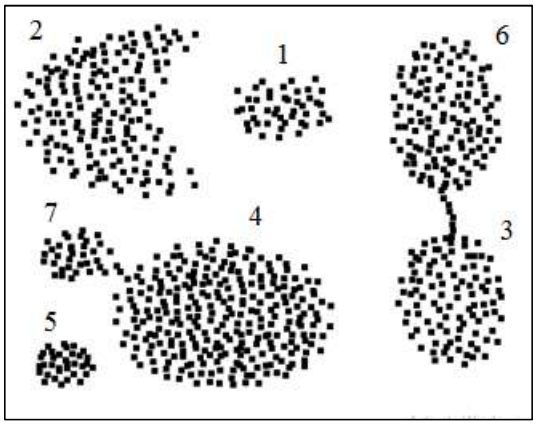


Figura 6.3 Conjunto de padrões Aggregation inspirado no usado em Gionis *et al.* (2005), cujos sete grupos estão numerados, para futura referência a eles.

6.2 Metodologia Utilizada para a Condução dos Experimentos

Nos experimentos realizados com o algoritmo AGNES e o algoritmo K-means foram utilizados os sete conjuntos de padrões, apresentados na Seção 6.1. Todos os padrões que descrevem os conjuntos de padrões consistem de dois atributos numéricos; a distância euclidiana foi a única função de dissimilaridade utilizada para calcular a distância entre tais padrões. Para uma comparação 'justa' entre o algoritmo aglomerativo e o algoritmo K-means, ambos foram executados usando o mesmo valor para o parâmetro k (número de grupos).

Em testes realizados previamente ao experimento, foi observado que o pré-processamento dos conjuntos de padrões, tanto por meio da padronização *z-score* [Jain & Dubes 1988] quanto da normalização *min-max* [Berthold *et al.* 2010], não impactaram, substancialmente, nos resultados dos agrupamentos obtidos. Diante deste fato, foi decidido manter os valores dos atributos em sua forma original.

O objetivo dos experimentos é avaliar o desempenho do algoritmo aglomerativo AGNES [Kaufman & Rousseeuw 2005] que faz uso do método UPGMA (*Unweighted Pair Group Method with Arithmetic Mean*) para o cálculo da distância entre grupos. Os experimentos foram realizados com conjuntos de padrões que têm características diferentes quanto à distância inter-grupos, tamanho (número de padrões), diâmetro e forma dos grupos.

A avaliação dos resultados dos experimento foi realizada por meio de análises dos dois índices de validação interna, o índice de Dunn (D) e o de Davies-Bouldin

(DB), descritos no Capítulo 3. Também, foram realizadas validações externas, utilizando os resultados do índice de Rand (R) e índice de Jaccard (J), brevemente descritos no Capítulo 3. O algoritmo aglomerativo AGNES foi executado apenas uma vez para cada conjunto de padrões. Nos experiementos realizados com o K-means, que é um algoritmo cujos resultados variam de acordo com a escolha inicial dos centróides, o algoritmo foi executado cinco vezes para cada conjunto e foram reportados os melhores e piores resultados.

6.3 Experimentos e Análise de Resultados

Nesta seção são descritos os passos executados em cada experimento. Ao final da seção são apresentados os resultados obtidos das execuções do algoritmo aglomerativo AGNES e K-Means e, também, uma discussão sobre cada um dos resultados obtidos.

Os passos executados nos experimentos com o algoritmo aglomerativo foram os mesmos para todos os conjuntos de padrões, conforme descrito a seguir:

- Importação do conjunto de padrões por meio do módulo *Preprocess* do sistema AggloCluster.
- Após a importação do conjunto, os padrões são plotados em um plano cartesiano. Em seguida é realizada uma inspeção visual para confirmação da quantidade de grupos no conjunto. A quantidade 'real' de grupos é obtida por meio da coluna *class*, descrita nos arquivos dos conjuntos de padrões.
- O algoritmo aglomerativo AGNES é executado passando o número de grupos contidos no agrupamento.
- Após a execução do algoritmo, o agrupamento resultante é validado por meio dos resultados dos índices D e DB e, também, por meio da validação externa (valores do índice R e do índice J).

A seguir são mostradas as figuras referentes aos agrupamentos resultantes de cada experimento realizado com o algoritmo aglomerativo e, na sequência, a análise dos resultados. As figuras 6.4, 6.5 e 6.6 mostram os agrupamentos resultantes com os conjuntos Sizes1, Sizes3 e Sizes5 respectivamente.

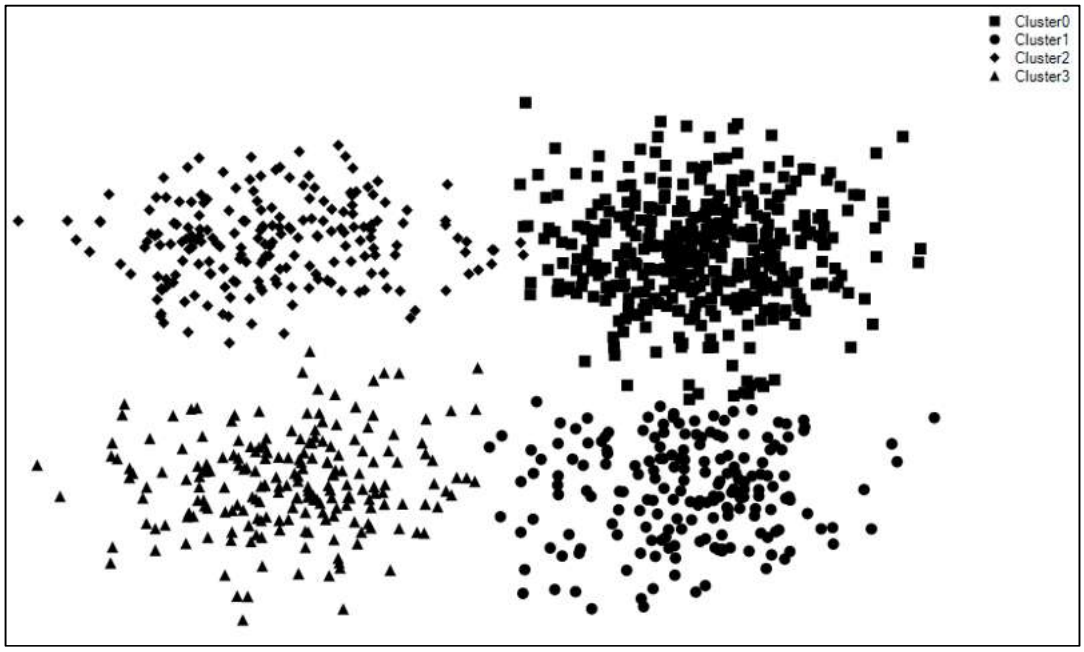


Figura 6.4 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto de padrões Sizes1.

Como pode ser percebido na Figura 6.4, o algoritmo AGNES identificou os quatro grupos visualmente perceptíveis, do conjunto de padrões Sizes1 com um grau de precisão próximo a 1 (em uma escala [0, 1]) conforme resultados dos índices R e J mostrados na Tabela 6.2.

Na Figura 6.5 é mostrado o agrupamento resultante da execução do algoritmo AGNES tendo como entrada o conjunto de padrões Sizes3. Note que as variações dos tamanhos e densidades dos grupos não alteram substancialmente os resultados dos índices R e J. O agrupamento resultante do algoritmo AGNES tendo como entrada o conjunto de padrões Sizes5 é mostrado na Figura 6.6. Assim como constatado no experimento com o conjunto de padrões Sizes3, a alteração dos tamanhos (sem alterar o diâmetro) e das densidades dos grupos não mudaram substancialmente a qualidade do agrupamento.

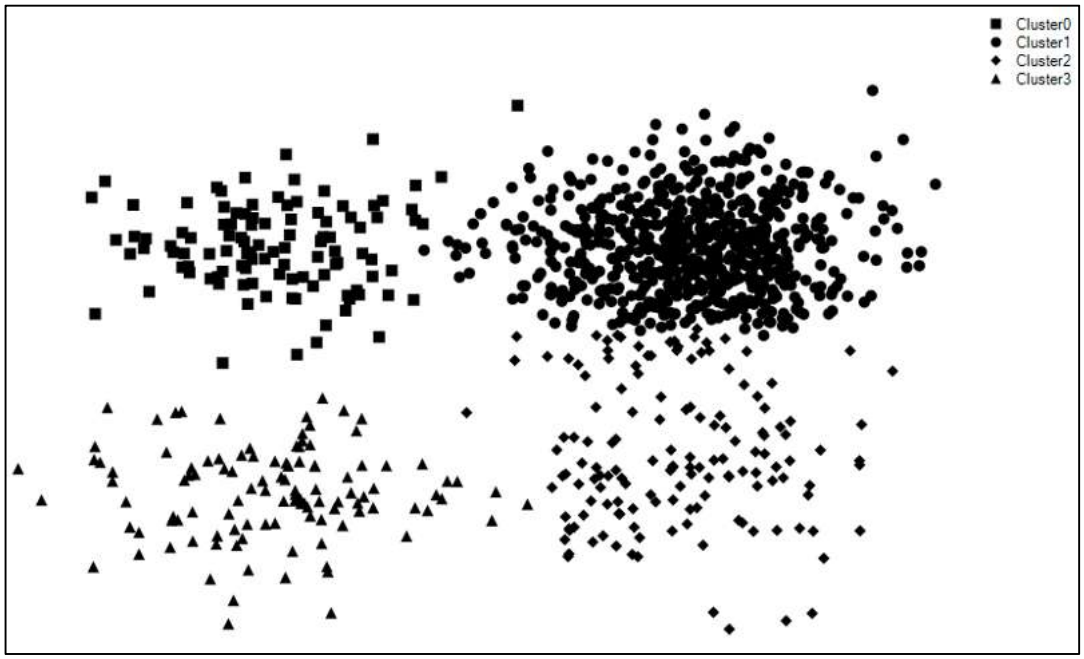


Figura 6.5 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto de padrões Sizes3.

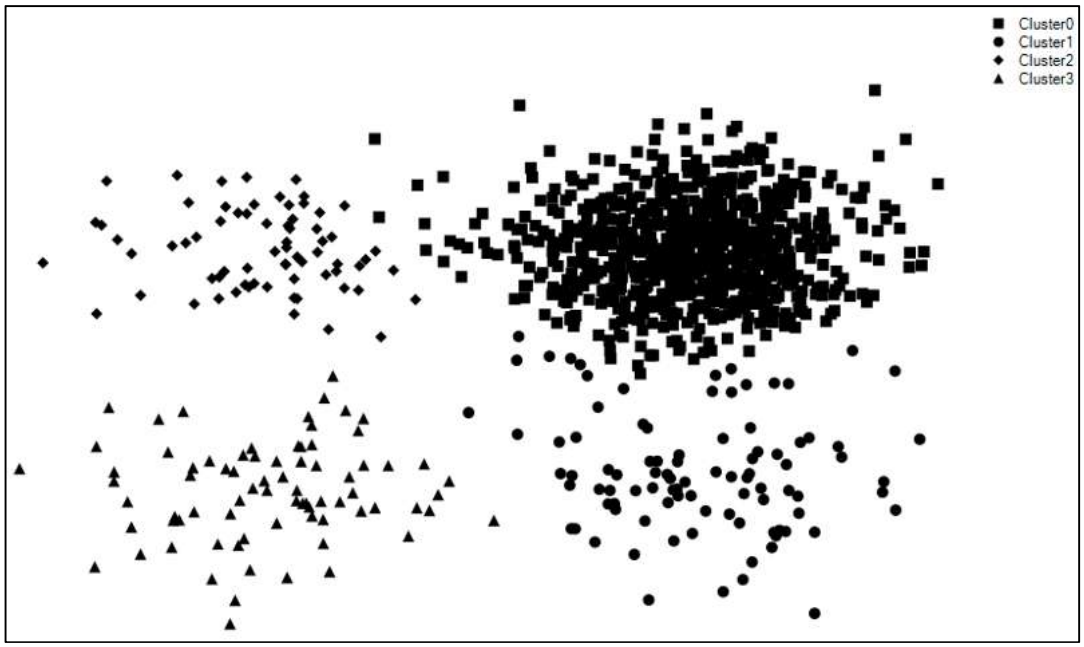


Figura 6.6 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto de padrões Sizes5.

A Tabela 6.2 mostra um resumo dos resultados dos experimentos realizados com o algoritmo AGNES nos conjuntos de padrões Sizes1, Sizes3 e Sizes5. A mesma tabela apresenta, também, o resultado do algoritmo particionante K-means que permite comparar o desempenho de ambos os algoritmos.

Tabela 6.2 Valores dos índices D, DB, R e J nos agrupamentos obtidos pelo AGNES e K-means, no conjunto de padrões Sizes. ($^{+}$) Melhores resultados do K-Means. ($^{-}$) Piores resultados do K-Means.

Algoritmo	Conjunto	D	DB	R	J
AGNES	Sizes1	0,04	0,43	0,974	0,912
AGNES	Sizes3	0,03	0,51	0,936	0,872
AGNES	Sizes5	0,05	0,44	0,963	0,940
K-means ($^{+}$)	Sizes1	0,03	0,40	0,982	0,938
K-means ($^{-}$)	Sizes1	0,03	0,47	0,972	0,941
K-means ($^{+}$)	Sizes3	0,03	0,43	0,972	0,941
K-means ($^{-}$)	Sizes3	0,03	0,47	0,972	0,941
K-means ($^{+}$)	Sizes5	0,02	0,46	0,966	0,945
K-means ($^{-}$)	Sizes5	0	0,69	0,688	0,499

Analisando os dados da Tabela 6.2, o nível de acerto (como mostram os índices R e J) do algoritmo de agrupamento na identificação dos quatro grupos nos conjuntos Sizes1, Sizes3 e Sizes5 foi próximo a 100%. Estes resultados sugerem que o algoritmo AGNES não é sensível a grupos de diferentes tamanhos, pois o incremento gradual no tamanho de um dos quatro grupos não modificou substancialmente o seu desempenho. É importante mencionar que a alteração na quantidade de padrões dos grupos não alterou seus diâmetros. Algoritmos como o K-means e o uso da estratégia *Complete Linkage* em algoritmos aglomerativos tendem a 'quebrar' grupos 'grandes' (de maior diâmetro) em grupos menores baseando-se nos diâmetros dos grupos menores.

As figuras 6.7, 6.8 e 6.9 mostram os agrupamentos resultantes da execução do algoritmo AGNES usando como entrada os conjuntos Square1, Square3 e Square5, respectivamente.

A Tabela 6.3 apresenta um resumo dos resultados dos experimentos realizados com o algoritmo aglomerativo AGNES e o algoritmo K-means.

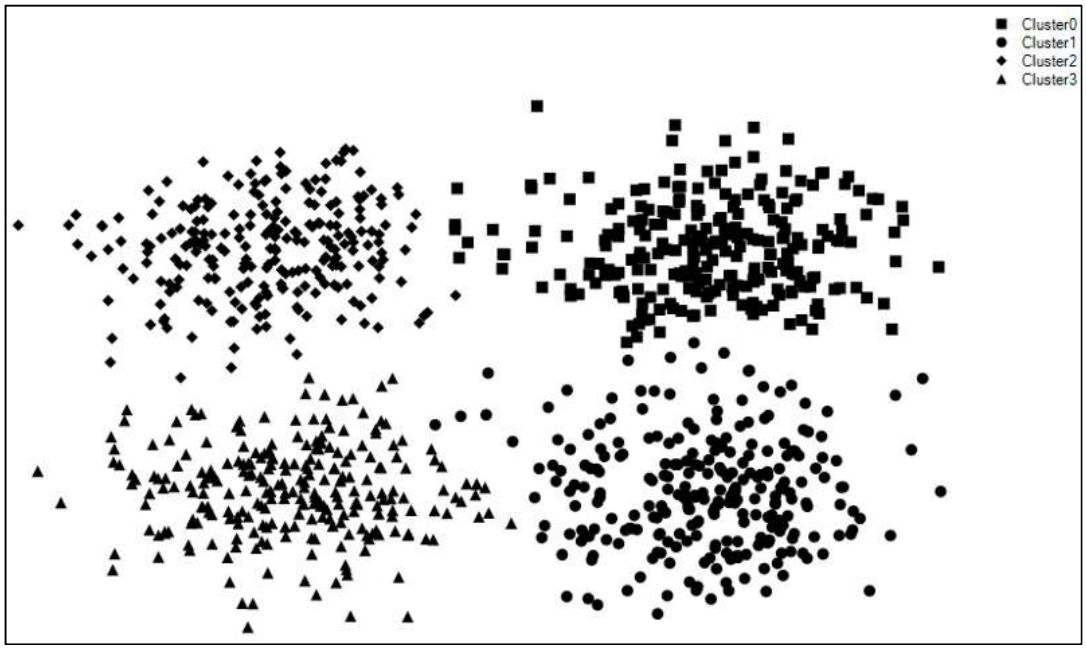


Figura 6.7 Plotagem do agrupamento produzido pelo algoritmo aglomerativo AGNES no conjunto de padrões Square1.

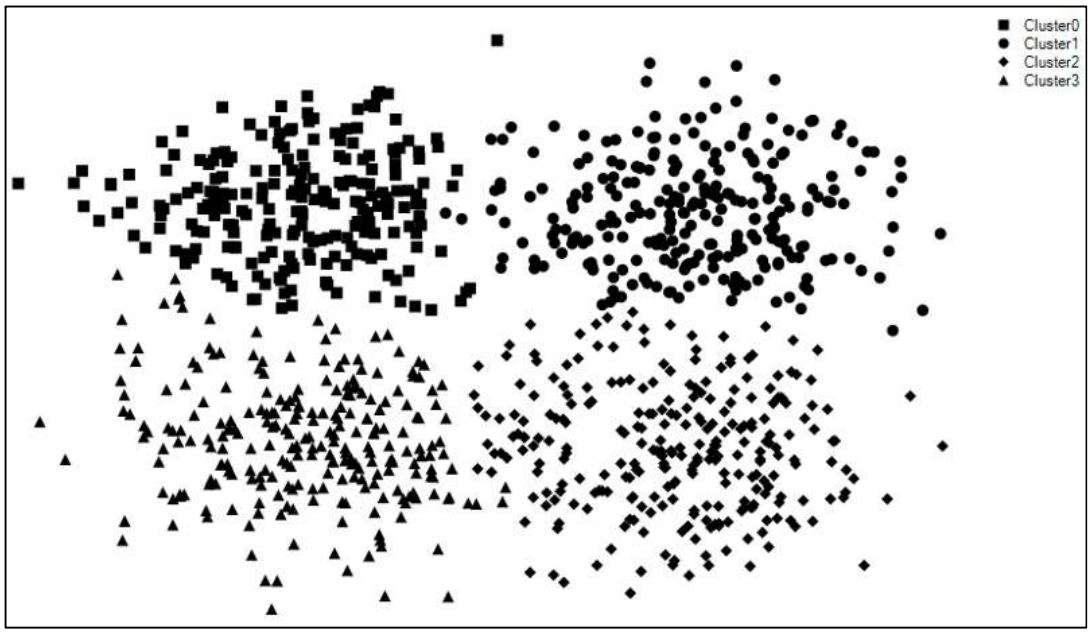


Figura 6.8 Plotagem do agrupamento produzido pelo algoritmo aglomerativo AGNES no conjunto de padrões Square3.

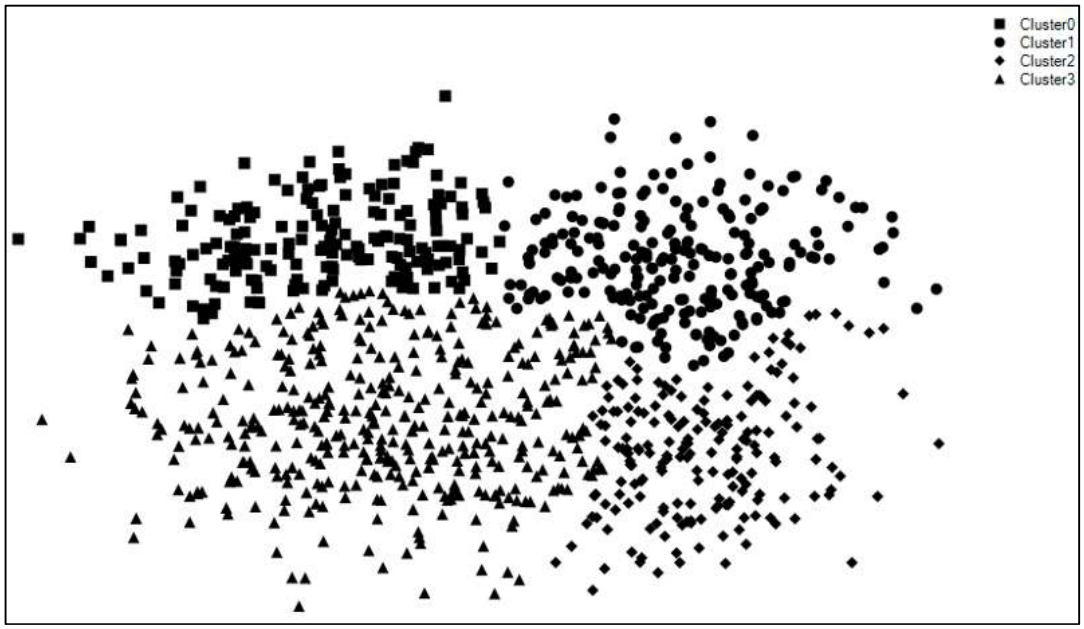


Figura 6.9 Plotagem do agrupamento produzido pelo algoritmo aglomerativo AGNES no conjunto de padrões Square5.

Como pode ser observado na Tabela 6.3, à medida que a distância inter-grupos foi sendo reduzida, o desempenho dos algoritmos piorou. Os resultados dos experimentos com os conjuntos Square sugerem que tanto o algoritmo aglomerativo AGNES quanto o algoritmo K-means são sensíveis à diminuição da distância inter-grupos.

Tabela 6.3 Valores dos índices D, DB, R e J nos agrupamentos obtidos pelo AGNES e K-Means, nos três conjuntos de padrões Squares. ($^+$) Melhores resultados do K-Means. ($^-$) Piores resultados do K-Means.

Algoritmo	Conjunto	D	DB	R	J
AGNES	Square1	0,06	0,42	0,974	0,903
AGNES	Square3	0,03	0,55	0,934	0,768
AGNES	Square5	0,02	0,79	0,789	0,414
K-means($^+$)	Square1	0,05	0,40	0,980	0,924
K-means($^-$)	Square1	0,05	0,42	0,980	0,924
K-means($^+$)	Square3	0,01	0,52	0,950	0,818
K-means($^-$)	Square3	0,01	0,52	0,947	0,809
K-means($^+$)	Square5	0,01	0,61	0,887	0,605
K-means($^-$)	Square5	0,01	0,58	0,880	0,611

Nos experimentos realizados com o conjunto de padrões Aggregation, o processo foi realizado de forma diferente dos experimentos discutidos anteriormente. A ideia, neste experimento, é avaliar o grau de interferência que um determinado grupo tem sobre o desempenho do algoritmo aglomerativo no processo de agrupamento com os sete grupos. Entre os resultados apresentados por [Gionis *et al*.,

2005], o algoritmo aglomerativo *Average Linkage* foi o que mostrou o melhor desempenho entre os algoritmos investigados (*Single Linkage*, *Complete Linkage*, *Average Linkage*, *Ward's clustering* e K-Means), mas não foram investigadas quais características do conjunto de padrões Aggregation contribuíram ou não com o desempenho do algoritmo ou qual configuração de grupos produziria um agrupamento ótimo por parte do algoritmo. O experimento foi iniciado com o conjunto de padrões Aggregation contendo apenas os grupos de padrões de número 1 e 6 (como mostra a Figura 6.10) e, gradualmente, incluindo os demais grupos.

A Figura 6.10 apresenta o resultado do agrupamento realizado pelo algoritmo AGNES no conjunto de padrões Aggregation contendo apenas os padrões dos grupos 1 e 6. Os grupos 1 e 6 são diferentes em relação aos seus tamanhos, diâmetros e formas, mas estão bem separados e a distância entre eles favorece o desempenho dos algoritmos AGNES e K-means.

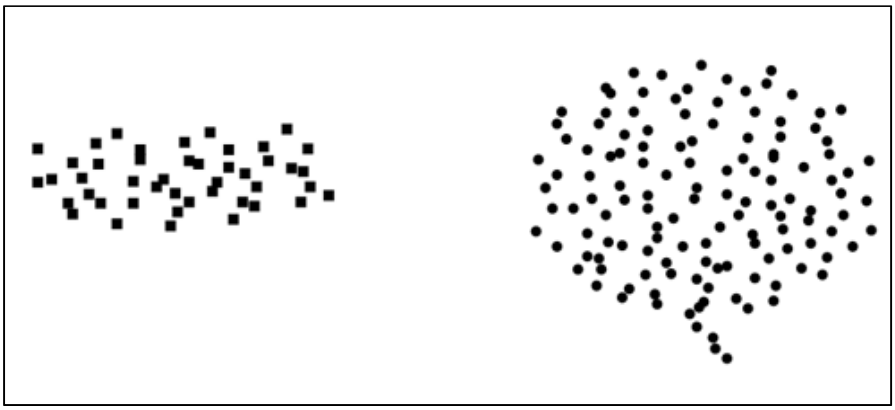


Figura 6.10 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto Aggregation constituído apenas dos grupos 1 e 6.

Como pode ser confirmado nos resultados mostrados na Tabela 6.4, os algoritmos AGNES e K-means identificaram corretamente os dois grupos.

Tabela 6.4 Valores dos índices D, DB, R e J nos agrupamentos obtidos pelo AGNES e K-Means, no conjunto de padrões Aggregation (contendo apenas os grupos 1 e 6). ($^{+}$) Melhores resultados do K-Means. ($^{-}$) Piores resultados do K-Means.

Conjunto Aggregation (Grupos 1 e 6)				
Algoritmo	**D**	**DB**	**R**	**J**
AGNES	0,35	0,48	1	1
K-means($^{+}$)	0,35	0,48	1	1
K-means($^{-}$)	0,35	0,48	1	1

O agrupamento apresentado na Figura 6.11 é resultante do experimento do algoritmo aglomerativo no conjunto de padrões Aggregation modificado com a inclusão de mais um grupo (grupo 3). A inclusão do grupo 3 no conjunto Aggregation não diminuiu o desempenho do algoritmo AGNES, mesmo com a 'ponte' entre os grupos 3 e 6.

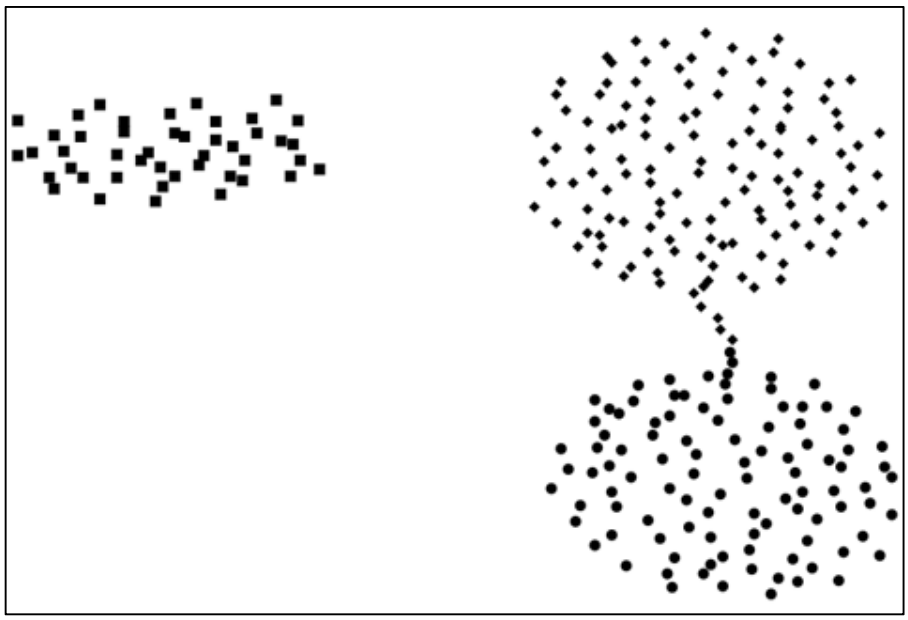


Figura 6.11 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto Aggregation constituído dos grupos 1, 3 e 6.

Como pode ser visto nos valores dos coeficientes R e J mostrados na Tabela 6.5, o algoritmo AGNES identificou corretamente os três grupos. Entretanto o algoritmo K-means obteve menor desempenho pois falhou ao identificar alguns padrões que pertencem ao grupo 3.

Tabela 6.5 Valores dos índices D, DB, R e J nos agrupamentos obtidos pelo AGNES e K-Means, no conjunto de padrões Aggregation (contendo apenas os grupos 1, 3 e 6). ($^{+}$) Melhores resultados do K-Means. ($^{-}$) Piores resultados do K-Means.

Conjunto Aggregation (Grupos 1, 3 e 6)				
Algoritmo	**D**	**DB**	**R**	**J**
AGNES	0,04	0,33	1	1
K-means($^{+}$)	0,04	0,32	0,998	0,969
K-means($^{-}$)	0,04	0,32	0,998	0,969

A Figura 6.12 mostra o agrupamento resultante no conjunto tendo apenas os padrões dos grupos 1 e 2 do conjunto Aggregation. Note que os dois grupos têm formatos bem diferentes, o que pode dificultar o desempenho dos algoritmos na

identificação correta de cada grupo. Como pode ser visto nos resultados mostrados na Tabela 6.6, o algoritmo AGNES identificou corretamente os dois grupos. O algoritmo K-means, entretanto, falha ao identificar alguns padrões do grupo 2.

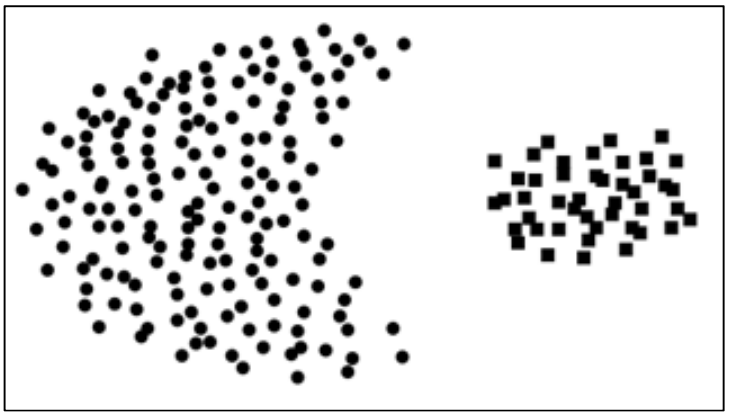


Figura 6.12 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto Aggregation constituído dos grupos 1, e 2.

Tabela 6.6 Valores dos índices D, DB, R e J nos agrupamentos obtidos pelo AGNES e K-Means, no conjunto de padrões Aggregation (contendo apenas os grupos 1 e 2). ($^{+}$) Melhores resultados do K-Means. ($^{-}$) Piores resultados do K-Means.

Conjunto: Aggregation (Grupos 1 e 2)				
Algoritmo	**D**	**DB**	**R**	**J**
AGNES	0,34	0,50	1	1
K-means($^{+}$)	0,04	0,32	0,998	0,969
K-means($^{-}$)	0,08	0.54	0,972	0,959

Ao conjunto de padrões mostrado na Figura 6.12 foi adicionado mais um grupo (grupo 6) e a Figura 6.13 mostra o agrupamento resultante com esta nova configuração.

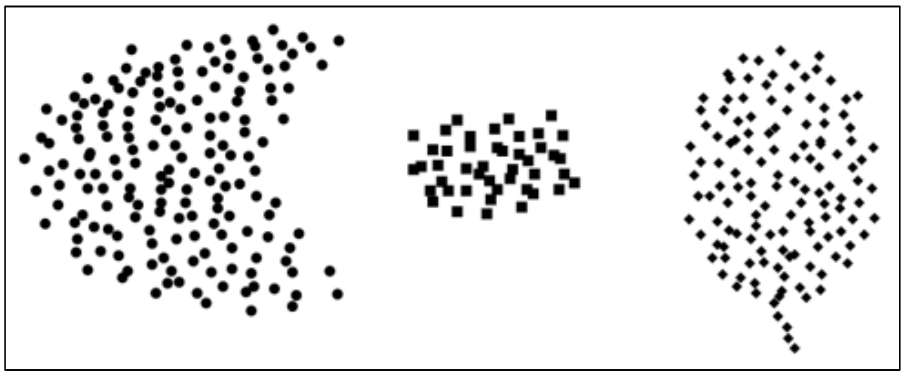


Figura 6.13 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto Aggregation constituído dos grupos 1, 2 e 6.

A Tabela 6.7 mostra os valores dos índices de validação interno e externo para o agrupamento exibido na Figura 6.13. O algoritmo AGNES identificou corretamente os três grupos, entretanto, o algoritmo K-means falhou novamente ao identificar alguns padrões do grupo 2.

Tabela 6.7 Valores dos índices D, DB, R e J nos agrupamentos obtidos pelo AGNES e K-Means, no conjunto de padrões Aggregation (contendo apenas os grupos 1, 2 e 6). ($^+$) Melhores resultados do K-Means. ($^-$) Piores resultados do K-Means.

Conjunto Aggregation (Grupos 1, 2 e 6)				
Algoritmo	**D**	**DB**	**R**	**J**
AGNES	0,33	0,33	1	1
K-means($^+$)	0,08	0,34	0,993	0,982
K-means($^-$)	0,08	0,35	0,989	0,973

A Figura 6.14 mostra o agrupamento resultante tendo como conjunto de entrada o constituído por padrões dos grupos 1, 2, 3 e 6.

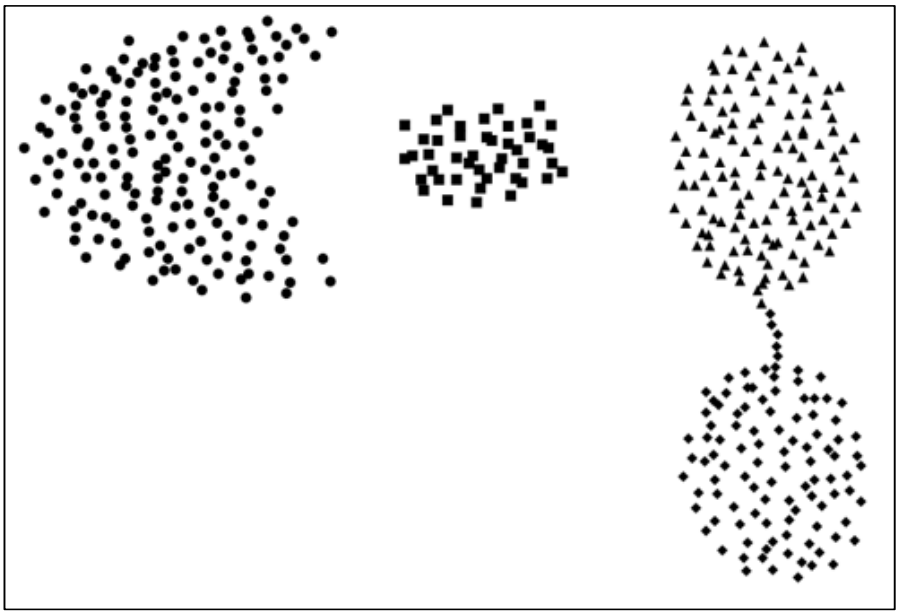


Figura 6.14 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto Aggregation constituído dos grupos 1, 2, 3 e 6.

A inclusão do grupo 3 no conjunto de padrões a serem agrupados, provocou uma redução no desempenho dos dois algoritmos, como mostram os dados da Tabela 6.8. Note na Figura 6.14 que o algoritmo AGNES identificou, incorretamente, alguns padrões do grupo 6 como pertencentes ao grupo 3. Comparando o agrupamento mostrado na Figura 6.14 com o mostrado na Figura 6.11, pode-se perceber que a inclusão do grupo 3 ao conjunto interferiu no cálculo da média (UPGMA) e

consequentemente na incorreta identificação dos padrões do grupo 6. Para confirmar esta hipótese, foi realizado um experimento com o mesmo conjunto, entretanto, utilizando o método WPGMA para cálculo da média da distância inter-grupos. O algoritmo WPGMA identificou corretamente os quatro grupos.

Tabela 6.8 Valores dos índices D, DB, R e J nos agrupamentos obtidos pelo AGNES e K-Means, no conjunto de padrões Aggregation (contendo apenas os grupos 1, 2, 3 e 6). ($^{+}$) Melhores resultados do K-Means. ($^{-}$) Piores resultado do K-Means.

Conjunto Aggregation (Grupos 1, 2, 3 e 6)				
Algoritmo	**D**	**DB**	**R**	**J**
AGNES	0,04	0,41	0,993	0,976
K-means($^{+}$)	0,04	0,36	0,989	0,963
K-means($^{-}$)	0,04	0,36	0,989	0,963

A Figura 6.15 mostra o agrupamento resultante tendo como conjunto de entrada o constituído por padrões dos grupos 1, 2, 3, 4 e 6.

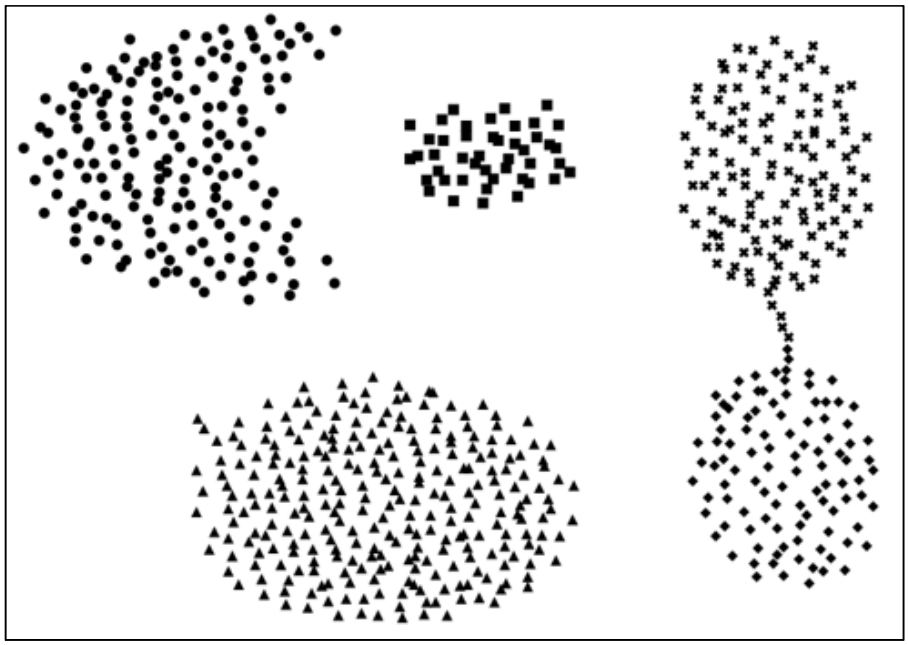


Figura 6.15 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto Aggregation constituído dos grupos 1, 2, 3, 4 e 6.

Observe que, em comparação com o resultado mostrado na Figura 6.14 a alteração introduzida pela inclusão dos padrões do grupo 4, provocou novamente, mudança no valor da distância média inter-grupos, calculada pelo algoritmo AGNES.

A Tabela 6.9 mostra os valores dos índices de validação para o agrupamento resultante da execução dos algoritmos AGNES e K-means. Como comentado no

parágrafo anterior, a inclusão de um novo grupo no conjunto de padrões influenciou no cálculo da distância média inter-grupos executada pelo AGNES o que fez o algoritmo identificar corretamente os cinco grupos.

Tabela 6.9 Valores dos índices D, DB, R e J nos agrupamentos obtidos pelo AGNES e K-Means, no conjunto de padrões Aggregation (contendo apenas os grupos 1, 2, 3, 4 e 6). Melhores resultados do K-Means. (+) Piores resultados do K-Means.

Conjunto Aggregation (Grupos 1, 2, 3, 4 e 6)				
Algoritmo	**D**	**DB**	**R**	**J**
AGNES	0,04	0,39	1	1
K-means(+)	0,03	0,38	0,997	0.987
K-means(-)	0,02	0,51	0,900	0,642

Na Figura 6.15 é mostrado o agrupamento resultante da execução do algoritmo AGNES tendo como entrada os padrões dos grupos 1,2,3,4,5 e 6 do conjunto Aggregation.

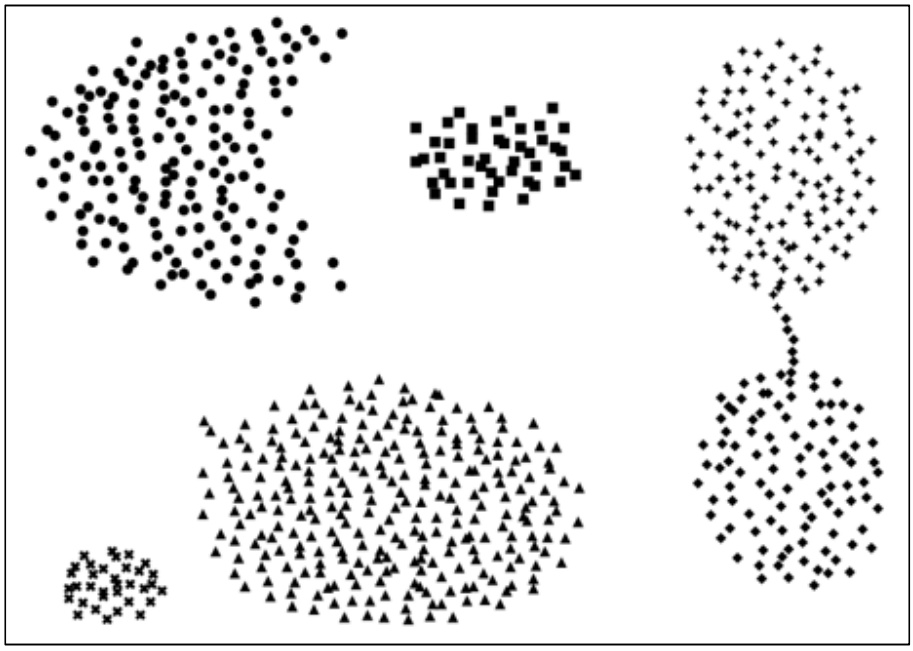


Figura 6.16 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto Aggregation constituído dos grupos 1, 2, 3, 4, 5 e 6.

Conforme mostrado na Figura 6.16 e nos valores dos índices mostrados na Tabela 6.10, o resultado do cálculo da distância média inter-grupos executada pelo algoritmo AGNES foi influenciado pela inclusão do grupo 5 no conjunto de padrões, fazendo com que os padrões do grupo 3 não fossem corretamente identificados como do grupo 3. Outra observação importante é que o K-Means teve desempenho significativamente pior, como mostram os valores de R e J na Tabela 6.10.

Tabela 6.10 Valores dos índices D, DB, R e J nos agrupamentos obtidos pelo AGNES e K-Means, no conjunto de padrões Aggregation (contendo apenas os grupos 1, 2, 3, 4, 5 e 6). Melhores resultados. ($^-$) Piores resultados.

Conjunto Aggregation (Grupos 1, 2, 3, 4, 5 e 6)				
Algoritmo	**D**	**DB**	**R**	**J**
AGNES	0,04	0,34	0,998	0,990
K-means($^+$)	0,03	0,41	0,928	0,709
K-means($^-$)	0,03	0,49	0,889	0,577

A Figura 6.17 mostra o resultado do agrupamento realizado com todos os sete grupos do conjunto Aggregation. Como pode ser percebido na Figura 6.17, a inclusão do grupo 7, não impactou o agrupamento obtido sem tal grupo, como pode ser observado na Figura 6.16. Diferentemente dos algoritmos *Complete Linkage* e WPGMA, o AGNES não tem a tendência de quebrar grupos grandes (i.e., de maior diâmetro), quando no agrupamento existem grupos menores ou de menor diâmetro (como os grupos 5 e 7).

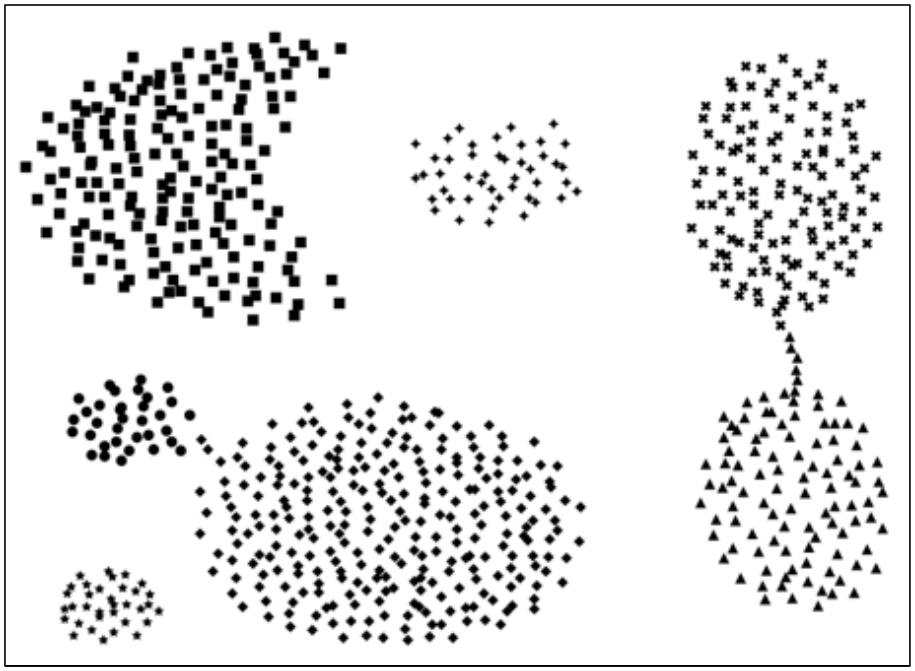


Figura 6.17 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto Aggregation constituído dos grupos 1, 2, 3, 4, 5, 6 e 7.

Analisando os dados da Tabela 6.11, os índices R e J sugerem que o AGNES apresentou um bom desempenho ao identificar, com quase 100% de precisão, os sete grupos do conjunto de padrões Aggregation, por outro lado, o K-Means teve seu desempenho, novamente, insatisfatório.

Tabela 6.11 Valores dos índices D, DB, R e J nos agrupamentos obtidos pelo AGNES e K-Means, no conjunto de padrões Aggregation (contendo os grupos 1, 2, 3, 4, 5, 6 e 7). Melhores resultados. ($^-$) Piores resultados.

Conjunto Aggregation Completo				
Algoritmo	**D**	**DB**	**R**	**J**
AGNES	0,04	0,33	0,998	0,990
K-means($^+$)	0,03	0,44	0,927	0,672
K-means($^-$)	0,03	0,50	0,919	0,649

Concluindo, os experimentos com os conjuntos Size, Square e Aggregation sugerem que o bom desempenho do algoritmo AGNES é mais diretamente influenciado pela distância inter-grupos do que por outras características do conjunto de padrões tais como quantidade de padrões, diâmetro e formas dos grupos (exceto com grupos alongados, em que o método *Single-Linkage* mostra melhor desempenho).

Capítulo 7

Agrupamentos em Conjuntos de Padrões Sintéticos com *outliers* – Experimentos e Análise dos Resultados

Este capítulo descreve os experimentos de agrupamento realizados com o algoritmo AGNES, disponível no sistema computacional AggloCluster, utilizando conjuntos de padrões com a presença de *outliers*. Os resultados obtidos dos experimentos são analisados e discutidos, com destaque às estratégias de agrupamento que apresentaram melhor desempenho. Os conjuntos de padrões bidimensionais utilizados nos experimentos são conjuntos artificialmente criados de forma que os grupos sejam visualmente identificáveis por seres humanos.

7.1 Descrição dos Conjuntos de Padrões Utilizados nos Experimentos

Para a realização dos experimentos foi criado, no ambiente estatístico R (Team, 2005), um conjunto de padrões com três grupos. Este conjunto foi alterado, a cada execução do algoritmo, com o incremento de 1 (um) *outlier*, resultando na criação de cinco conjuntos de padrões distintos, como mostra a Tabela 7.1.

Tabela 7.1 Resumo dos seis conjuntos de padrões sintéticos. #NP: número de padrões, #NG: número de grupos, #Outliers: número de *outliers*.

Conjunto de Padrões	#NP	#NG	Tamanhos dos Grupos	#Outliers
Outliers	300	3	100-100-100	0
Outliers1	301	3	100-100-100	1
Outliers2	302	3	100-100-100	2
Outliers3	303	3	100-100-100	3
Outliers4	304	3	100-100-100	4
Outliers5	305	3	100-100-100	5

A Figura 7.1 mostra os seis conjuntos de padrões *Outliers* (conjunto original, cujos grupos estão identificados como *Cluster1*, *Cluster2* e *Cluster3* para futura referência), *Outliers1*, *Outliers2*, *Outliers3*, *Outliers4* e *Outliers5* plotados no plano cartesiano. A plotagem foi realizada por meio do módulo de pré-processamento do sistema computacional AggloCluster.

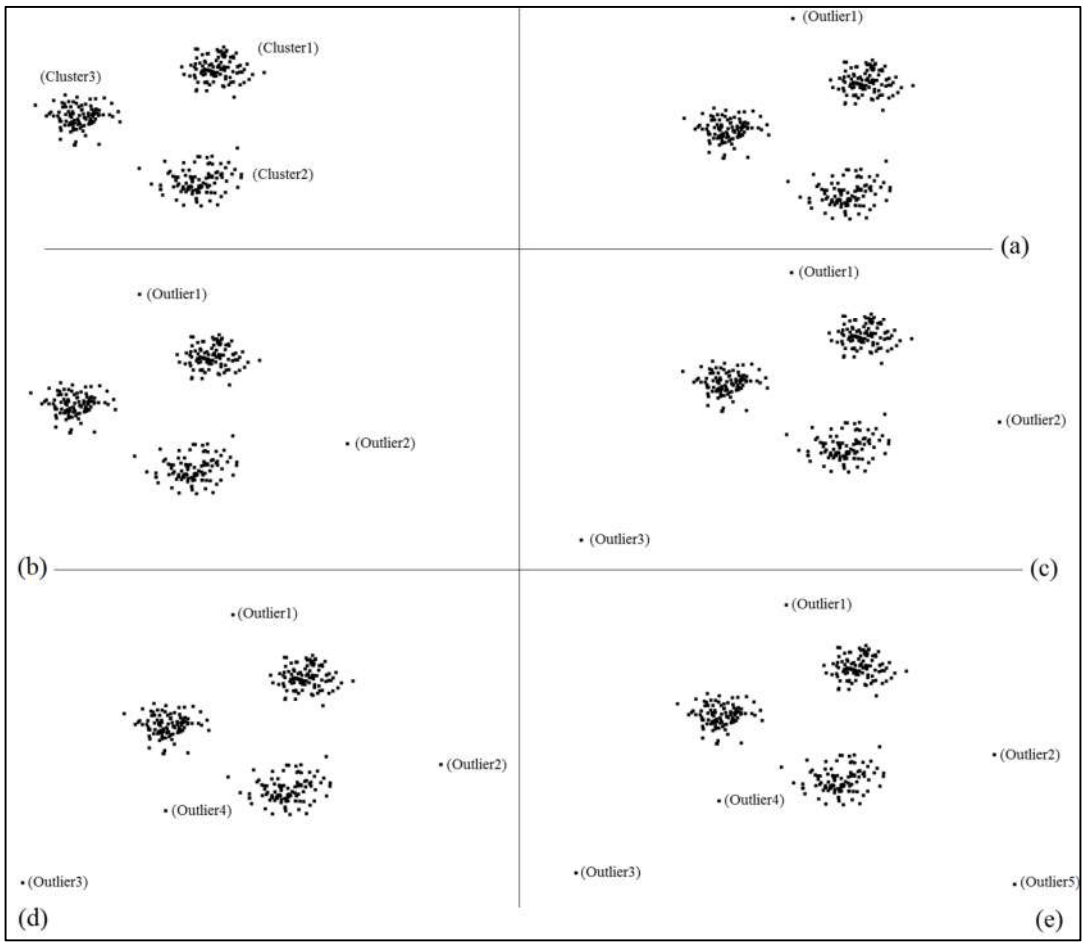


Figura 7.1 Conjuntos de padrões *Outliers* original com grupos identificados para futura referência, sem a presença de *outliers*. (a) *Outliers1,* com a introdução de 1 *outlier* ao conjunto *Outliers*; (b) *Outliers2,* com a adição de 1 *outlier* ao conjunto *Outliers1;* (c) *Outliers3,* com a adição de 1 *outlier ao conjunto Outliers2*; (d) *Outliers4,* com a adição de 1 *outlier* ao conjunto *Outliers3* e (e) *Outliers5,* com a adição de 1 *outlier* ao conjunto *Outliers4*.

7.2 Metodologia Utilizada para a Condução dos Experimentos

A metodologia utilizada nos experimentos segue o mesmo formato da metodologia descrita no Capítulo 6; entretanto, como comentado anteriormente, os experimentos foram conduzidos modificando o conjunto de padrões com o

incremento de 1 (um) *outlier* a cada execução do algoritmo AGNES até que 5 (cinco) *outliers* tenham sido adicionados e, portanto, o AGNES foi executado 5 vezes, cada uma delas com uma das 5 versões do conjunto original *Outliers*.

7.3 Experimentos e Análise de Resultados

Nesta seção são apresentados e discutidos os resultados obtidos das execuções do algoritmo aglomerativo AGNES, tendo em vista seu desempenho em um cenário com ruídos, e do algoritmo particionante K-Means, cujo resultados dos agrupamentos são utilizados como *baseline*. Para facilitar a visualização dos grupos identificados no agrupamento produzido pelo algoritmo AGNES, cada grupo foi marcado com uma linha de contorno tracejada. Na Figura 7.2, por exemplo, são mostrados três grupos. Na sequência, são mostradas as figuras acompanhadas das discussões dos resultados de cada experimento.

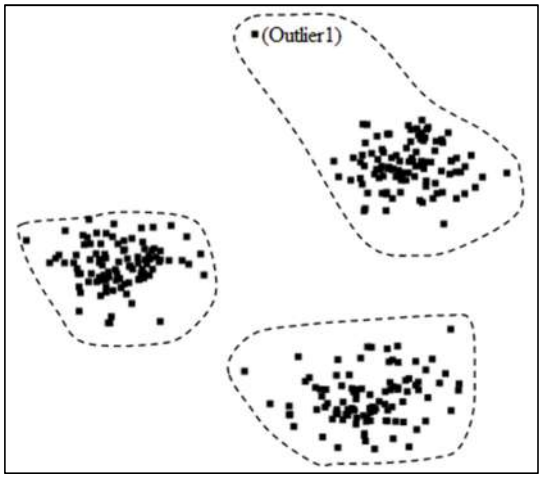


Figura 7.2 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto de padrões mostrado na Figura 7.1(a) (*Outliers1*).

Como mostra a Figura 7.2, a inclusão de 1 (um) *outlier* (*Outlier1*) não diminuiu o desempenho do algoritmo AGNES quanto à correta identificação dos três grupos. Note que o algoritmo identificou o *Outlier1* como um padrão pertencente ao grupo *Cluster1*, por ser este grupo o grupo mais próximo do *outlier***.**

No experimento cujo resultado está mostrado na Figura 7.3, o algoritmo AGNES teve o seu desempenho reduzido. A inclusão de um segundo *outlier* (*Outlier2*) no conjunto de padrões influenciou no cálculo da distância média entre grupos o que, por sua vez, resultou na junção de dois grupos (identificados no conjunto *Outliers* original como *Cluster2* e *Cluster3*). Note que o algoritmo AGNES

identificou o *Outlier2* como sendo um grupo *singleton* (grupo contendo um único padrão).

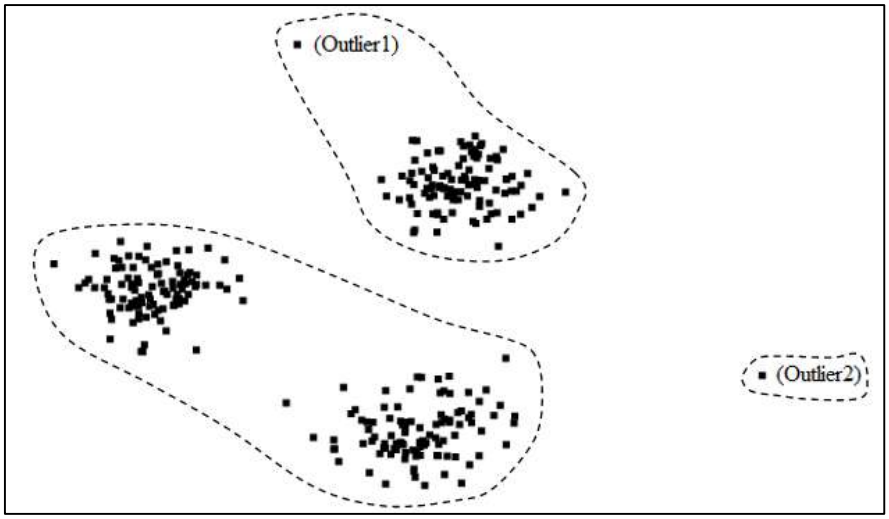


Figura 7.3 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto de padrões mostrado na Figura 7.1(b) *(Outliers2).*

A Figura 7.4 mostra o agrupamento produzido pelo algoritmo AGNES no conjunto de padrões *Outliers,* que tem 3 *outliers.* O novo *outlier* adicionado ao conjunto (comparativamente ao experimento anterior) alterou novamente o resultado do cálculo da distância média entre grupos executado pelo algoritmo. Observe que o agrupamento produzido pelo AGNES produziu um grupo, representando a união dos três grupos de padrões (inicialmente existentes no conjunto *Outliers* como *Cluster1*, *Cluster2* e *Cluster3*) com um *singleton* com o primeiro *outlier* (*Outlier1*) e, os dois outros *outliers* (*Outlier2* e *Outlier3*) formaram dois grupos *singletons*.

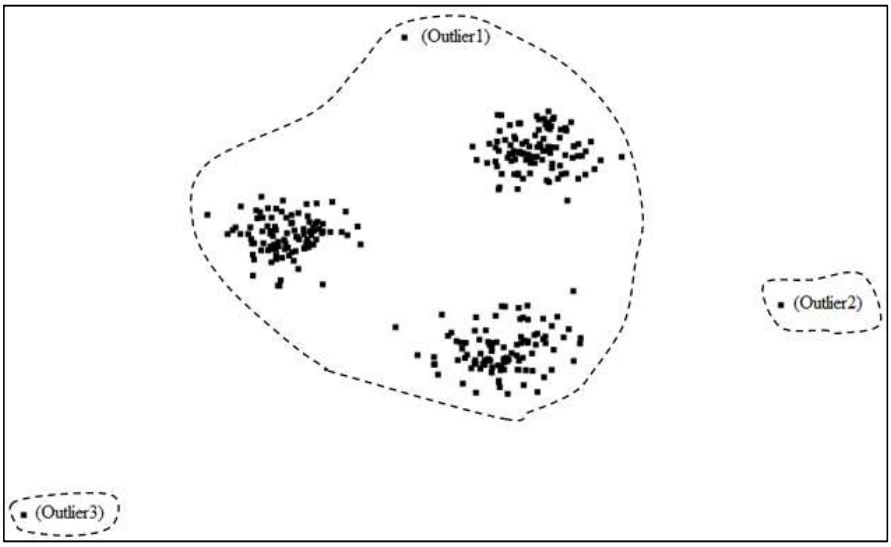


Figura 7.4 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto de padrões mostrado na Figura 7.1(c) (*Outliers3*). Note que o agrupamento tem três grupos; o primeiro inclui os três grupos originais mais o primeiro outlier; os outros dois são grupos *singleton*, cada um deles com um dos dois *outliers* restantes.

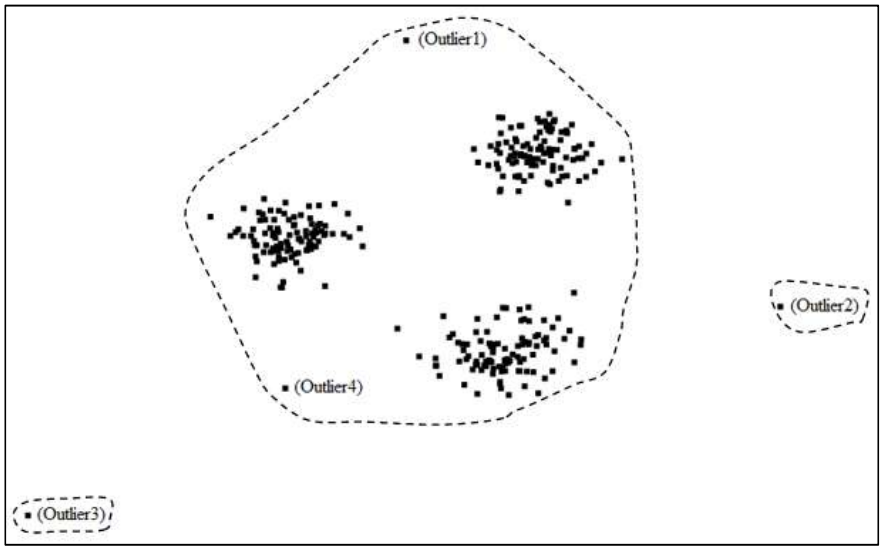


Figura 7.5 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto de padrões mostrado na Figura 7.1(d) (*Outliers4*).

O agrupamento mostrado na Figura 7.5 é muito semelhante ao agrupamento mostrado na Figura 7.4. O quarto *outlier* (*Outlier4*) incluído no conjunto foi incluído no grupo que representa a união dos padrões pertencentes aos três grupos iniciais (*Cluster1*, *Cluster2* e *Cluster3*) unidos ao primeiro *outlier* (*Outlier1*). Cada um dos outros dois grupos, é formado apenas por um *outlier.*

O último experimento foi realizado com o conjunto *Outliers5*, que tem a presença de 5 *outliers*. O agrupamento produzido pelo algoritmo AGNES é mostrado na Figura 7.6, e é constituído por três grupos, sendo o primeiro constituído da união dos três grupos originais (*Cluster1*, *Cluster2* e *Cluster3*) unidos a dois *outliers* (*Outlier1* e *Outlier4*); um segundo é constituído por dois *outliers* (*Outlier2* e *Outlier5*), e um terceiro grupo, *singleton*, formado apenas pelo *Outlier3*.

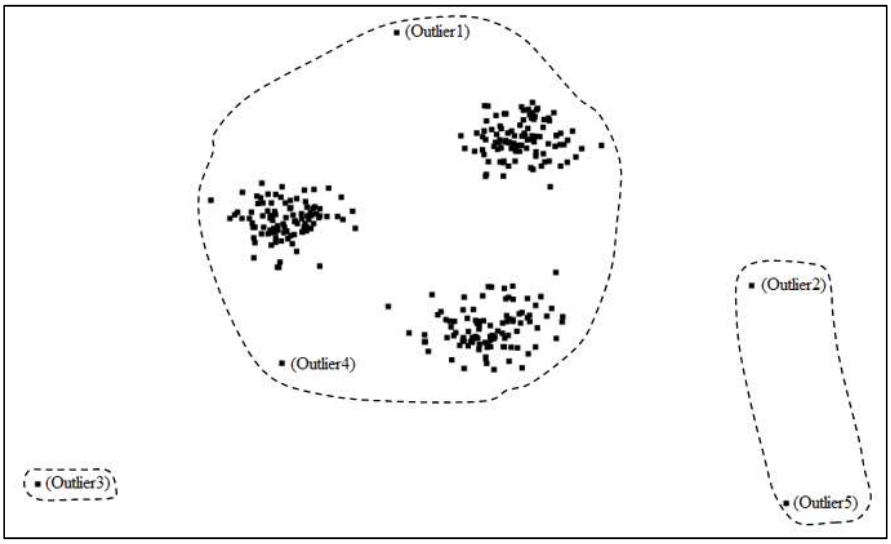


Figura 7.6 Plotagem do agrupamento produzido pelo algoritmo AGNES no conjunto de padrões mostrado na Figura 7.1(e) (*Outliers5*).

A Tabela 7.2 mostra um resumo dos valores obtidos com quatro índices de validação em que é possível observar a influência de *outliers* no desempenho do algoritmo aglomerativo AGNES.

Os valores da Tabela 7.2, assim como a inspeção visual dos agrupamentos, sugerem que o desempenho do algoritmo aglomerativo AGNES é influenciado pela quantidade e localização de *outliers* no conjunto de padrões *Outliers*. Pode-se sugerir também, que quando *outliers* estão localizados 'distantes' dos centros dos grupos, estes *outliers* formam seus próprios grupos.

Tabela 7.2 Resumo dos valores dos índices D, DB, R e J dos experimentos realizados no conjunto de padrões *Outliers*. ($^{+}$) Melhores resultados. ($^{-}$) Piores resultados.

Conjunto: *Outliers*					
Algoritmo	**# *Outliers***	**D**	**DB**	**R**	**J**
AGNES	0	0,50	0,19	1	1
AGNES	1	0,34	0,19	0,998	0,993
AGNES	2	0,30	0,36	0,778	0,595
AGNES	3	0,41	0,25	0,338	0,329
AGNES	4	0,41	0,25	0,335	0,327
AGNES	5	0,41	0,34	0,340	0,327
K-means ($^{+}$)	0	0,50	0,19	1	1
K-means ($^{-}$)	0	0,02	0,84	0,720	0,494
K-means ($^{+}$)	1	0,34	0,19	0,998	0,993
K-means ($^{-}$)	1	0,34	0,19	0,998	0,993
K-means ($^{+}$)	2	0,24	0,19	0,996	0,987
K-means ($^{-}$)	2	0,01	0,64	0,723	0,496
K-means ($^{+}$)	3	0,18	0,20	0,993	0,980
K-means ($^{-}$)	3	0,18	0,20	0,993	0,980
K-means ($^{+}$)	4	0,18	0,20	0,991	0,973
K-means ($^{-}$)	4	0,01	0,71	0,721	0,490
K-means ($^{+}$)	5	0.18	0.21	0.989	0.967
K-means ($^{-}$)	5	0,01	0,65	0,725	0,495

Analisando os resultados dos agrupamentos produzidos pelo K-means, podemos sugerir que seu desempenho é superior, em comparação com o algoritmo AGNES, em conjuntos de padrões com a presença de *outliers*. No melhor caso, quando o K-means identifica corretamente os três grupos nos conjuntos *Outliers1*, *Outliers2*, *Outliers3*, *Outliers4* e *Outliers5*, os *outliers* são incluídos nos grupos mais próximos devido as suas proximidades aos centroides de tais grupos. É importante mencionar que os valores dos índices D e DB não são suficientes para avaliar o aumento ou diminuição do desempenho do algoritmo AGNES, isto porquê tais índices consideram somente informações intrínsecas do agrupamento (isolamento e coesão) e não levam em conta informações externas (como rótulos ou classes).

Capítulo 8

Experimentos com Conjuntos de Padrões Sintéticos *Gestalt* e Análise dos Resultados

Esse capítulo descreve os experimentos realizados com três versões distintas do algoritmo aglomerativo de agrupamento, criadas devido às três diferentes estratégias de medida de distância entre grupos utilizadas, a saber: *Single Linkage* (SL), *Complete Linkage* (CL) e *Average Linkage* (AL), abordadas antes neste documento, no Capítulo 3.

Nos experimentos foram utilizados, em sua maioria, conjuntos de padrões inspirados naqueles apresentados em [Zahn, 1971]. Os resultados obtidos dos experimentos são analisados e discutidos, com destaque nas estratégias de agrupamento que apresentaram melhor desempenho. Os conjuntos de padrões utilizados nos experimentos são conjuntos artificialmente criados de forma que seus grupos sejam visualmente identificáveis por seres humanos.

8.1 Descrição dos Conjuntos de Padrões Utilizados nos Experimentos

Para a realização dos experimentos foram utilizados oito conjuntos de padrões sintéticos, sendo sete destes criados com base no padrão dos grupos dos conjuntos de dados descritos em [Zahn, 1971]. Em sua pesquisa, Zahn (1971) menciona que o método proposto para identificação de grupos levou em conta a forma como a percepção humana é organizada, quando escolheu conjuntos de padrões que refletisse determinados padrões de agrupamento. Conjuntos de padrões bidimensionais, em que padrões se dispõem em grupos separados, nomeados *gestalts*, foram utilizados, cujo princípio utilizado para realizar um agrupamento é a proximidade. Além dos conjuntos caracterizados como *gestalts*, foi também utilizado nos experimentos conduzidos, um conjunto que representa um problema bem conhecido em tarefas de

classificação, aquele que envolve três espirais aninhadas. Os oito conjuntos de padrões utilizados nos experimentos são, brevemente, apresentados a seguir:

- *Dataset Figura 8.1(a)*: Conjunto de padrões com dois grupos, com 20 (vinte) padrões em cada um deles, bem separados, e com densidades e homogeneidades similares. É um fato conhecido que a maioria dos métodos de agrupamento existentes, via de regra, apresentam bom desempenho em tal configuração de grupos [Nagi, 1968].
- *Dataset Figura 8.1(b)*: Conjunto de padrões formado por dois grupos que, juntos, totalizam 40 (quarenta) padrões, com características similares às do conjunto *Dataset Figura 8.1(a)*, exceto, por seus grupos terem quantidades de padrões e densidades diferentes.
- *Dataset Figura 8.1(c)*: Conjunto com 72 (setenta e dois) padrões, divididos em dois grupos, com 26 e 46 padrões, respectivamente, visivelmente bem separados, com volumes e densidades diferentes, cujas respectivas densidades são maiores em seus centros.
- *Dataset Figura 8.1(d)*: Conjunto de padrões formado por dois grupos bem separados, com 100 (cem) padrões em cada um deles, com características similares às descritas no conjunto *Dataset Figura 8.1(a)*, entretanto, com formatos completamente diferentes.
- *Dataset Figura 8.1(e)*: Conjunto com 174 (cento e setenta e quatro) padrões divididos em dois grupos, com 158 e 16 padrões, respectivamente, cujos formatos e volumes são diferentes.
- *Dataset Figura 8.1(f)*: Conjunto com 312 (trezentos e doze) padrões divididos em três grupos, com 101, 105 e 106 padrões, respectivamente, cujos formatos representam, visualmente, três espirais. Este conjunto de padrões não foi utilizado em [Zahn, 1971], entretanto, sua configuração mostrou-se interessante para o propósito desta pesquisa.
- *Dataset Figura 8.1(g)*: Conjunto com 142 (cento e quarenta e dois) padrões divididos em dois grupos, 50 e 92 padrões, respectivamente, que envolve, por parte do algoritmo de agrupamento, detectar gradientes em regiões densas.

- *Dataset Figura 8.1(h)*: Conjunto com 83 (oitenta e três padrões), com características semelhantes à do conjunto do *Dataset Figura 8.1(c)*, exceto pelos grupos estarem bem próximos a ponto de se tocarem, com 39 e 44 padrões, respectivamente.

A Tabela 8.1 mostra um resumo das características dos oito conjuntos de padrões brevemente descritos anteriormente.

Tabela 8.1 Resumo das características dos 8 conjuntos de padrões sintéticos utilizados nos experimentos. #NP: número de padrões, #1º Grupo: número de padrões do primeiro grupo, #2º Grupo: número de padrões do segundo grupo, #3º Grupo: número de padrões do terceiro grupo.

Conjunto de Padrões	#NP	#1º Grupo	#2º Grupo	#3º Grupo
Dataset Figura 8.1(a)	40	20	20	0
Dataset Figura 8.1(b)	40	16	24	0
Dataset Figura 8.1(c)	72	26	46	0
Dataset Figura 8.1(d)	200	111	89	0
Dataset Figura 8.1(e)	174	158	16	0
Dataset Figura 8.1(f)	312	101	105	106
Dataset Figura 8.1(g)	142	50	92	0
Dataset Figura 8.1(h)	83	39	44	0

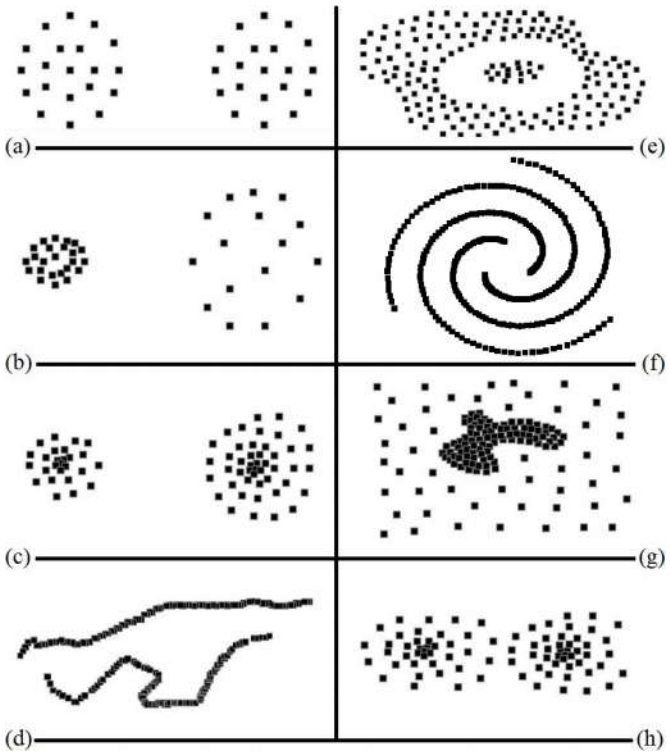


Figura 8.1 Plotagem dos oito conjuntos de padrões sintéticos utilizados nos experimentos.

A Figura 8.1 mostra cada um dos oito conjuntos de padrões plotados no plano cartesiano. A plotagem foi realizada por meio do módulo de pré-processamento do sistema computacional AggloCluster.

8.2 Metodologia Utilizada para a Condução dos Experimentos

Os padrões dos conjuntos descritos na Seção 8.1 são descritos por dois atributos numéricos e a distância Euclidiana foi a única função de dissimilaridade utilizada para calcular a distância entre tais padrões. Para uma comparação 'justa' entre o algoritmo aglomerativo e o algoritmo K-means, ambos foram executados com o mesmo valor para o parâmetro K (número de grupos).

No sistema computacional AggloCluster a estratégia de agrupamento *Average Linkage* (AL) foi disponibilizada com duas implementações para o cálculo da distância média entre grupos: (1) UPGMA (*Unweighted Pair Group Method with Arithmetic Mean*), cujo cálculo é brevemente descrito no Capítulo 3 e utilizado na versão clássica do algoritmo AGNES [Kaufman & Rousseeuw 2005]; (2) WPGMA (*Weighted Pair Group Method with Arithmetic Mean*) que, no cálculo da média, atribui o mesmo peso à padrões de grupos distintos no cálculo da distância média inter-grupos [Everitt *et al.*, 2011].

A avaliação dos resultados obtidos dos experimento foi realizada utilizando-se os valores dos índices de validação Dunn (D), Davies-Bouldin (DB) e Rand (R), que foram abordados no Capítulo 4. Os quatro algoritmos aglomerativos foram executados apenas uma vez em cada conjunto de padrões mostrado na Figura 8.1. Nos experiementos realizados com o K-means, cujos resultados variam de acordo com a escolha inicial dos centróides, o algoritmo foi executado 10 vezes em cada conjunto, entretanto, foram reportados apenas os resultados mais frequentes (i.e., o número de vezes que o mesmo resultado foi obtido).

8.3 Experimentos e Análise de Resultados

Na sequência são apresentados e discutidos os resultados obtidos dos experimentos realizados com os algoritmos aglomerativos e o algoritmo K-Means. Nas tabelas que apresentam os valores dos índices de validação, os melhores resultados estão em negrito. A Figura 8.2 mostra os agrupamentos produzidos pelos algoritmos aglomerativos no conjunto de padrões *Figura 8.1(a)*. As quatro estratégias

de agrupamento (SL, CL, UPGMA e WPGMA) identificaram corretamente os dois grupos e tiveram o mesmo desempenho, como mostram os valores de D = 0,99, DB = 0,38 e R = 1,00 informados na Tabela 8.2.

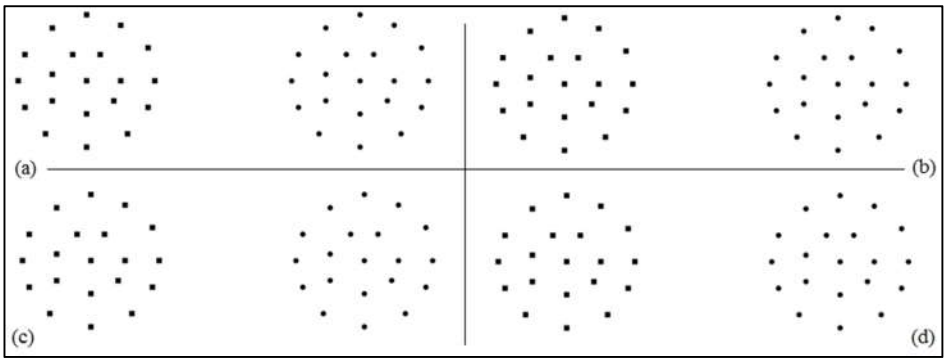


Figura 8.2 Plotagem dos agrupamentos produzidos pelos algoritmos aglomerativos no conjunto de padrões *Figura 8.1(a)*. (a) Agrupamento produzido pela estratégia *Single-Linkage*. (b) Agrupamento produzido pela estratégia *Complete-Linkage*. (c) Agrupamento produzido pela estratégia *Average-Linkage* (UPGMA). (d) Agrupamento produzido pela estratégia *Average-Linkage* (WPGMA).

O agrupamento produzido pelo algoritmo K-Means tem configuração idêntica às dos agrupamentos mostrados na Figura 8.2.

Tabela 8.2 Valores dos índices D, DB e R nos agrupamentos obtidos pelo SL, CL, UPGMA, WPGMA e K-means, no conjunto de padrões *Figura 8.1(a)*.

Estratégia de Agrupamento	# grupos	D	DB	R
Single-Linkage	2	**0,99**	**0,38**	**1,00**
Complete-Linkage	2	**0,99**	**0,38**	**1,00**
Average-Linkage (UPGMA)	2	**0,99**	**0,38**	**1,00**
Average-Linkage (WPGMA)	2	**0,99**	**0,38**	**1,00**
K-means	2	**0,99**	**0,38**	**1,00**

Com relação aos experimentos realizados no conjunto *Figura 8.1(b)*, cujos agrupamentos produzidos pelos algoritmos aglomerativos são mostrados na Figura 8.3, foi observado que as estratégias de agrupamento SL, CL, UPGMA e WPGMA identificaram, corretamente, os dois grupos. O agrupamento produzido pelo algoritmo K-Means tem configuração idêntica às dos agrupamentos mostrados na Figura 8.3.

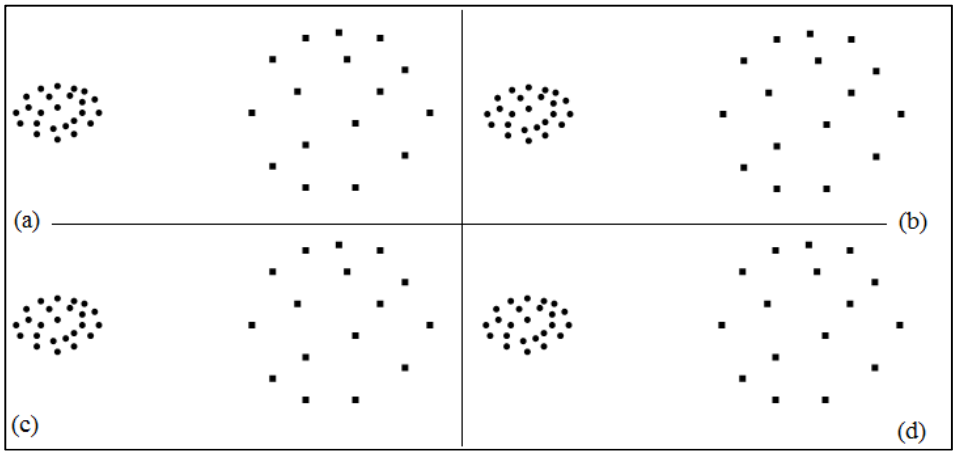


Figura 8.3 Plotagem dos agrupamentos produzidos pelos algoritmos aglomerativos no conjunto de padrões *Figura 8.1(b)*. (a) Agrupamento produzido pela estratégia *Single-Linkage*. (b) Agrupamento produzido pela estratégia *Complete-Linkage*. (c) Agrupamento produzido pela estratégia *Average-Linkage* (UPGMA). (d) Agrupamento produzido pela estratégia *Average-Linkage* (WPGMA).

Como mostram os valores dos índices na Tabela 8.3, todos os algoritmos de agrupamento utilizados no experimento mostraram o mesmo desempenho, em que D = 0,86, DB = 0,31 e R = 1,00 e identificaram corretamente os dois grupos do conjunto.

Tabela 8.3 Valores dos índices D, DB e R nos agrupamentos obtidos pelo SL, CL, UPGMA, WPGMA e K-means, no conjunto de padrões *Figura 8.1(b)*.

Estratégia de Agrupamento	# grupos	D	DB	R
Single-Linkage	2	**0,86**	**0,31**	**1,00**
Complete-Linkage	2	**0,86**	**0,31**	**1,00**
Average-Linkage (UPGMA)	2	**0,86**	**0,31**	**1,00**
Average-Linkage (WPGMA)	2	**0,86**	**0,31**	**1,00**
K-means	2	**0,86**	**0,31**	**1,00**

Os algoritmos aglomerativos utilizados nos experimentos realizados no conjunto *Figura 8.1(c)*, que produziram os agrupamentos mostrados na Figura 8.4, tiveram desempenhos equivalentes aos obtidos dos experimentos realizados com os conjuntos *Figura 8.1(a)* e *Figura 8.1(b)*.

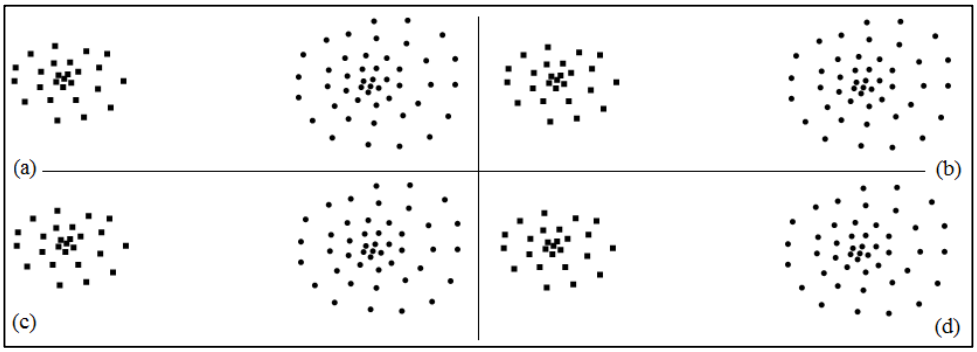


Figura 8.4 Plotagem dos agrupamentos produzidos pelos algoritmos aglomerativos no conjunto de padrões *Figura 8.1(c)*. (a) Agrupamento produzido pela estratégia *Single-Linkage*. (b) Agrupamento produzido pela estratégia *Complete-Linkage*. (c) Agrupamento produzido pela estratégia *Average-Linkage* (UPGMA). (d) Agrupamento produzido pela estratégia *Average-Linkage* (WPGMA).

Observe que, mesmo com a diferença de volume e densidade entre os dois grupos, as estratégias SL, CL, UPGMA e WPGMA tiveram o mesmo desempenho, em que D = 0,99, DB = 0,28 e R = 1,00 como mostra a Tabela 8.4. O agrupamento produzido pelo algoritmo K-Means tem configuração idêntica às dos agrupamentos mostrados na Figura 8.4.

Com base nos resultados obtidos dos experimentos nos conjuntos *Figura 8.1(a)*, *Figura 8.1(b)* e *Figura 8.1(c)*, pode-se conjecturar que os algoritmos aglomerativos apresentam bom desempenho devido aos conjuntos de padrões serem formados por grupos compactos e bem separados.

Tabela 8.4 Valores dos índices D, DB e R nos agrupamentos obtidos pelo SL, CL, UPGMA, WPGMA e K-means, no conjunto de padrões *Figura 8.1(c)*.

Estratégia de Agrupamento	# grupos	D	DB	R
Single-Linkage	2	**0,99**	**0,28**	**1,00**
Complete-Linkage	2	**0,99**	**0,28**	**1,00**
Average-Linkage (UPGMA)	2	**0,99**	**0,28**	**1,00**
Average-Linkage (WPGMA)	2	**0,99**	**0,28**	**1,00**
K-means	2	**0,99**	**0,28**	**1,00**

No experimento realizado com o conjunto de padrões *Figura 8.1(d)*, apenas a estratégia de agrupamento SL identificou corretamente os dois grupos, como mostra a Figura 8.5. Como comentado no Capítulo 3, a estratégia SL mostra bom desempenho na tarefa de detectar grupos alongados, como é o caso dos grupos do conjunto *Figura 8.1(d)*. Para melhorar a visualização dos grupos identificados pelos algoritmos

aglomerativos mostrados na Figura 8.5, tais grupos foram contornados por uma linha tracejada.

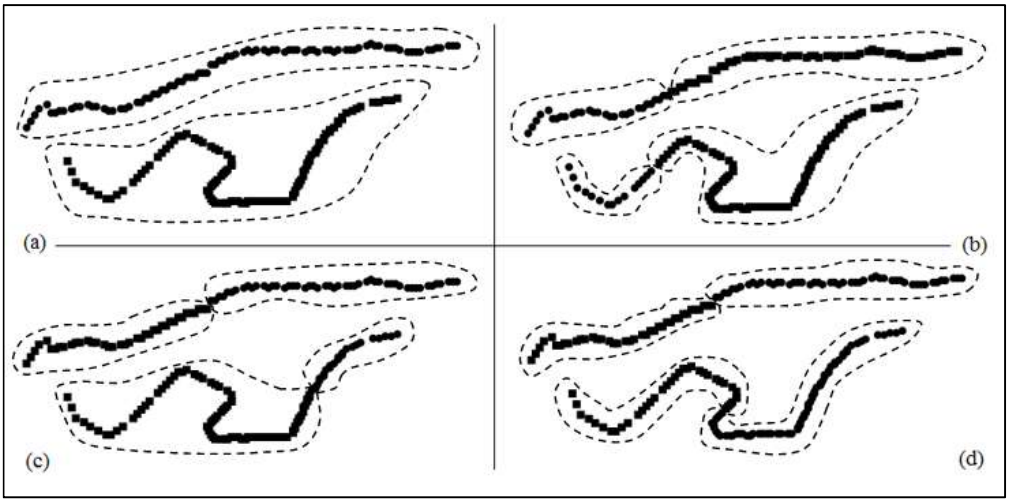


Figura 8.5 Plotagem dos agrupamentos produzidos pelos algoritmos aglomerativos no conjunto de padrões *Figura 8.1(d)* cujos grupos estão contornados por uma linha tracejada. (a) Agrupamento produzido pela estratégia *Single-Linkage*. (b) Agrupamento produzido pela estratégia *Complete-Linkage*. (c) Agrupamento produzido pela estratégia *Average-Linkage* (UPGMA). (d) Agrupamento produzido pela estratégia *Average-Linkage* (WPGMA).

Tanto os valores dos índices de validação mostrados na Tabela 8.5, quanto a inspeção visual dos agrupamentos mostrados na Figura 8.5 indicam que a estratégia de agrupamento SL obteve o melhor desempenho, com valores de D = 0,10 e R = 1,00; entretanto, o valor de DB = 1,95 sugere que o agrupamento produzido por SL foi o que apresentou o pior resultado. Por outro lado, o valor de DB sugere que o melhor desempenho foi obtido pela estratégia de agrupamento CL cujo valor de DB = 0,76. O agrupamento produzido pelo algoritmo K-Means tem configuração semelhante à do agrupamento mostrados na Figura 8.5 *(c)*.

Como brevemente discutido no Capítulo 4, os índices de validação interno (D e DB) consideram o grau de compactação dos grupos, o que pode não ser recomendado para avaliar um agrupamento formado por grupos alongados, que é o caso dos grupos do conjunto *Figura 8.1(d)*. O índice DB apresenta ainda uma outra limitação, pois a distância entre grupos é calculada com base na distância entre seus centroides o que, no caso de grupos alongados, pode levar a resultados imprecisos.

Tabela 8.5 Valores dos índices D, DB e R nos agrupamentos obtidos pelo SL, CL, UPGMA, WPGMA e K-means, no conjunto de padrões *Figura 8.1(d)*.

Estratégia de Agrupamento	# grupos	D	DB	R
Single-Linkage	2	**0,10**	1,95	**1,00**
Complete-Linkage	2	0,02	**0,76**	0,53
Average-Linkage (UPGMA)	2	0,02	0,87	0,56
Average-Linkage (WPGMA)	2	0,02	0,89	0,51
K-means	2	0,02	0,84	0,50

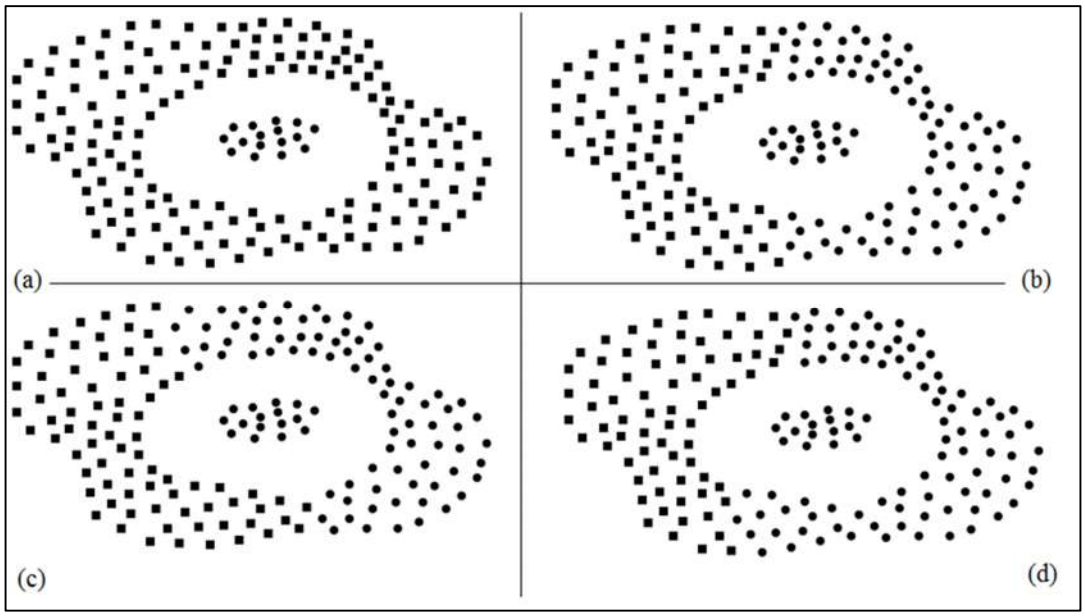


Figura 8.6 Plotagem dos agrupamentos produzidos pelos algoritmos aglomerativos no conjunto de padrões *Figura 8.1(e)*. (a) Agrupamento produzido pela estratégia *Single-Linkage*. (b) Agrupamento produzido pela estratégia *Complete-Linkage*. (c) Agrupamento produzido pela estratégia *Average-Linkage* (UPGMA). (d) Agrupamento produzido pela estratégia *Average-Linkage* (WPGMA).

Com relação aos experimentos realizados com o conjunto *Figura 8.1(e)*, apenas a estratégia de agrupamento SL identificou corretamente os dois grupos, cujos valores dos índices D = 0,13 e R = 1,00 estão mostrados na Tabela 8.6. O agrupamento produzido pelo algoritmo K-Means tem configuração idêntica à do agrupamentos mostrado na Figura 8.6 *(b)*.

Tabela 8.6 Valores dos índices D, DB e R nos agrupamentos obtidos pelo SL, CL, UPGMA, WPGMA e K-means, no conjunto de padrões *Figura 8.1(e)*.

Estratégia de Agrupamento	# grupos	D	DB	R
Single-Linkage	2	**0,13**	7,88	**1,00**
Complete-Linkage	2	0,07	0,92	0,50
Average-Linkage (UPGMA)	2	0,06	0,99	0,50
Average-Linkage (WPGMA)	2	0,07	**0,91**	0,50
K-means	2	0,06	0,92	0,50

Na sequência, os algoritmos aglomerativos têm seu desempenho avaliado no problema que envolve a identificação das três espirais, problema este conhecido em tarefas de classificação. A Figura 8.7 mostra os agrupamentos produzidos pelos algoritmos aglomerativos no conjunto *Figura 8.1(f)*.

Como pode ser percebido nos agrupamentos mostrados na Figura 8.7, apenas a estratégia de agrupamento SL detectou corretamente os três grupos (Figura 8.7 (a)) cujo valor de R = 1,00; entretanto, os valores dos índices DB sugerem que as estratégias CL e UPGMA apresentaram melhor desempenho, o que pode ser evidenciado na Tabela 8.7.

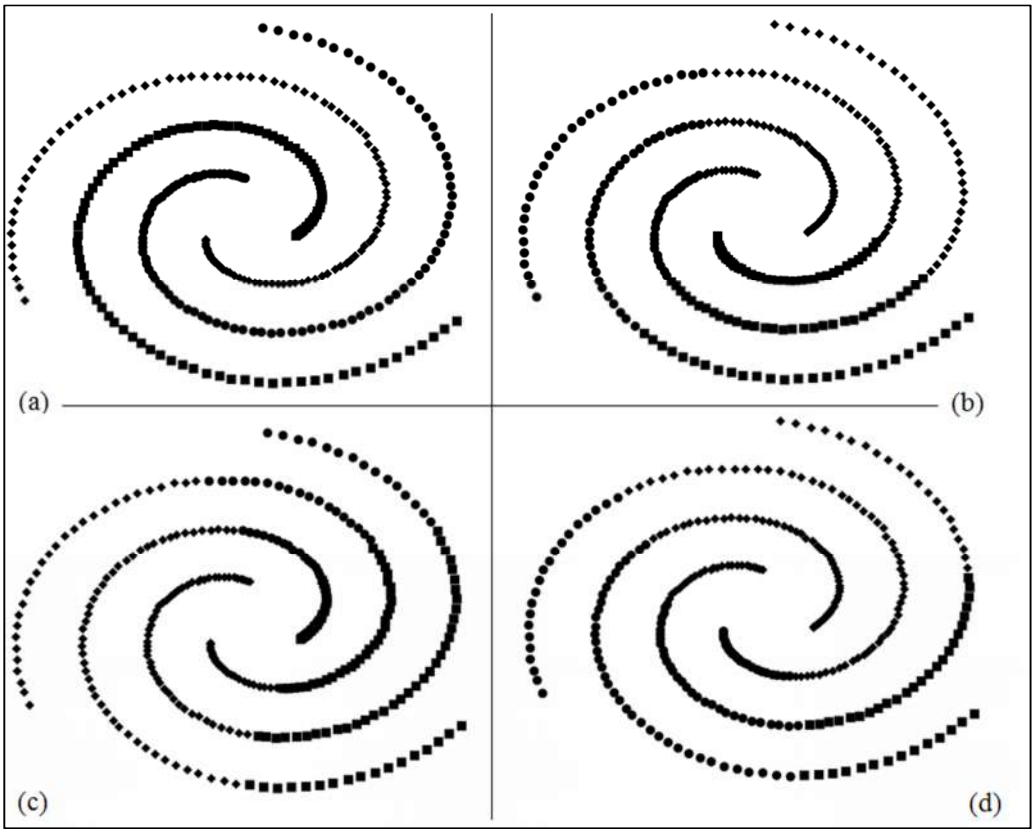


Figura 8.7 Plotagem dos agrupamentos produzidos pelos algoritmos aglomerativos no conjunto de padrões *Figura 8.1(f)*. (a) Agrupamento produzido pela estratégia *Single-Linkage*. (b) Agrupamento produzido pela estratégia *Complete-Linkage*. (c) Agrupamento produzido pela estratégia *Average-Linkage* (UPGMA). (d) Agrupamento produzido pela estratégia *Average-Linkage* (WPGMA).

Tabela 8.7 Valores dos índices D, DB e R nos agrupamentos obtidos pelo SL, CL, UPGMA, WPGMA e K-means, no conjunto de padrões *Figura 8.1(f)*.

Estratégia de Agrupamento	# grupos	D	DB	R
Single-Linkage	3	**0,14**	3,90	**1,00**
Complete-Linkage	3	0,02	0,59	0,56
Average-Linkage (UPGMA)	3	0,02	0,59	0,54
Average-Linkage (WPGMA)	3	0,02	0,73	0,55
K-means	3	0,01	**0,58**	0,55

Observe que, assim como os valores obtidos dos experimentos com o conjunto *Figura 8.1(d)*, os valores dos índices D, DB e R, mostrados na Tabela 8.7, são divergentes e sugerem avaliações diferentes dos resultados obtidos.

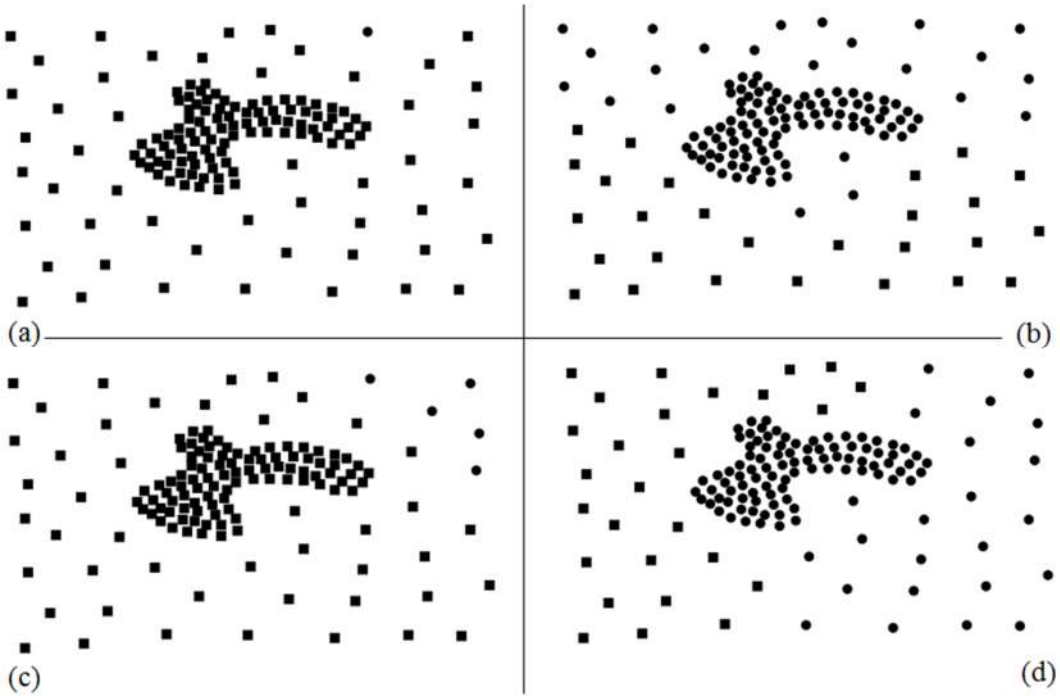


Figura 8.8 Plotagem dos agrupamentos produzidos pelos algoritmos aglomerativos no conjunto de padrões *Figura 8.1(g)*. (a) Agrupamento produzido pela estratégia *Single-Linkage*. (b) Agrupamento produzido pela estratégia *Complete-Linkage*. (c) Agrupamento produzido pela estratégia *Average-Linkage* (UPGMA). (d) Agrupamento produzido pela estratégia *Average-Linkage* (WPGMA).

A Figura 8.8 mostra os agrupamentos produzidos pelas estratégias de agrupamento aglomerativo no conjunto *Figura 8.1(g)*. Note que, tanto por meio da inspeção visual quanto com base nos valores dos índices mostrados na Tabela 8.8, nenhuma estratégia de agrupamento identificou corretamente os dois grupos. Entretanto o melhor agrupamento foi obtido pela estratégia WPGMA (Figura 8.8(d)) cujo valor de R = 0,73. Veja que os valores dos índice D e B mostrados na Tabela 8.8

sugerem que a estratégia WPGMA obteve o pior desempenho entre as quatro estratégias aglomerativas. O agrupamento produzido pelo algoritmo K-Means tem configuração semelhante à do agrupamento mostrado na Figura 8.8 *(b)*.

Tabela 8.8 Valores dos índices D, DB e R nos agrupamentos obtidos pelo SL, CL, UPGMA, WPGMA e K-means, no conjunto de padrões *Figura 8.1(g)*.

Estratégia de Agrupamento	# grupos	D	DB	R
Single-Linkage	2	**0,12**	**0,52**	0,54
Complete-Linkage	2	0,11	1,20	0,68
Average-Linkage (UPGMA)	2	0,11	0,70	0,56
Average-Linkage (WPGMA)	2	0,08	1,69	**0,73**
K-means	2	0,04	1,21	0,68

A seguir são mostrados, na Figura 8.9, os quatro agrupamentos produzidos pelos algoritmos aglomerativos no conjunto *Figura 8.1(h)*.

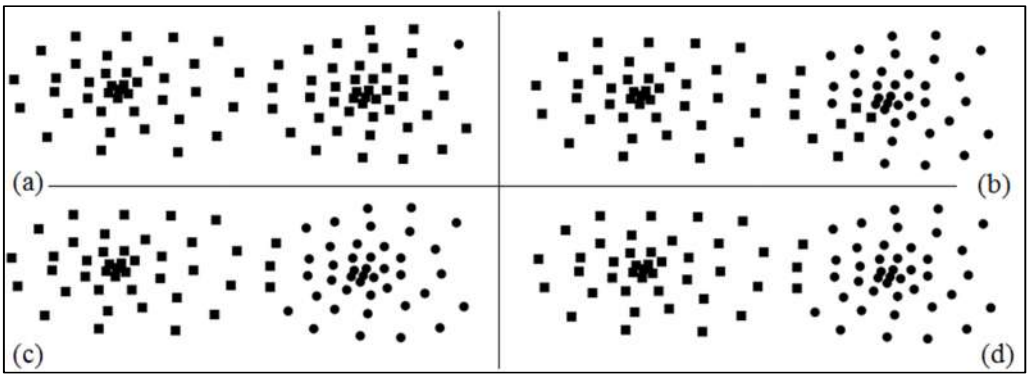


Figura 8.9 Plotagem dos agrupamentos produzidos pelos algoritmos aglomerativos no conjunto de padrões *Figura 8.1(h)*. (a) Agrupamento produzido pela estratégia *Single-Linkage*. (b) Agrupamento produzido pela estratégia *Complete-Linkage*. (c) Agrupamento produzido pela estratégia *Average-Linkage* (UPGMA). (d) Agrupamento produzido pela estratégia *Average-Linkage* (WPGMA).

Como mostra a Tabela 8.9, o K-means foi o único algoritmo que identificou corretamente os dois grupos do conjunto *Figura 8.1(h)*. Entre as estratégias de agrupamento aglomerativo, apenas a UPGMA e WPGMA obtiveram melhores desempenhos, cujos valores dos índices R = 0,93.

Tabela 8.9 Valores dos índices D, DB e R nos agrupamentos obtidos pelo SL, CL, UPGMA, WPGMA e K-means, no conjunto de padrões *Figura 8.1(h)*.

Estratégia de Agrupamento	# grupos	D	DB	R
Single-Linkage	2	0,11	0,59	0,49
Complete-Linkage	2	0,06	0,68	0,80
Average-Linkage (UPGMA)	2	0,15	0,57	0,93
Average-Linkage (WPGMA)	2	0,15	0,57	0,93
K-means	2	**0,17**	**0,54**	**1,00**

8.4 Considerações Finais sobre os Resultados Descritos neste Capítulo

Os experimentos descritos neste capítulo tem como objetivo avaliar os algoritmos aglomerativos (SL, CL, UPGMA e WPGMA) em tarefas de agrupamento que envolvem identificar grupos visivelmente bem separados, nomeados *gestalt*. Exceto para os valores informados nas três primeiras tabelas, em que os resultados foram os mesmos para os cinco algoritmos, os números mostrados da Tabela 8.5 à Tabela 8.9 sugere que, na maioria dos casos, os melhores resultados foram obtidos com a abordagem aglomerativa. A estratégia de agrupamento *Single-Linkage* pode ser considerada a vencedora em três dos cinco experimentos.

Quando considerando somente os resultados dos experimentos com os conjuntos *Figura 8.1 (d)*, *Figura 8.1 (e)* e *Figura 8.1 (f)*, a estratégia de agrupamento *Single-Linkage* apresentou sua melhor performance, como confirmado pelos valores do índice R.

Capítulo 9

Conclusões e Trabalhos Futuros

Este trabalho de pesquisa investigou os algoritmos de agrupamento aglomerativos baseados em conceitos da Teoria de Matrizes e identificou algumas das principais características em conjuntos de padrões que promovem um bom desempenho de tais algoritmos. O esquema *Matrix Updating Algorithmic Scheme* (MUAS) [Theodoridis & Koutroumbas, 2009], parametrizável para quatro estratégias de agrupamento (*Single Linkage*, *Complete Linkage*, UPGMA e WPGMA), bem como o algoritmo AGNES, adaptado para induzir um número k de grupos, foram implementados com vistas a disponibilizar um ambiente de experimentação e viabilizar a investigação empírica de tais algoritmos.

Na sequência, a Seção 9.1 resume os principais pontos levantados e investigados na pesquisa realizada, bem como as principais contribuições desta dissertação. Ainda na Seção 9.1 são discutidas as conclusões obtidas dos experimentos, evidenciando o desempenho e as limitações dos algoritmos aglomerativos de agrupamento investigados neste trabalho. A Seção 9.2 encerra este capítulo com sugestões para continuidade do trabalho.

9.1 Resumo dos Principais Pontos Investigados e Contribuições desta Pesquisa

Como comentado no Capítulo 2, a versão clássica do algoritmo AGNES [Kaufman & Rousseeuw 2005] implementa apenas a estratégia *Average Linkage* UPGMA (*Unweighted Pair Group Method with Arithmetic Mean*); entretanto, a versão do AGNES implementada e utilizada nos experimentos desta pesquisa contempla as quatro estratégias de agrupamento mencionadas anteriormente. O trabalho contemplou, também, a implementação de índices de validação (interna e externa) de agrupamento, com vistas à sua utilização na comparação de resultados dos experimentos realizados nesta pesquisa.

As quatro estratégias de agrupamento consideradas neste trabalho produzem, dependendo das características do conjunto de padrões, agrupamentos distintos no

mesmo conjunto de padrões. No caso de grupos com formato alongado ou não elipsoidal, caso dos grupos dos conjuntos *Figura 8.1 (d)*, *Figura 8.1 (e)* e *Figura 8.1 (f)* utilizados nos experimentos descritos no Capítulo 8, a estratégia de agrupamento *Single Linkage* apresenta bom desempenho, confirmando os resultados publicados em [Blashfield 1976], [Hubert 1974] e [Milligan 1981]. Entretanto, a estratégia *Single Linkage* tende a unir grupos (*efeito de encadeamento*) em conjuntos com presença de *outliers* e/ou 'pontes' (*bridges*) entre tais grupos, caso de alguns dos grupos dos conjuntos utilizados nos experimentos descritos nos Capítulo 7 e Capítulo 6, respectivamente, confirmando os resultados publicados em [Milligan 1981] e [Hands & Everitt 1987].

Por outro lado, a estratégia de agrupamento *Complete Linkage* é menos sensível a *outliers* e tende a produzir grupos mais compactos (i.e., de menor diâmetro) e, consequentemente, tende a 'quebrar' grupos 'grandes' (caso do grupo 4 do conjunto *Aggregation*, cujo diâmetro é maior do que os diâmetros dos grupos 5 e 7) e produzir grupos com formato elipsoidal, confirmando os resultados publicados em [Cunningham & Ogilvie 1972] e [Hubert 1974]. Sempre que a estratégia *Complete Linkage* faz a junção de dois grupos, A e B por exemplo, para formar um novo grupo, C, a dissimilaridade entre A e B corresponde ao diâmetro do grupo C [Kaufman & Rousseeuw 2005].

Os resultados obtidos dos experimentos realizados neste trabalho sugerem que as estratégias de agrupamento *Single Linkage* e *Complete Linkage* mostram algumas limitações em relação à forma geométrica, diâmetro, densidade, compactação, distância inter-grupo e etc. Algumas destas limitações, discutidas em [Kaufman & Rousseeuw 2005], foram consideradas pelos autores do AGNES na decisão de incorporar apenas a estratégia de agrupamento UPGMA em seu algoritmo.

Como apontado em [Kaufman & Rousseeuw 2005] e com base nos resultados obtidos dos experimentos, a estratégia de agrupamento UPGMA foi avaliada como sendo a mais robusta (exceto para grupos com formas alongadas ou lineares) entre as quatro estratégias de agrupamentos investigadas nesta pesquisa. Entretanto, algumas características do conjunto de padrões podem melhorar ou piorar o desempenho da estratégia UPGMA, tais como a distância inter-grupo, a quantidade de grupos, a esparsidade dos grupos, presença de *outliers* etc. A estratégia de agrupamento WPGMA, que é uma variante da estratégia *Average Linkage*, mostrou bom desempenho em apenas um dos oitos experimentos descritos no Capítulo 8, o

experimento que envolve identificar grupos em regiões de densidades diferentes, caso dos dois grupos do conjunto *Figura 8.1 (g)*.

Algoritmos de agrupamento diferentes têm, geralmente, desempenhos diferentes. O algoritmo K-Means, cujos resultados serviram de *baseline* na comparação com o algoritmo AGNES, é sensível à grupos que não têm o mesmo diâmetro (caso dos grupos do conjunto *Aggregation*) e os agrupamentos por ele produzidos dependem da escolha inicial dos centroides. Entretanto, o K-Means, com base nos resultados obtidos dos experimentos do Capítulo 7, mostrou-se menos sensível à presença de *outliers*.

Com relação aos índices de validação Dunn (D) e Davies-Bouldin (DB), utilizados para avaliar a qualidade dos agrupamentos obtidos dos algoritmos aglomerativos, observou-se que, em alguns casos, não há correlação entre seus valores, como mostrado nas Tabelas 8.5, 8.6 e 8.7. No caso dos experimentos realizados com o conjunto *Figura 8.1 (g),* os valores de D e DB são associados (como mostram os valores informados na Tabela 8.8), mas indicam, equivocadamente, qual o melhor agrupamento. Dessa forma este trabalho contribui, também, para mostrar que diferentes índices de validação fornecem diferentes 'visões' em relação à qualidade de um mesmo agrupamento.

Obviamente que, diante de inúmeras características presentes nos diversos conjuntos de padrões, não se pode afirmar que apenas uma estratégia de agrupamento é a melhor. Como exemplo, os algoritmos propostos por [Kang & Landry 2015] e [Ghrab *et al*., 2016] fazem uso da estratégia *Single-Linkage* em tarefas que envolvem analisar sequências de movimentos oculares que ocorrem durante o rastreamento visual de múltiplos alvos em movimento, e análises de eventos em vídeos de câmeras de monitoramento, respectivamente. Já em [Anantharajah *et al*., 2015] a estratégia *Complete Linkage* foi escolhida para incorporar um algoritmo para agrupamento de imagens faciais. No *framework* proposto por [Xu *et al*., 2016] com o objetivo de identificar características relacionadas a confiabilidade de *software*, os autores optaram por empregar a estratégia *Average Linkage* UPGMA, como parte da solução. Dito isto, pode-se sugerir que cada estratégia de agrupamento aglomerativa, ou a combinação delas, apresenta melhor ou pior desempenho dependendo das características do conjunto de padrões ou domínio de dados.

9.2 Sugestões para Continuidade e Trabalhos Futuros

Como comentado na Seção 9.1, diferentes estratégias de agrupamento podem produzir diferentes agrupamentos no mesmo conjunto de padrões. Diante deste fato, este trabalho sugere a investigação de técnicas, empregadas por algoritmos genéticos, com o objetivo de descobrir o agrupamento 'mais natural' em um conjunto de padrões. Algoritmos Genéticos são algoritmos de busca e otimização fundamentados nos princípios da seleção natural [Goldberg 1989]. Tais algoritmos aplicam certos operadores à populações de soluções ao problema em questão, de forma que as novas populações são mais evoluídas em comparação com as anteriores de acordo com uma função critério. Este procedimento é aplicado para um número pré-selecionado de iterações e a saída do algoritmo é a melhor solução encontrada na última população ou, em alguns casos, a melhor solução é encontrada durante a evolução do algoritmo [Theodorids & Koutroumbas 2009].

Na pesquisa realizada por [Raghavan & Birchad 1979] foi proposta uma estratégia de agrupamento baseada em formalismo do processo reprodutivo em sistemas naturais. A pesquisa realizada por [Krovi 1992] investigou a viabilidade de utilizar algoritmos genéticos em problemas de agrupamento e propõe uma técnica baseada nos princípios da seleção natural. Em [Lipczak & Milios 2009] foi proposta uma extensão do algoritmo aglomerativo na forma de um algoritmo genético para problemas de agrupamento envolvendo redes sociais.

Referências

Abu-Mostafa, S. Y., Magdon-Ismail, M., & Lin, H. (2012) *Learning From Data.* USA: AMLBook.

Amorim, R. C. (2011) *Learning feature weights for K-Means clustering using Minkowski metric*, *Ph. D.* Birkbeck, University of London.

Anantharajah, K., Denman, S., Tjondronegoro, D., Sridharan, S. & Fookes, C. (2015) Robust Automatic Face Clustering in News Video. *In: Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, pp. 1-8.

Berry, M., & Linoff, G. (1996). *Data Mining Techniques For Marketing, Sales and Customer Support.* USA: John Wiley & Sons, Inc.

Berkhin, P. (2002). Survey of clustering data mining techniques. *Relatório técnico*, Accruel Software, San Jose, CA. http://citeseer.nj.nec.com/berkhin02survey.html.

Berthold, M. R., Borgelt, C., Höppner, F. and Klawonn, F. (2010) Guide to Intelligent Data Analysis, Texts in Computer Science, D. Gries & F. B. Schneider (Eds.), *Springer-Verlag*, pp. 115-143.

Bishop, C. M. (2006) *Pattern recognition and machine learning.* USA: Springer Verlag.

Blashfield, R. K. (1976) Mixture model tests of cluster analysis. Accuracy of four agglomerative hierarchical methods. *Psychological Bulletin*, pp. 377–385.

Chapelle, O., Scholkopf, B., Zien, A. (2006) *Semi-Supervised Learning*. Cambridge, MA: MIT Press.

Cunningham, K. M. and Ogilvie, L. C. (1972) Evaluation of hierarchical grouping techniques: a preliminary study. *Computer Journal*, pp. 209–213.

Davies, D. L., Bouldin, D. W. (1979) A cluster separation measure. *IEEE Trans. Pattern Anal. Machine Intelligence*, 1(4), pp. 224-227.

Duda, R. O., Hart, P. F., & Stork, D. G. (2001) *Pattern Classification.* USA: John Wiley & Sons, Inc.

Dunn J. C. (1973) A fuzzy relative of the isodata process and its use in detecting compact well separated clusters, *In: Journal of Cybernetics*, v. 3, pp. 32-57.

Eriksson, B., Dasarathy, G., Singh, A., & Nowak, R. (2001) Active clustering: robust and efficient hierarchical clustering using adaptively selected similarities. *In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), 15*, pp. 260-268.

Everitt, S. B., Landau, S., Leese, M., & Stahl, D. (2011) *Cluster Analysis* (5th ed.). United Kingdom: John Wiley & Sons, Ltd.

Gan, G., Ma, C., & Wu, J. (2007) *Data Clustering: Theory, Algorithms and Applications.* USA: ASA-SIAM.

Ghrab, B., N., Fendri, E., & Hammami, M. (2016) Abnormal events detection based on trajectory clustering. *In: 13th International Conference Computer Graphics, Image and Visualization*. IEEE.

Gionis, A., Mannila, H., & Tsaparas. P. (2005) Clustering Aggregation. *In: Proceedings of the 21st International Conference on Data Engineering (ICDE 2005)*.

Goldberg, D. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. USA: Addison Wesley.

Gower J.C. (1967) A comparison of some methods of cluster analysis. *Biometrics*,v. 23, pp. 623–628.

Gower, J. C. (1990) Clustering axioms. *Classification Society of North America Newsletter*, July, pp. 2–3.

Guha, S. R. (1998) CURE: an efficient clustering algorithm for large databases. *ACM Sigmod International Conference on Management of Data (SIGMOD)*, pp. 73–84.

Guha, S. R. (2000) ROCK: A robust clustering algorithm for categorical attributes. *Information Systems 25*, pp. 345–366.

Halkidi, M., Vazirgiannis, M. (2001) Cluster Validity Assessment: Finding the optimal partitioning of a data set, *IEEE International Conference on Data Mining (ICDM'01)*, pp. 187–194.

Han, J., Kamber, M., & Pei, J. (2012) *Data Mining Concepts and Techniques.* USA: Elsevier Inc.

Handl, J. & Knowles, J. (2004) Multiobjective clustering with automatic determination of the number of clusters. Technical Report TR-COMPS/SBIO 2004-02. UMIST, Manchester.

Hands, S. & Everitt, B. S. (1987) A Monte Carlo study of the recovery of cluster structure in binary data by hierarchical clustering techniques. *Multivariate Behavioral Research*, pp. 235–243.

Hartigan, J. A. (1967) Representation of similarity matrices by trees. *Journal of the American Statistical Association*, pp. 1140–1158.

Hartigan, J. A. (1975) *Clustering algorithms.* USA: John Wiley & Sons, Inc.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statiscal Learning: Data Mining, Inference, and Prediction* (2th ed.). New York, NY, USA: Springer.

Hubert, L. (1974) Approximate evaluation techniques for the single-link and complete-link hierarchical clustering procedures. *Journal of the American Statistical Association*, pp. 698–704.

Jaccard, P. (1908) Nouvelles recherches sur la distribution florale. *Bulletin de la Société Vaudoise de Sciences Naturelles*. pp, 223–370.

Jain, A. (2010) Data clustering: 50 years beyond K-Means. *Pattern Recognition Letters, 31*, pp. 651–666.

Jain, A. K. (1999) Data clustering: a review. *ACM Computing Surveys 31*, pp. 264–323.

Jain, A., & Dubes, R. (1988) *Algorithms for Clustering Data.* Prentice Hall.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statiscal Learning with Applications in R.* USA: Springer.

Jardine, C. J., Jardine, N., & Sibson, R. (1967) The structure and construction of taxonomic hierarchies. *Mathematical Biosciences*, pp. 173–179.

Johnson, S. C. (1967) Hierarchical clustering schemes. *Psychometrika*, pp. 241–254.

Kang Z. & Landry. S. J. (2015) An Eye Movement Analysis Algorithm for a Multielement Target Tracking Task: Maximum Transition-Based Agglomerative Hierarchical Clustering. *In: Transactions on Human-Machine Systems*. IEEE, pp. 13-24.

Karypis, G. E.-H. (1999) CHAMELEON: Hierarchical clustering using dynamic modeling. *Computer 32*, pp. 68–75.

Kaufman, L. & Rousseeuw, P. (2005) *Finding Groups in Data: An Introduction to Cluster Analysis.* New Jersey, USA: John Wiley & Sons, Inc.

Krishnamurthy, A., Balakrishnan, S., Xu, M., & Singh, A. (2012). Efficient active algorithms for hierarchical clustering. *In: Proceedings of the 29th International Conference on Machine Learning*, Edinburg, Scotland, UK.

Krovi, R. (1992) Genetic algorithms for clustering: a preliminary investigation. *In: Proceedings of the Twenty-Fifth Hawaii International Conference*, pp. 540-544 v.4

Kurita, T. (1991) An efficient agglomerative clustering algorithm using a heap. *Pattern Recognition, 24*, pp. 777-783.

Lance, G. N., & Williams, W. T. (1967) A general theory of classificatory sorting strategies. *Computer Journal*, pp. 373–380.

Langley, P. (2011) The changing science of machine learning. *Machine Learning*, pp. 275–279.

Li, X. (1990) Parallel algorithms for hierarchical clustering and cluster validity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1088–1092.

Lipczak, M. & Milios, E. (2009) Agglomerative genetic algorithm for clustering in social networks. *ACM Proceeding of the 11th Annual conference on Genetic and evolutionary computation*, pp. 1243-1250.

Michalski, R.S., Bratko, I., & Kubat, M. (1998) *Machine Learning and Data Mining: Methods and Applications*. John Wiley & Sons. West Sussex, England.

Milligan, G. W. (1981) A review of Monte Carlo tests of cluster analysis. *Multivariate Behavioral Research*, pp. 379–407.

Milligan & Cooper (1988) *A study of standardization of variables in cluster analysis*. *Journal of Classification*, v.5, Issue 2, pp. 181-204.

Mitchell, T. M. (1997) *Machine Learning.* USA: McGraw-Hill.

Müller, D. (s.d.). Modern hierarchical, agglomerative clustering algorithms. *arXic:1109.2378*.

Murtagh, F. (1983) A survey of recent advances in hierarchical clustering algorithms. *Journal of Computation, 1*, pp. 354–359.

Murtagh, F. (1984) Complexities of hierarchic clustering algorithms: State of the art. *Computational Statistics Quarterly, 1*, pp. 101–113.

Murtagh, F. (1985) COMPSTAT Lectures 4. Em *Multidimensional Clustering Algorithms.* Vienna.

Nagi, G. (1968) State of the art in pattern recognition. *In: Proceedings of the IEEE,* vol 56, pp. 836-862.

Nicoletti, M. C. (1994) *Ampliando os limites do aprendizado indutivo de máquina através das abordagens construtiva e relacional, Ph. D.* IFSC-USP.

Nicoletti, M., Magalhães, J., & Nicoletti, M. (1998) *O uso do sistema CN2 na indução de conhecimento em domínio farmacotécnico.* Relatório Técnico DC 005/98 UFSCar/DC, São Carlos.

Olson, C. F. (1993) Parallel algorithms for hierarchical clustering. *Parallel Computing* , pp. 1313–1325.

Quinlan, J. R. (1993) *C4.5: Programs for Machine Learning.* USA: Morgan Kaufmann Publishers.

Raghavan, V. & Birchard K. (1979) A clustering strategy based on a formalism of the reproductive process in natural systems. *ACM Proceedings of the 2nd annual international ACM SIGIR conference on Information storage and retrieval: information implications into the eighties*, pp. 10-22.

Tamura, Y., Miyamoto S. (2014) A Method of Two Stage Clustering Using Agglomerative Hierarchical Algorithms with One-Pass k-Means++ or k-Median++. *In: International Conference on Granular Computing*. IEEE, pp. 281-285.

Tan, P., Steinback, M., & Kumar, V. (2005) *Introduction to Data Mining.* USA: Pearson.

Theodoridis, S., & Koutroumbas, K. (2009) *Pattern Recognition* (4th ed.). USA: Academic Press.

Ward, J. H. (1963) Hierarchical grouping to optimize an objective funtion. *Journal of American Statistics Association*, pp. 236–244.

Webb, A. R., & Copsey, K. D. (2011) *Statistical Pattern Recognition* (3th ed.). USA: John Wiley & Sons, Inc.

Willett, P. (1989). Efficiency of hierarchic agglomerative clustering using the ICL distributed array processor. *Journal of Documentation, 45*, pp. 1-45.

Witten, H. I., Frank, E., & Hall, A. M. (2011) *Data Mining: Practical Machine Learning Tools and Techniques* (3th ed.). USA: Morgan Kaufmann.

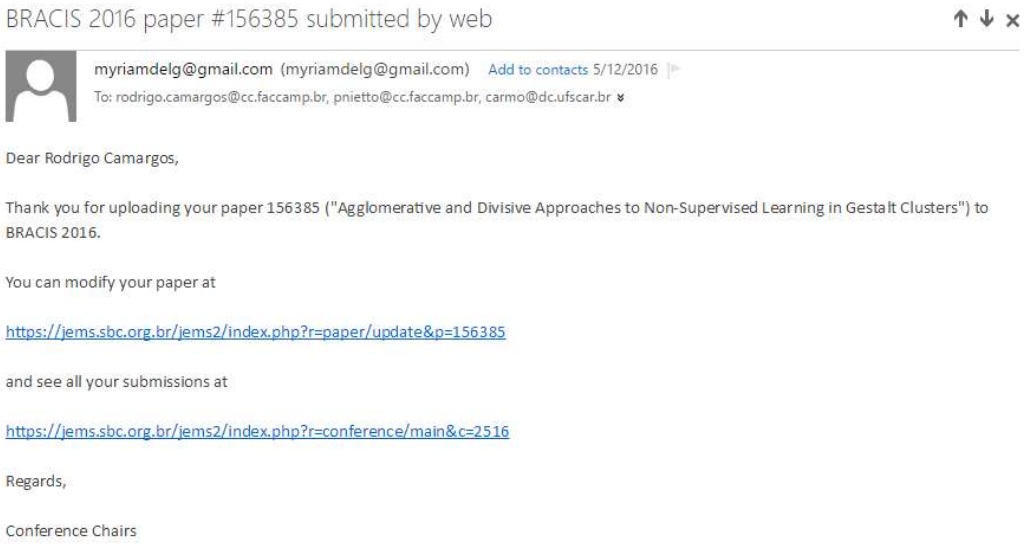
Zaki, J. M., & Meira Jr, W. (2014) *Data Mining and Analysis: Fundamental Concepts and Algorithms.* New York, NY, USA: Cambridge University Press.

Zhang, T. R. (1996) BIRCH: An Efficient Data Clustering Method for Very Large Databases. *Proceedings of the ACM SIGMOD Conference on Management of Data*, pp. 103–114.

Zhou, X., Jifeng, X., Jin, L., & Xiaohui, C. (2016) MICHAC: Defect Prediction via Feature Selection based on Maximal Information Coefficient with Hierarchical Agglomerative Clustering. *In: 23rd International Conference on Software Analysis, Evolution and Reengineering*. IEEE.

Anexo

Submissão do artigo *Agglomerative and Divisive Approaches to Non-Supervised Learning in Gestalt Clusters* para o *5th Brazilian Conference on Intelligent System* (BRACIS 2016).



BRACIS 2016 paper #156385 submitted by web

myriamdelg@gmail.com (myriamdelg@gmail.com) Add to contacts 5/12/2016

To: rodrigo.camargos@cc.faccamp.br, pnietto@cc.faccamp.br, carmo@dc.ufscar.br

Dear Rodrigo Camargos,

Thank you for uploading your paper 156385 ("Agglomerative and Divisive Approaches to Non-Supervised Learning in Gestalt Clusters") to BRACIS 2016.

You can modify your paper at

https://jems.sbc.org.br/jems2/index.php?r=paper/update&p=156385

and see all your submissions at

https://jems.sbc.org.br/jems2/index.php?r=conference/main&c=2516

Regards,

Conference Chairs