

UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA  
DEPARTAMENTO DE ESTATÍSTICA

## **Comparação e validação de diferentes métodos de agrupamentos**

**José Matheus Badaró**

**Trabalho de Conclusão de Curso**



UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA  
DEPARTAMENTO DE ESTATÍSTICA

## Comparação e validação de diferentes métodos de agrupamentos

**José Matheus Badaró**

**Orientador: Márcio Luis Lanfredi Viola**

Trabalho de Conclusão de Curso apresentado ao Departamento de Estatística da Universidade Federal de São Carlos - DEs-UFSCar, como parte dos requisitos para obtenção do título de Bacharel em Estatística.

**São Carlos**  
**Março de 2023**



José Matheus Badaró

## Comparação e validação de diferentes métodos de agrupamentos

Este exemplar corresponde à redação final do trabalho de conclusão de curso devidamente corrigido e defendido por José Matheus Badaró e aprovado pela banca examinadora.

Aprovado em 27 de março de 2023.

Banca Examinadora:

- Márcio Luis Lanfredi Viola (Orientador)
- Pedro Ferreira Filho
- Rafael Izbicki



# Resumo

Durante a realização de uma análise multivariada em um conjunto de dados, um dos principais desafios em estatística é resumir as informações do conjunto. Uma maneira de enfrentar tal desafio é a criação de agrupamentos cujos elementos possuam semelhanças e/ou pertençam a algum segmento. Embora existam diversos métodos de classificação, não há consenso sobre quais métodos são mais adequados para um determinado conjunto de dados. A comparação dos métodos de forma abrangente em muitos cenários possíveis é importante para analisar a qualidade da execução do método. Além das comparações, a validação dos resultados em uma análise de agrupamentos é uma maneira de verificar os resultados das análises com avaliações objetivas e quantitativas. Neste trabalho serão realizadas comparações entre os métodos de agrupamento *DBSCAN* (Ester *et al.* 1996), *Fuzzy C-Means* (Dunn 1973), *Gaussian Mixture Model* abordado por Fraley e Raftery (2000), *Spectral* (Donath e Hoffman 1973) e *K-Means* (Forgy 1965). Esses métodos de agrupamento foram escolhidos por serem adequados para grandes conjuntos de dados e por apresentarem abordagens distintas na forma como realizam a clusterização. Com o auxílio do pacote do R “MixSIM”, uma ferramenta capaz de realizar simulações e mensurar o grau de interação entre os componentes, serão realizadas comparações entre os algoritmos citados.

**Palavras-chave:** *agrupamentos, dbscan, fuzzy, k-means, mixsim, gaussian-mixture, spectral, dunn, rand, clusterização.*





# Lista de Figuras


2.1	Agrupamentos com formato ovalizado. . . . .	21
2.2	Agrupamentos com formatos arbitrários. . . . .	21
2.3	Vizinhança $V_\epsilon$ dos pontos $p$ e $q$ . . . . .	22
2.4	Alcance por densidade dos pontos $p_1, \dots, p_n$ . . . . .	23
2.5	Conexão por densidade dos pontos $p$ e $q$ pelo ponto $r$ . . . . .	23
2.6	Funções de densidade de distribuições normais univariadas dos dois componentes de uma mistura finita bidimensional. . . . .	30
2.7	Contornos da função de densidade de uma distribuição normal bivariada associada a um modelo de mistura finito bidimensional com dois componentes de mistura. . . . .	31
2.8	Conjuntos de dados simulados obtidos a partir de modelos de mistura gaussiana. . . . .	40
3.1	Índice de Rand Ajustado de cada método utilizado sobre diferentes níveis de sobreposição de pares. . . . .	47
3.2	Proporção de classificações corretas de cada método utilizado sobre diferentes níveis de sobreposição de pares. . . . .	48
3.3	Variação da informação de cada método utilizado sobre diferentes níveis de sobreposição de pares. . . . .	50
3.4	Coeficiente de Silhueta de cada método utilizado sobre diferentes níveis de sobreposição de pares. . . . .	51
3.5	Índice de <i>Dunn</i> de cada método utilizado sobre diferentes níveis de sobreposição de pares. . . . .	52
3.6	Tempo de execução em segundos de cada método utilizado sobre diferentes níveis de sobreposição de pares. . . . .	54

3.7	Tempo de execução em segundos de cada método utilizado, exceto pelo <i>Spectral</i> , sobre diferentes níveis de sobreposição de pares. . . . .	54
3.8	Índice de Rand de cada método utilizado sobre diferentes quantidades de agrupamentos. . . . .	55
3.9	Índice de <i>Dunn</i> para cada método utilizado sobre diferentes quantidades de agrupamentos. . . . .	57
3.10	Tempo de execução em segundos para cada método utilizado sobre diferentes quantidades de agrupamentos. . . . .	57
3.11	Índice de Rand Ajustado de cada método utilizado sobre diferentes quantidades de dimensões. . . . .	58
3.12	Variação da Informação de cada método utilizado sobre diferentes quantidades de dimensões. . . . .	59
3.13	Tempo de execução em segundos de cada método utilizado sobre diferentes quantidades de dimensões. . . . .	60
3.14	Quantidade máxima de observações em que cada método conseguiu processar durante as aplicações em dados simulados, considerando 3 agrupamentos e 5 dimensões, sem sobreposição de pares. . . . .	61
3.15	Tempo de execução em segundos de cada método utilizado sobre diferentes quantidades de observações. . . . .	62
3.16	Índice de Rand Ajustado de cada método utilizado sobre diferentes níveis de sobreposição de pares, considerando 50 dimensões e 3 agrupamentos, com 5mil observações. . . . .	63
3.17	Índice de Rand Ajustado de cada método utilizado sobre diferentes níveis de sobreposição de pares, considerando 5 dimensões e 30 agrupamentos, com 5mil observações. . . . .	64
3.18	Índice de Rand Ajustado de cada método utilizado sobre diferentes números de agrupamentos, considerando 50 dimensões e 5mil observações. . . . .	65
3.19	Índice de Rand Ajustado de cada método utilizado sobre diferentes números de dimensões, considerando 30 agrupamentos e 5mil observações. . . . .	65
3.20	Índice de Rand Ajustado de cada método utilizado sobre diferentes números de dimensões, considerando 3 agrupamentos, 5mil observações e $\bar{\omega} = 5\%$ . . . . .	66
3.21	Boxplot para o Índice de Rand Ajustado considerando todos os cenários simulados. . . . .	67

3.22	Boxplot para a Variação da Informação considerando todos os cenários simulados. . . . .	68
3.23	Boxplot para o Coeficiente de Silhueta considerando todos os cenários simulados. . . . .	69
3.24	Boxplot para o Índice de <i>Dunn</i> considerando todos os cenários simulados. .	69
3.25	Boxplot para o tempo de execução em segundos considerando todos os cenários simulados. . . . .	70
3.26	Comparação dos métodos em estudo aplicados respectivamente nos conjuntos de dados gerados por <i>noisy_circles</i> , <i>noisy_moons</i> , <i>varied</i> , <i>aniso</i> , <i>blobs</i> e <i>no_structure</i> do <i>Toy Datasets</i> . . . . .	73
3.27	Comparação dos métodos em estudo aplicados nos conjuntos de dados <i>Compound</i> , <i>Aggregation</i> , <i>PathBased</i> , <i>S2</i> , <i>Flame</i> e <i>Face</i> , obtidos a partir do repositório Ultsch e Pasewaldt (2005) e do elbamos (2021) (“Rótulos” na figura representa os rótulos reais dos agrupamentos). . . . .	75



# Lista de Tabelas

2.1	Resumo dos argumentos e valores disponíveis retornados pela função <i>Mix-Sim</i>  .	35
2.2	Resumo dos argumentos disponíveis e valores retornados pela função <i>sim-dataset()</i> .	39
3.1	Medidas resumo para o Índice de Rand Ajustado por método utilizado em todos os cenários simulados.	66
3.2	Medidas resumo para a Variação da Informação por método utilizado em todos os cenários simulados.	68
3.3	Medidas resumo para o Coeficiente de Silhueta por método utilizado em todos os cenários simulados.	68
3.4	Medidas resumo para o Índice de <i>Dunn</i> por método utilizado em todos os cenários simulados.	69
3.5	Medidas resumo para o tempo de execução em segundos por método utilizado em todos os cenários simulados.	70



# Sumário

<b>1</b>	<b>Introdução</b>	<b>15</b>
1.1	Objetivos	17
<b>2</b>	<b>Materiais e Métodos</b>	<b>19</b>
2.1	<i>DBSCAN</i>	20
2.2	<i>K-Means</i>	26
2.3	<i>Fuzzy C-Means (FCM)</i>	27
2.4	<i>Gaussian Mixture Models</i>	29
2.4.1	Estimação via método de máxima verossimilhança	32
2.5	Método <i>Spectral Clustering</i>	33
2.5.1	Grafos de similaridade	33
2.6	<i>MixSim</i> : Um pacote em R para simulação de dados e avaliação da performance de algoritmos de agrupamento	35
2.6.1	Sobreposição de pares	36
2.6.2	Algoritmos de simulação	37
2.6.3	Modelos de mistura e geração de conjuntos de dados	38
2.6.4	Índices de classificação	40
2.7	Validação interna de agrupamentos	42
2.7.1	Coefficiente de silhueta	42
2.7.2	Índice de Dunn	43
<b>3</b>	<b>Comparações em cenários controlados</b>	<b>45</b>
3.1	Variação dos níveis de sobreposição de pares	46
3.1.1	Índice de Rand Ajustado	47
3.1.2	Proporção de Classificações Corretas	48
3.1.3	Variação de Informação	49

3.1.4	Coeficiente de Silhueta . . . . .	51
3.1.5	Índice de <i>Dunn</i> . . . . .	52
3.1.6	Tempo de Execução dos métodos . . . . .	53
3.2	Variação do número de agrupamentos . . . . .	55
3.3	Variação do número de variáveis . . . . .	58
3.4	Variação do número de observações . . . . .	61
3.5	Variações combinadas de parâmetros . . . . .	63
3.6	Conclusões preliminares . . . . .	66
3.7	Comparação dos métodos em agrupamentos não convexos . . . . .	71
3.7.1	Comparando diferentes algoritmos de agrupamento por meio do <i>Toy</i> <i>Datasets</i> . . . . .	71
3.7.2	Comparando diferentes algoritmos de agrupamento com <i>Clustering</i> <i>Datasets</i> . . . . .	74
4	<b>Considerações Finais</b>	<b>77</b>
	<b>Referências Bibliográficas</b>	<b>79</b>



# Capítulo 1

## Introdução

Análise de agrupamentos é uma das técnicas estatísticas amplamente utilizadas para análise exploratória de dados multivariados, sendo aplicada por exemplo, em Matemática, Estatística, Ciência da Computação, bem como Biologia, Ciências Sociais, Psicologia, etc. Sua utilização está presente em todo campo científico devido à sua utilidade na visualização de dados, auxiliando a ter impressões a priori do comportamento do objeto em estudo.

Em pesquisas para classificação de pacientes com câncer, subgrupos são constituídos de acordo com informações úteis para a elaboração de perfil para um diagnóstico da doença. Também é utilizado em Marketing para a identificação de grupos e perfis de consumidores, para otimização de campanhas de anúncios, bem como em *credit scoring*, auxiliando os credores a tomarem decisões na concessão de crédito ([Kassambara 2017](#)).

O problema de agrupamentos consiste em classificar cada uma das observações em grupos específicos de acordo com suas similaridades, enquanto que unidades amostrais alocadas em diferentes grupos diferem quanto a suas características. O objetivo de um algoritmo de agrupamentos é descobrir automaticamente grupos naturais de itens ou variáveis, utilizando medidas quantitativas de associação entre objetos.

Uma das primeiras aparições do emprego de agrupamentos em pesquisas científicas é encontrada em [Pearson e Henrici \(1894\)](#), em que utiliza um método de correspondência de momentos para determinar os parâmetros de mistura de dois componentes de variáveis únicas.

Segundo [Wu \(2012\)](#), existem dois propósitos na utilização de análise de agrupamentos, sendo um deles, o emprego para compreensão de dados, que consiste em encontrar objetos semelhantes com relação a características e, assim, agrupá-los, auxiliando na obtenção de

informações.

Um problema em análises de agrupamentos é a escolha da metodologia que melhor se adapte às características dos dados em diferentes cenários. Há algoritmos que são eficientes em alguns cenários, porém não há um melhor algoritmo para todos os cenários e objetivos, sendo assim necessário entender em quais situações uma determinada metodologia possui melhor ou pior desempenho.

Para o melhor desempenho do algoritmo, utiliza-se diversos procedimentos para calibração e adequação do método, a fim de se mensurar níveis de separação e sobreposição entre agrupamentos. Isso requer medidas objetivas para controle e bancos de dados simulados para realizar as calibrações e ajustes necessários.

A validação dos resultados em uma análise de agrupamentos é uma tarefa necessária, mas desafiadora e pouco utilizada. [Hennig e Liao \(2013\)](#) afirmam que “não há agrupamentos únicos ‘verdadeiros’ ou ‘melhores’ em um conjunto de dados”. Ao invés disso, o usuário pode especificar características relacionadas ao agrupamento a partir de informações a priori. Mesmo que especificações fossem fornecidas quanto ao tamanho, forma e com quais características as observações sejam classificadas como diferentes, existe também a se considerar a escolha do algoritmo para determinado conjunto de dados.

Assim, além da comparação entre diversos métodos de agrupamento, a validação é uma forma de aferir os resultados das análises com avaliações objetivas e quantitativas. A motivação para realizar a validação dos resultados é atribuída à consideração de que todo algoritmo de agrupamento formará subgrupos no conjuntos de dados, mesmo não tendo um contexto de agrupamento natural.

## 1.1 Objetivos

Este trabalho tem como objetivo o estudo de diferentes métodos de agrupamentos de dados, analisando o funcionamento e aplicabilidade de cada um, bem como o estudo sobre a validação dos resultados do método de agrupamento.

Para este estudo, serão gerados dados numéricos através de simulações e funções auxiliares de geração de conjuntos de dados, podendo estes apresentar grande volume. Em seguida, os métodos de clusterização selecionados serão aplicados nesses cenários controlados, a fim de avaliar e comparar suas performances.



# Capítulo 2

## Materiais e Métodos

Esse capítulo tem como objetivo ilustrar o funcionamento dos métodos escolhidos para o estudo e detalhar o funcionamento das medidas de avaliação de clusterizações. Os métodos apresentados a seguir foram escolhidos para contrapor cenários e objetivos diferentes quando se utiliza um algoritmo de clusterização. O *K-Means* foi escolhido por ser simples e amplamente utilizado; o *DBSCAN*, por ser um método de densidade que pode lidar com agrupamentos de formas arbitrárias e com ruídos; o *Fuzzy C-Means*, por permitir que um objeto pertença a mais de um agrupamento; o *Spectral Clustering*, por lidar bem com dados que apresentam estruturas complexas; e o *Gaussian Mixture Model*, por ser um modelo probabilístico que pode lidar com agrupamentos em diversos formatos (diferentes do convexo).

O método *K-Means* é um dos mais populares e amplamente utilizados para a clusterização de dados. Ele funciona dividindo o conjunto de dados em um número pré-determinado de agrupamentos, em que cada objeto é atribuído a um único agrupamento com base em sua proximidade com o centroide do agrupamento. O objetivo do algoritmo é minimizar a soma dos quadrados das distâncias de cada objeto ao seu centroide atribuído.

O *DBSCAN* é um método de clusterização de densidade eficiente em agrupamentos de formas arbitrárias e com ruídos (Ester *et al.* 1996). Ele funciona identificando regiões de alta densidade de objetos e reunindo essas regiões em agrupamentos. O algoritmo é capaz de identificar objetos que não pertencem a nenhum agrupamento, denominados ruídos.

O *Fuzzy C-Means* é um método de clusterização que permite que um objeto pertença a mais de um agrupamento, ou seja, ele pode ter um “grau de pertinência” em cada agrupamento. Ele funciona atribuindo um grau de pertinência a cada objeto pertencer em cada agrupamento, com base em sua proximidade com os centroides dos agrupamen-

tos (Jafelice *et al.* 2012). O objetivo do algoritmo é minimizar a soma ponderada das distâncias dos objetos aos centroides dos agrupamentos.

O *Spectral Clustering* é um método de clusterização eficiente em dados com agrupamentos em formatos complexos, como grupos conectados por pontes estreitas ou conjuntos de dados que se assemelham a um conjunto de anéis concêntricos (Shinnou e Sasaki 2008). Ele funciona convertendo os dados em um espaço de características de maior dimensionalidade e, em seguida, aplicando um algoritmo de clusterização em cima dos dados transformados.

O *Gaussian Mixture Model* é um modelo probabilístico que pode lidar com dados que não possuem uma estrutura clara de agrupamentos (Bouveyron *et al.* 2019). Ele funciona modelando cada agrupamento como uma distribuição normal multivariada e, em seguida, estimando os parâmetros dessas distribuições a partir dos dados. O objetivo do algoritmo é encontrar o modelo que melhor descreve os dados, ou seja, a combinação de distribuições normais que melhor se ajusta aos dados.

Cada um desses métodos possui peculiaridades e objetivos específicos, e a escolha de qual método utilizar depende do tipo de dados a serem agrupados e dos objetivos da análise. Em geral, o *K-Means* é uma boa escolha para conjuntos de dados com estruturas bem definidas e que possuem um número conhecido de agrupamentos (Tan *et al.* 2005). O *DBSCAN* é uma boa escolha para conjuntos de dados com densidades variáveis ou com ruídos (Ester *et al.* 1996). O *Fuzzy C-Means* é uma boa escolha para conjuntos de dados em que a pertinência a mais de um grupo é desejável (Jafelice *et al.* 2012). O *Spectral Clustering* é uma boa escolha para conjuntos de dados que apresentam estruturas complexas (Meilă e Shi 2000) e o *Gaussian Mixture Model* é uma boa escolha para conjuntos de dados com agrupamentos de formatos diversos e que se ajustam bem a uma distribuição (Bouveyron *et al.* 2019).

A seguir, apresentamos os métodos com maiores detalhes sobre origem e funcionamento.

## 2.1 *DBSCAN*

O algoritmo *Density Based Spatial Clustering of Application with Noise* (*DBSCAN*) é um método de agrupamento não paramétrico baseado em densidade, proposto por Ester *et al.* (1996), que visualiza os agrupamentos como áreas de alta densidade separadas

por áreas de baixa densidade. Isso ocorre utilizando distancias entre os pontos e uma quantidade mínima de pontos. Os métodos de agrupamento hierárquico e os que se baseiam em particionamento (como o *K-Means*) são altamente eficientes com *clusters* de formato ovalizado, como apresentado na Figura 2.1. No entanto, quando se trata de *clusters* de formatos arbitrários ou de detecção de pontos *outliers*, como apresentado na Figura 2.2, as técnicas baseadas em densidade são mais eficientes.

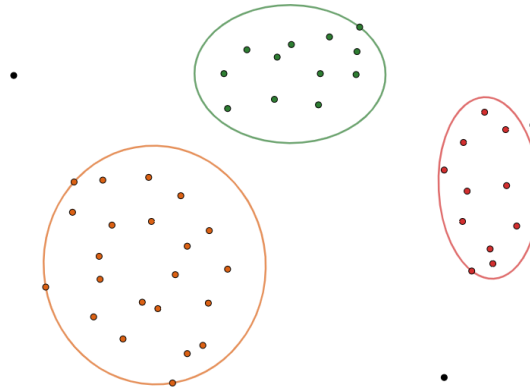


Figura 2.1: Agrupamentos com formato ovalizado.

A ideia principal por trás do **DBSCAN** é que um ponto pertence a um *cluster* se sua vizinhança determinada a partir de um raio pré-estabelecido conter, no mínimo, certo número de pontos (*MinPts*), conforme Equação 2.2, ou seja, a densidade na vizinhança precisa ultrapassar um limite definido.

Devido a essa visão bastante genérica, o **DBSCAN** pode encontrar agrupamentos de qualquer forma, ao contrário de um algoritmo como *K-Means*, que minimiza a soma dos quadrados dentro do *cluster*, algo que funciona melhor para formas convexas, devido ao particionamento a partir dos centroides de cada agrupamento.

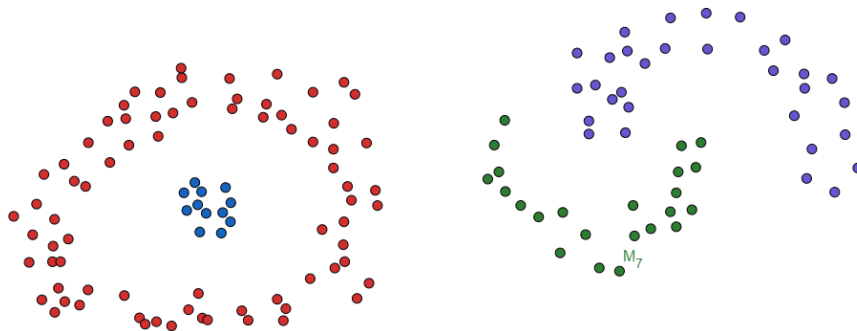


Figura 2.2: Agrupamentos com formatos arbitrários.

Para entender como esse método funciona, listamos algumas definições importantes inerentes à técnica.

### Definição 2.1 (Vizinhança de um ponto)

A vizinhança  $V_\varepsilon$  de um objeto  $p$  é o conjunto de objetos pertencentes a  $D$  com relação ao raio  $\varepsilon$ , isto é,

$$V_\varepsilon(p) = \{\forall q \in D \mid \text{dist}(p, q) \leq \varepsilon\}. \quad (2.2)$$

Uma visão geométrica da vizinhança de  $p$  é mostrada na Figura 2.3.

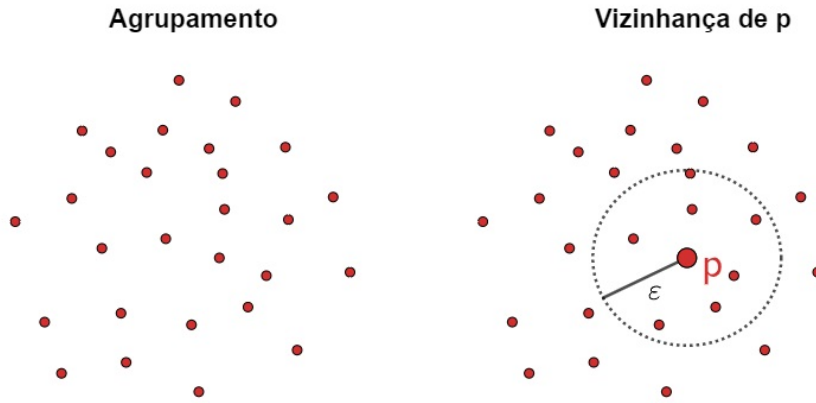


Figura 2.3: Vizinhança  $V_\varepsilon$  dos pontos  $p$  e  $q$ .

### Definição 2.3 (Ponto central)

Se a vizinhança  $V_\varepsilon$  de um objeto  $p$  contém um valor mínimo ( $MinPts$ ) de objetos com respeito ao raio  $\varepsilon$ , então  $p$  é chamado de ponto central.

Como uma derivação da Definição 2.3, se a vizinhança desse objeto não satisfaz  $MinPts$  e contém pelo menos um ponto central, então esse objeto é considerado ponto de borda. A escolha de um valor adequado para  $MinPts$  e do raio  $\varepsilon$  determina quais pontos estão no mesmo agrupamento, mas não delimita os pontos de borda desse agrupamento. Assim, considerando essa definição de ponto central, é possível delimitar as fronteiras do agrupamento.

### Definição 2.4 (Alcance por densidade)





Por meio da conexão por densidade, é possível conectar diversos pontos centrais que são alcançáveis por densidade a partir de um ponto em comum, mas não pertencem a um única cadeia de objetos. Supondo que apenas a Definição 2.4 fosse considerada, diversos subconjuntos se formariam para cada cadeia de objetos.

### Definição 2.6 (Ruído)

Sejam  $C_1, \dots, C_n$  agrupamentos pertencentes ao conjunto de dados  $D$  com respeito aos parâmetros  $V_{\epsilon_i}$  e  $MinPts_i$ ,  $i = 1, \dots, k$ . Chamamos de Ruído o conjunto de pontos pertencentes a  $D$  que não pertencem a nenhum agrupamento  $C_i$ , isto é,  $ruído = \{p \in D \mid \forall i : p \notin C_i\}$ .

Desde que  $C$  contenha ao menos um ponto  $p$ , este precisa ser conectável via densidade pelo menos com ele mesmo através de algum ponto  $O$ , portanto, pelo menos  $O$  satisfaria a condição de ponto central e consequentemente, da  $V_{\epsilon}$ .

**Lema 2.7** *Seja  $p$  um ponto em  $D$  e  $|N_{V_{\epsilon}}(p)| > MinPts$ . Então o conjunto definido por  $O = \{o \mid o \in D \text{ e } O \text{ é alcançável por densidade a partir de } p \text{ com respeito a } MinPts \text{ e } V_{\epsilon}\}$  é um agrupamento com respeito a  $MinPts$  e  $V_{\epsilon}$ .*

Cada ponto pertencente a um agrupamento  $C_i$  é alcançável por densidade a partir de ponto central de  $C_i$ , assim, um agrupamento  $C_i$  contém exatamente os pontos que são alcançáveis via densidade a partir de qualquer ponto central de  $C_i$ .

**Lema 2.8** *Seja  $C$  um agrupamento com respeito ao  $MinPts$  e  $V_{\epsilon}$  e  $p$  um ponto qualquer pertencente a  $C$  com  $|N_{V_{\epsilon}}(p)| > MinPts$ . Então  $C$  é igual ao conjunto  $O = \{o \mid o \in D \text{ e } O \text{ é alcançável por densidade a partir de } p \text{ com respeito a } MinPts \text{ e } V_{\epsilon}\}$ .*

Ester *et al.* (1996) destacam que o algoritmo **DBSCAN** é designado para sempre encontrar os agrupamentos e ruídos em um conjunto de dados, de acordo com as Definições 2.5 e 2.6. Outros métodos de agrupamento como *k-means* atribuíam esses ruídos equivocadamente a algum cluster.

A seguir, apresentamos uma sequência resumida de passos relacionada ao funcionamento do algoritmo **DBSCAN**:

- (i) Para encontrar um agrupamento, o **DBSCAN** começa com um ponto arbitrário  $p$  e recupera todos os pontos alcançáveis por densidade a partir de  $p$  com respeito a

$MinPts$  e  $V_\varepsilon$ , isto é,  $\forall q \in D$  tal que  $dist(p, q) < \varepsilon$  (Lema 2.7);

- (ii) Se  $p$  for um ponto central, isto é,  $V_\varepsilon$  de  $p$  contém ao menos um número mínimo de pontos denotado por  $MinPts$ , então um agrupamento é formado;
- (iii) Se  $V_\varepsilon$  de  $p$  apresenta um número inferior ao  $MinPts$  mas possua um ponto central, então  $p$  é um ponto de fronteira e nenhum ponto é alcançável por densidade a partir de  $p$ . Assim, o **DBSCAN** visita o próximo ponto nos dados;
- (iv) Continue o processo até que todos os pontos tenham sido processados.

Como o **DBSCAN** utiliza-se de valores globais para  $\varepsilon$  e  $MinPts$ , o algoritmo pode fundir dois agrupamentos de acordo com a Definição 2.5 em um único agrupamento, caso dois agrupamentos de densidade diferente estejam “próximos” (Ester *et al.* 1996). Isso ocorre em situações quando existem cadeias de pontos centrais alcançáveis por densidade conectando dois agrupamentos por meio de pontos de fronteira de ambos relativamente próximos um do outro, sendo alcançáveis por diferentes pontos centrais de diferentes agrupamentos na  $V_\varepsilon$  desses pontos.

Os autores em Tran *et al.* (2013) revisam o conceito do **DBSCAN** em vista desse problema de robustez quando há alta densidade e agrupamentos adjacentes. Na tentativa de identificar corretamente os pontos de fronteira, o algoritmo atribui os pontos na área de contato entre os agrupamentos como pontos de fronteira, pois quanto maior é o número de objetos na área de contato, mais objetos são atribuídos como pontos de fronteira de maneira equivocada.

A cadeia de objetos alcançáveis por densidade apresentada na Definição 2.4 na forma  $[p_1, p_2, \dots, p_n]$  também pode ser representada como  $[p_{Central_1}, p_{Central_2}, \dots, p_{Central_n}]$  compreendendo todos os pontos centrais, ou mesmo  $[p_{Central_1}, p_{Central_2}, \dots, p_{Central_{n-1}}, p_{borda}]$  representando o ponto de borda conectado à cadeia de pontos centrais.

Como o ponto de borda não contribui para a expansão do agrupamento através de conexões por densidade, Tran *et al.* (2013) propuseram a desconexão do ponto de borda da cadeia de pontos centrais, passando a utilizar uma cadeia de pontos centrais alcançáveis por densidade, isto é,  $[p_1, p_2, \dots, p_n]$  sendo  $p_i$  um ponto central para todo  $i \leq n$ . Como o número de objetos centrais é o mesmo para cada agrupamento, a expansão do método passa a utilizar uma cadeia de objetos centrais alcançáveis por densidade, deixando os pontos de fronteira não classificados até que todos os pontos centrais de todos os agru-

pamentos sejam classificados e atribuídos. O objeto de fronteira é então atribuído ao agrupamento com a cadeia de objetos centrais mais próxima.

## 2.2 *K-Means*

*K-means* é um algoritmo de agrupamento particional simples baseado em protótipo que tenta encontrar  $K$  agrupamentos não sobrepostos representados por seus centroides (normalmente a média dos pontos nesse agrupamento). Suponha que  $D = \{x_1, \dots, x_n\}$  é o conjunto de dados a ser agrupado. A representação de *k-means* apontada por Wu (2012) é dada a seguir pela função que expressa as distâncias dos pontos dados ao centroide do agrupamento:

$$\min_{\{m_k\}, 1 \leq k \leq K} \sum_{k=1}^K \sum_{x \in C_k} \pi_x \text{dist}(x, m_k), \quad (2.9)$$

em que  $\pi_x$  é o peso de  $x$ ,  $n_k$  o número de objetos de dados designados para o agrupamento  $C_k$ ,  $m_k = \sum_{x \in C_k} \frac{\pi_x x}{n_k}$  é o centroide do agrupamento  $C_k$ ,  $K$  é o número de agrupamentos definidos pelo condutor do algoritmo, e a função "dist" calcula a distância entre os objetos  $X$  e o centroide  $m_k$ ,  $1 \leq k \leq K$ .

O processo de clusterização de *k-means* consiste em:

- (i) Primeiro,  $K$  centroides iniciais são selecionados (aleatoriamente ou de acordo com a inicialização especificada pelo usuário), onde  $K$  é especificado pelo usuário e indica o número desejado de agrupamentos;
- (ii) Cada ponto nos dados é então atribuído ao centroide mais próximo, e cada conjunto de pontos atribuído a um centroide forma um agrupamento;
- (iii) O centroide de cada agrupamentos é então atualizado com base nos pontos atribuídos a esse agrupamento;
- (iv) Este processo é repetido até que nenhum ponto é capaz de mudar os agrupamentos.

O processo de clusterização pelo método *k-means* depende dos valores iniciais escolhidos para os centroides. Em alguns casos, se os valores iniciais estiverem muito distantes dos centroides ideais, o *k-means* pode não convergir para uma solução ótima ou demorar mais para alcançá-la. Essa dependência inicial pode ser um desafio para aplicação do método em algumas situações (Bishop 2006). Para evitarmos esses problemas de inicialização, será utilizado nesse estudo o algoritmo *k-means++* (Arthur e Vassilvitskii 2007).

## 2.3 *Fuzzy C-Means (FCM)*

A teoria de conjuntos *fuzzy* foi, inicialmente, apresentada por Zadeh (1965) como uma alternativa para a teoria clássica, permitindo uma avaliação gradual da associação de elementos de um conjunto, em contraste ao uso de termos binários - um elemento pertence ou não pertence ao conjunto.

Com o desenvolvimento da teoria *fuzzy*, o algoritmo *FCM* de agrupamento baseia-se na teoria de agrupamento proposta por Ruspini (1977). Este algoritmo é usado para análise com base na distância entre vários pontos dos dados de entrada. Os agrupamentos são formados de acordo com a distância entre os pontos e o centroide dos agrupamentos.

Um subconjunto *fuzzy*  $\mathcal{F}$  de um conjunto  $\mathcal{U}$  é caracterizado por uma função de pertinência  $u_F$  que associa a cada elemento  $x \in \mathcal{U}$  um grau de pertinência  $u_F(x) \in [0, 1]$  (Jafelice *et al.* 2012). Em outras palavras, a função

$$u_F : \mathcal{U} \rightarrow [0, 1],$$

atribui o grau com que o elemento  $x$  pertence ao subconjunto *fuzzy*  $F$ .

Um conjunto composto por  $N$  subconjuntos caracterizados a partir da função de pertinência  $u_{k_i}$  é representado pela seguinte restrição:

$$\begin{aligned} \mathcal{U}_f &= \{U = (u_{k_i}) : \sum_{j=1}^c u_{k_j} = 1, 1 \leq k \leq N; \\ &u_{k_i} \in [0, 1], 1 \leq k \leq N, 1 \leq i \leq c\}. \end{aligned}$$

Seja uma matriz de pertinência de dimensão  $N \times c$  dada por  $U = (u_{k_i}), 1 \leq k \leq N, 1 \leq i \leq c$ , em que  $u_{k_i}$  assume um valor real. Vamos assumir que  $u_{k_i}$  tem valor baseado na lógica *fuzzy*

$$u_{k_i} \in [0, 1], \quad 1 \leq k \leq N, 1 \leq i \leq c.$$

Assim, cada componente de  $U$  mostra o grau de associação pela função de pertinência de um objeto para um agrupamento, de forma que os pontos situados na borda de um agrupamento podem estar presentes em um grau menor do que pontos situados no centro do agrupamento.

A principal diferença entre o *FCM* e outros algoritmos de clusterização, como o *K-Means* por exemplo, é que o *FCM* permite que um objeto pertença parcialmente a vários

*clusters*, em vez de ser atribuído a apenas um.

O *FCM* começa selecionando aleatoriamente  $K$  centroides para representar cada um dos  $K$  agrupamentos iniciais. Em seguida, o algoritmo calcula a distância entre cada objeto e cada centroide e usa essa informação para determinar a “pertinência” de cada objeto pertencer a cada agrupamento. Essas “pertinências” são chamadas de “graus de pertinência” e são usadas para ajustar a posição dos centroides dos agrupamentos.

O processo é repetido iterativamente, com a atualização dos graus de pertinência e dos centroides dos agrupamentos, até que uma condição de parada seja atendida (por exemplo, um número máximo de iterações ou uma alteração mínima na posição dos centroides).

O algoritmo *Fuzzy C-Means* é baseado na minimização da função objetivo

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|\mathbf{x}_i - \mathbf{v}_j\|^2,$$

na qual  $N$  é o número de observações,  $C$  é o número de agrupamentos,  $m$  é o parâmetro de “fuzzificação” que controla o grau de “borramento” das partições,  $\mathbf{x}_i$  é o vetor de características da  $i$ -ésima observação,  $\mathbf{v}_j$  é o vetor de centroides do  $j$ -ésimo cluster e  $u_{ij}$  é o grau de pertinência da  $i$ -ésima observação ao  $j$ -ésimo agrupamento, variando entre 0 e 1. A ideia do algoritmo é encontrar os valores de  $u_{ij}$  e  $\mathbf{v}_j$  que minimizam  $J_m$ .

O algoritmo começa inicializando aleatoriamente os valores de  $u_{ij}$  e  $\mathbf{v}_j$ . Então, o valor de  $u_{ij}$  é atualizado iterativamente de acordo com a seguinte equação:

$$u_{ij} = \left[ \sum_{k=1}^C \left( \frac{\|\mathbf{x}_i - \mathbf{v}_j\|}{\|\mathbf{x}_i - \mathbf{v}_k\|} \right)^{\frac{2}{m-1}} \right]^{-1}.$$

Em seguida, os valores de  $\mathbf{v}_j$  são atualizados de acordo com a seguinte equação:

$$\mathbf{v}_j = \frac{\sum_{i=1}^N u_{ij}^m \mathbf{x}_i}{\sum_{i=1}^N u_{ij}^m}.$$

O processo de atualização de  $u_{ij}$  e  $\mathbf{v}_j$  é repetido até que a função objetivo  $J_m$  convirja ou até que um número máximo de iterações seja atingido. Ao final do processo, cada observação é atribuída ao cluster com maior grau de pertinência  $u_{ij}$ .

## 2.4 Gaussian Mixture Models

De acordo com Bouveyron *et al.* (2019), as primeiras aparições bem sucedidas dessa metodologia foram desenvolvidas no início dos anos 50 em estatística multivariada discreta aplicada em Sociologia. A maioria dos métodos de agrupamentos criados anteriormente era heurístico e algorítmico, baseando-se em uma matriz de similaridade entre objetos.

A proposta de um método de agrupamento baseado em modelos de probabilidade é dividir ou particionar os dados em grupos, de modo que os objetos de um mesmo grupo sejam similares entre si e distintos dos objetos de outros grupos. O principal modelo estatístico para agrupamento é o modelo de mistura finita, em que cada grupo é modelado a partir de sua própria distribuição de probabilidade.

A utilização de modelos de probabilidade como base para a análise de agrupamento tem algumas vantagens significativas. A análise fica mais próxima da metodologia estatística, sendo possível aplicar técnicas de inferência para auxiliar no modelo de probabilidade que melhor corresponde ao método de agrupamento escolhido.

Considere um conjunto de  $n$  variáveis aleatórias,  $Y_1, \dots, Y_n$ , em que cada variável é um vetor de dimensão  $d$ ,  $Y_i = (Y_{i1}, \dots, Y_{id})$ . Um modelo de mistura finita representa a distribuição de uma variável  $Y_i$ , como uma mistura finita ou média ponderada de  $G$  funções de densidade de probabilidade, chamadas componentes da mistura (Bouveyron *et al.* 2019), ou seja,

$$p(Y = y_i) = \sum_{g=1}^G \tau_g f_g(y_i \mid \theta_g),$$

sendo  $\tau_g$  a probabilidade de uma observação a ser gerada pelo  $g$ -ésimo componente,  $\tau \geq 0$ ,  $g = 1, \dots, G$ , e  $\sum_{g=1}^G \tau_g = 1$ , enquanto que  $f_g(\cdot, \theta_g)$  representa a função de densidade do  $g$ -ésimo componente dado os valores de seus parâmetros  $\theta_g$ .

A Figura 2.6 mostra as funções de densidade de duas componentes normais univariadas e a função densidade resultante da mistura finita dos dois componentes. As funções de densidade individuais dos componentes multiplicadas por suas probabilidades de mistura são mostradas em vermelho e azul, respectivamente, e a função de densidade geral da mistura resultante, que é a soma das curvas vermelha e azul, é a curva preta. Os pontos mostram uma amostra de 200 observações gerada a partir de uma distribuição normal bivariada, com as cores indicando o componente da mistura a partir do qual foram gerados. O código utilizado para construção desse procedimento pode ser encontrado em <https://github.com/BadaroMath/Cluster-Comparison>, e tem como referência os

códigos fornecidos por (Bouveyron *et al.*, 2019, pg. 17).

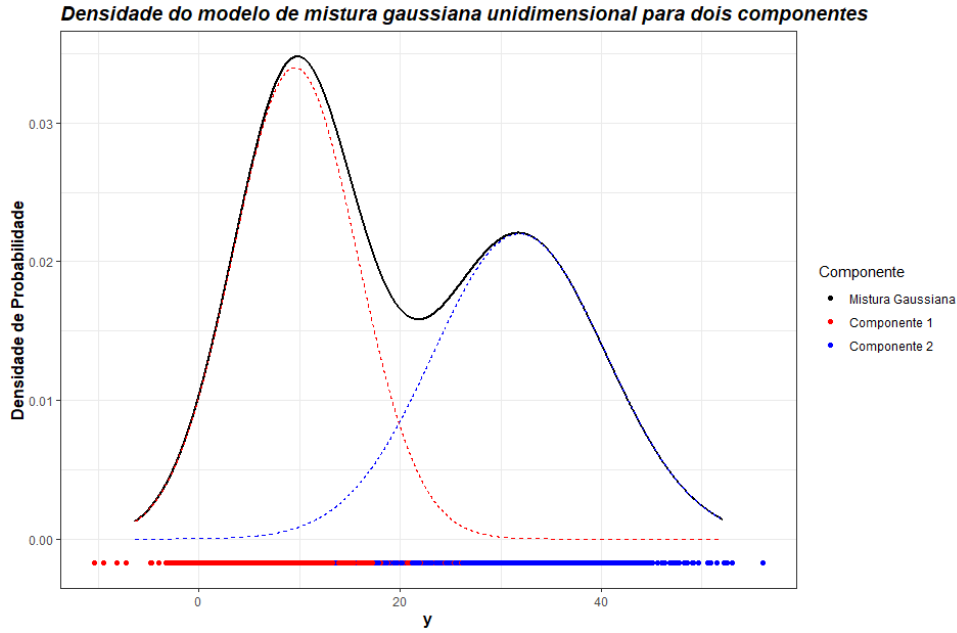


Figura 2.6: Funções de densidade de distribuições normais univariadas dos dois componentes de uma mistura finita bidimensional.

Pode-se afirmar que  $f_g$  é uma função de densidade de uma distribuição normal multivariada. Quando  $y_i$  é unidimensional,  $f_g(y_i|\theta_g)$  é uma função de densidade com distribuição  $N(\mu_g, \sigma_g^2)$ , e  $\theta_g = (\mu_g, \sigma_g)$  sendo respectivamente a média e desvio padrão para o  $g$ -ésimo componente de mistura.

Quando os dados são multivariados, a densidade da  $g$ -ésima componente é também uma normal multivariada ou Gaussiana parametrizada pelo vetor de médias  $\mu_g$  e pela matriz de covariâncias  $\Sigma_g$ , isto é

$$\Phi_g(y_i|\mu_g, \Sigma_g) = |2\pi\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2}(y_i - \mu_g)^T \Sigma_g^{-1} (y_i - \mu_g) \right\}.$$

Os parâmetros do modelo podem ser estimados usando o algoritmo *Expectation-Maximization* (EM). Cada agrupamento  $k$  é centrado nas médias  $\mu_k$ , com densidade aumentada para pontos próximos à média. Características geométricas como forma, volume e orientação de cada agrupamento são determinadas pela matriz de covariâncias  $\Sigma_k$  (Kassambara 2017).

A Figura 2.7 apresenta os contornos da função densidade de uma distribuição normal bivariada associado a um modelo de mistura finito bidimensional com dois componentes de mistura. Os pontos correspondem a uma amostra de 200 observações gerada a partir das funções de densidade de uma distribuição normal bivariada, cujas cores indicam o



componente da mistura a partir do qual foram gerados. O código utilizado para construção desse procedimento pode ser encontrado em <https://github.com/BadaroMath/Cluster-Comparison>, e tem como referência os códigos fornecidos por (Bouveyron *et al.*, 2019, pg. 20).

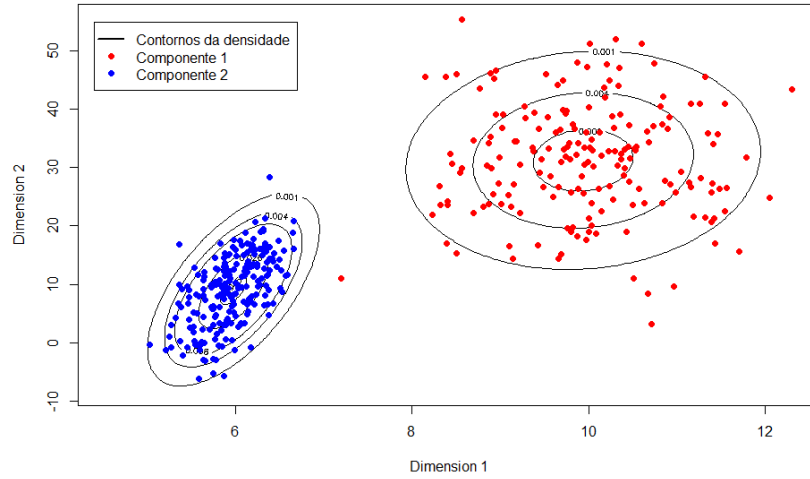


Figura 2.7: Contornos da função de densidade de uma distribuição normal bivariada associada a um modelo de mistura finito bidimensional com dois componentes de mistura.

Na [Figura 2.7](#) nota-se uma natureza elipsoidal dos contornos dos quantis da densidade dos mesmos dados gerados, a partir de misturas de componentes multivariados sem sobreposição. O uso de modelos de mistura finita normal multivariada é bastante útil em diversas aplicações, porém trabalhar com todos os parâmetros é custoso. Nesse exemplo, seriam  $(G - 1) + Gd + G\{d(d + 1)/2\} = 11$ , sendo  $G$  o número de componentes e  $d$  o número de dimensões cada.

Uma maneira de tornar esse problema menos custoso é o uso da decomposição *Spectral* da matriz de covariâncias  $\Sigma_g$

$$\Sigma_g = \lambda_g D_g A_g D_g^T, \quad (2.10)$$

sendo  $\lambda_g$  a constante de proporcionalidade,  $D_g$  a matriz dos autovetores de  $\Sigma_g$  e  $A_g$  a matriz em que os elementos da diagonal são proporcionais aos autovalores de  $\Sigma_g$ . Cada fator na decomposição corresponde a uma propriedade geométrica na mistura da  $g$ -ésima componente, sendo  $D_g$  responsável pela orientação em  $R^d$ , a matriz diagonal  $A_g$  determina o formato e a constante de proporcionalidade atua sobre o volume em que a  $g$ -ésima componente de mistura ocupará, considerando o espaço  $R^d$ .

### 2.4.1 Estimação via método de máxima verossimilhança

Uma variável aleatória latente é aquela cujos valores são desconhecidos ou não observáveis diretamente. Existem duas abordagens para lidar com variáveis aleatórias latentes em uma estrutura de máxima verossimilhança; a probabilidade dos dados observados e a probabilidade dos dados completos.

[Bouveyron et al. \(2019\)](#) afirmam que a função de verossimilhança dos dados completos é uma abordagem geral, em que os dados consistem em  $n$  observações multivariadas  $(y_i, z_i)$ , em que  $y_i$  é observado e  $z_i$  não é observado. Assim, a função de verossimilhança é construída considerando todos os valores possíveis das variáveis aleatórias latentes e suas probabilidades associadas, de acordo com os dados observados.

Se  $(y_i, z_i)$  são independentes e identicamente distribuídos (*i.i.d.*) de acordo com uma distribuição de probabilidade  $f$  com parâmetros  $\theta$ , então a verossimilhança de dados completos é dada por

$$\mathcal{L}_C(y, z \mid \theta) = \prod_{i=1}^n f(y_i, z_i \mid \theta),$$

sendo  $y = (y_1, \dots, y_n)$  e  $z = (z_1, \dots, z_n)$ .

A função de verossimilhança dos dados observados,  $\mathcal{L}_O(y \mid \theta)$ , pode ser obtida in-

tegrando a função de verossimilhança dos dados completos em relação aos dados não observados  $z$ ,

$$\mathcal{L}_O(y \mid \theta) = \int \mathcal{L}_C(y, z \mid \theta) dz.$$

O estimador de máxima verossimilhança de  $\theta$  baseado nos dados observados maximiza a função  $\mathcal{L}_O(y \mid \theta)$ .

A função de verossimilhança da mistura é dada por

$$\mathcal{L}_O(y \mid \theta) = \prod_{i=1}^n \sum_{g=1}^G \tau_g f_g(y_i \mid \theta_g) = \prod_{i=1}^n \sum_{g=1}^G \tau_g \Phi_g(y_i \mid \mu_g, \Sigma_g).$$

O algoritmo EM alterna-se entre duas etapas: a primeira é uma “etapa E” ou etapa de “Esperança”, na qual a esperança condicional da log-verossimilhança dos dados completos de acordo com os dados observados e as estimativas de parâmetros atuais é computado. A segunda é uma “etapa M” ou etapa de “Maximização”, na qual os parâmetros que maximizam o log da verossimilhança esperada da “etapa E” são determinados ([Bouveyron et al. 2019](#)).

## 2.5 Método *Spectral Clustering*

O agrupamento *Spectral* remonta a [Donath e Hoffman \(1973\)](#), os primeiros a sugerirem a construção de um grafo com partições baseadas em autovetores da matriz de adjacência. No mesmo ano, [Fiedler \(1973\)](#) descobriu que as bi-partições de um grafo estão intimamente conectadas com o segundo autovetor do grafo Laplaciano, sugerindo a utilização de autovetores para particionar um grafo. Desde então, o agrupamento Spectral foi descoberto, redescoberto e ampliado muitas vezes em diferentes comunidades

### 2.5.1 Grafos de similaridade

Existem duas maneiras comuns de definir a similaridade entre pontos: a primeira é a similaridade por  $\epsilon$ -vizinhança, onde pontos que estão a uma distância menor do que um certo parâmetro  $\epsilon$  são conectados, e a segunda é a similaridade por  $k$ -vizinhos mais próximos, onde um ponto é conectado aos seus  $k$ -vizinhos mais próximos.

Uma vez que o grafo é construído, é possível encontrar uma partição do grafo em que as bordas entre os diferentes grupos possuem pesos muito baixos (o que significa que pontos

em diferentes agrupamentos são diferentes uns dos outros), e se essas bordas dentro de um grupo possuírem pesos elevados, então os pontos dentro do mesmo agrupamento são semelhantes entre si.

O agrupamento *Spectral* utiliza autovetores da matriz de adjacência ou da matriz Laplaciana do grafo para encontrar a partição do grafo. A ideia básica é que os autovetores correspondentes aos menores autovalores da matriz de Laplaciana podem ser usados para particionar o grafo em  $k$  grupos, onde  $k$  é o número desejado de agrupamentos. Essa técnica é útil para identificar estruturas não lineares nos dados e tem sido amplamente utilizada em muitas áreas, incluindo análise de imagem, aprendizado de máquina e ciência de redes.

### Algoritmo de clusterização *Spectral*

O algoritmo apresentado a seguir ([Weiss 1999](#) e [Meilă e Shi 2000](#)) utiliza os  $k$  autovetores simultaneamente de uma matriz derivada da distância entre os pontos.


Definindo  $S = \{s_1, \dots, s_n\}$  pertencente ao conjunto de pontos objeto de interesse para agrupamentos em  $k$  subgrupos de  $\mathbb{R}^m$ :

- (i) Construa um gráfico de similaridade por uma das maneiras descritas na [Subseção 2.5.1](#).  
Seja  $W$  a matriz de adjacência ponderada;
- (ii) Construção da matriz Laplaciana  $L = D - W$ ;
- (iii) Encontrar valores para  $u_1, u_2, \dots, u_k$ , o maior valor de  $k$  autovetores da matriz  $L$  e formação da matriz  $U = [u_1, u_2 \dots u_K] \in \mathbb{R}$  a partir dos autovetores obtidos;
- (iv) Para  $i = 1, \dots, n$ , seja  $y_i \in k$  o vetor correspondente à  $i$ -ésima linha de  $U$ ;
- (v) Tratar cada uma dos vetores como pontos pertencentes a  $\mathbb{R}^K$ , realizar o agrupamento utilizando *K-Means* ou outro algoritmo que atenda a minimização de distorção.

## 2.6 *MixSim*: Um pacote em R para simulação de dados e avaliação da performance de algoritmos de agrupamento

De acordo com [Melnykov et al. \(2012\)](#), o pacote *MixSim* é uma ferramenta que permite simular misturas de distribuições gaussianas com diferentes níveis de sobreposição entre os componentes da mistura. A chamada sobreposição de pares, definido como uma soma de duas probabilidades de classificação incorreta, mede o grau de interação entre componentes e podem ser prontamente empregados para controlar a complexidade do agrupamento de conjuntos de dados simulados de misturas.

A [Tabela 2.1](#) mostra os argumentos e valores disponíveis retornados pela função *MixSim()*.

Tabela 2.1: Resumo dos argumentos e valores disponíveis retornados pela função *MixSim* .

Argumentos	Descrição
BarOmega	Sobreposição média par-a-par $\bar{\omega}$
MaxOmega	Sobreposição máxima par-a-par $\tilde{\omega}$
K	Número de componentes da mistura
p	Número de dimensões
sph	Componentes da mistura esféricos (TRUE) ou não-esféricos (FALSE)
hom	Componentes da mistura homogêneos (TRUE) ou não-homogêneos (FALSE)
ecc	Excentricidade máxima
PiLow	Menor proporção de mistura
int	Lado de um hipercubo para simulação de vetores médios
resN	Número máximo de resimulações da mistura
eps	Limite de erro
lim	Número máximo de termos de integração de acordo com <a href="#">Davies (1980)</a>
Valores	Descrição
\$Pi	Vetor de proporções de mistura
\$Mu	Vetores médios
\$S	Matrizes de covariância
\$OmegaMap	Mapa de probabilidades de má classificação
\$BarOmega	Sobreposição média par-a-par $\bar{\omega}$
\$MaxOmega	Sobreposição máxima par-a-par $\tilde{\omega}$
\$srcMax	Índice do par de clusters que produz a maior sobreposição
\$fail	Indicador de conclusão bem-sucedida

### 2.6.1 Sobreposição de pares

A noção de sobreposição de pares foi recentemente introduzida por [Melnykov e Maitra \(2010\)](#). Aqui, explicamos brevemente o funcionamento por trás da medida e suas propriedades.

Seja  $X$  uma variável aleatória com distribuição de acordo com o modelo de mistura finita  $g(x) = \sum_{k=1}^K \pi_k \phi(x; \mu_k, \Sigma_k)$ , em que  $\phi(x; \mu_k, \Sigma_k)$  é uma densidade gaussiana multivariada da  $k$ -ésima componente com vetor médio  $\mu_k$  e matriz de covariância  $\Sigma_k$ . Então, a sobreposição entre o  $i$ -ésimo e a  $j$ -ésimo componente é definida como  $\omega_{ij} = \omega_{i|j} + \omega_{j|i}$ , onde  $\omega_{j|i}$  é a probabilidade de classificação incorreta de que a variável aleatória  $X$  originou-se da  $i$ -ésima componente, mas foi erroneamente atribuída à  $j$ -ésima componente;  $\omega_{i|j}$  é definido de forma análoga. Assim,  $\omega_{i|j}$  é dado por

$$\omega_{i|j} = P[\pi_i \phi(X; \mu_i, \Sigma_i) < \pi_j \phi(X; \mu_j, \Sigma_j) | X \sim N(\mu_i, \Sigma_i)].$$

De maneira genérica, a probabilidade de classificação incorreta são fornecidas por

$$\begin{aligned} \omega_{i|j} &= P_{N(\mu_i, \Sigma)}(\mu_i, \Sigma_i) \left[ \sum_{\substack{l=1 \\ l: \lambda_l \neq 1}}^p (\lambda_l - 1) U_l + 2 \sum_{\substack{l=1 \\ l: \lambda_l = 1}}^p \delta_l W_l \right. \\ &\quad \left. \leq \sum_{\substack{l=1 \\ l: \lambda_l \neq 1}}^p \frac{\lambda_l \delta_l^2}{\lambda_l - 1} - \sum_{\substack{l=1 \\ l: \lambda_l \neq 1}}^p \delta_l^2 + \log \frac{\pi_j^2 |\Sigma_i|}{\pi_i^2 |\Sigma_j|} \right], \end{aligned}$$

em que  $\lambda_1, \lambda_2, \dots, \lambda_p$  são autovalores da matriz  $\Sigma_i^{-\frac{1}{2}} \Sigma_i^{-1} \Sigma_i^{\frac{1}{2}}$  e  $\gamma_1, \gamma_2, \dots, \gamma_p$  são os autovetores correspondentes,  $U_l' S$  são variáveis aleatórias independentes com distribuição  $\chi^2$  um grau de parâmetro de liberdade e não centralidade dado por  $\lambda_l^2 \delta_l^2 / (\lambda_l - 1)^2$  com  $\delta_l = \gamma_l' \Sigma_i^{-\frac{1}{2}} (\mu_i - \mu_j)$ , independente de  $W_l' S$ , que são variáveis aleatórias independentes que seguem distribuição normal padrão.

Adicionando uma constante  $c$  positiva multiplicando as matrizes de covariância  $\Sigma$ , ocasionaria uma inflação ( $c > 1$ ) ou deflação ( $c < 1$ ) dos componentes. Assim, podemos manipular o valor de  $c$  para atingir o nível pré-especificado de sobreposição  $\omega_{ij}(c)$  entre os componentes.

## 2.6.2 Algoritmos de simulação

Os algoritmos a seguir foram desenvolvidos por [Melnykov e Maitra \(2010\)](#) para simulação de misturas Gaussianas de acordo com valores pré fornecidos da sobreposição média de  $(\bar{\omega})$  e/ou sobreposição máxima de  $(\tilde{\omega})$ .

O primeiro algoritmo funciona controlando a sobreposição média **ou** a sobreposição máxima dos pares, enquanto que o segundo controla ambos os parâmetros simultaneamente.

### Gerando modelos de mistura controlando $(\bar{\omega})$ ou $(\tilde{\omega})$

- (i) **Gerando parâmetros iniciais.** Gere  $K$  proporções de mistura, vetores de médias e matrizes de covariância como apresentado em 2.4. Calcule o limite da sobreposição média  $(\bar{\omega}^\infty)$  (limite da sobreposição máxima  $(\tilde{\omega}^\infty)$ ). Se  $\bar{\omega} > \bar{\omega}^\infty$  ( $\tilde{\omega} > \tilde{\omega}^\infty$ ), descarte a realização e inicie a primeira etapa novamente;
- (ii) **Calculando sobreposições de pares.** Calcule todas as sobreposições de pares. Calcule a estimativa atual de  $\hat{\omega}$  ( $\hat{\tilde{\omega}}$ ). Se a diferença entre  $\hat{\omega}$  e  $\bar{\omega}$  ( $\hat{\tilde{\omega}}$  e  $\tilde{\omega}$ ) for insignificante, o algoritmo para e fornece os parâmetros atuais;
- (iii) **Dimensionando clusters.** Use técnicas de localização de raízes para encontrar um multiplicador de matriz de covariância  $c$  de forma que a diferença entre  $\hat{\omega}(c)$  e  $\bar{\omega}$  ( $\hat{\tilde{\omega}}(c)$  e  $\tilde{\omega}$ ) é desprezível.

O algoritmo a seguir é preferível para melhor controle da sobreposição entre as componentes dos dados simulados.

### Gerando modelos de mistura controlando $(\bar{\omega})$ e $(\tilde{\omega})$

- (i) **Dimensionando os *clusters* para encontrar  $\tilde{\omega}$ :** Utilizar o primeiro algoritmo para obter o conjunto de parâmetros que satisfaça  $\tilde{\omega}$  e fixar os dois componentes que produziram a maior sobreposição; as matrizes de covariância não estarão envolvidas no processo de inflação/deflação;
- (ii) **Encontrar  $\mathbf{c}_v$ :** Encontre o maior valor de  $c$  (definido como sendo  $\mathbf{c}_v$ ) de forma que nenhuma das sobreposições de pares  $\omega_{ij}(\mathbf{c}_v)$  exceda  $\tilde{\omega}$ . Se  $\hat{\tilde{\omega}}(\mathbf{c}_v) < \bar{\omega}$ , descarte a realização e volte para o item (i);

- (iii) **Dimensionamento limitado:** Enquanto mantém os dois componentes fixos inalterados, aplique o item (iii) do primeiro algoritmo para o resto dos componentes para atingir o valor desejado para  $\bar{\omega}$ . Se os parâmetros obtidos satisfazem  $\bar{\omega}$  e  $\tilde{\omega}$ , retornar eles na saída. Caso contrário, comece com item (i) novamente.

### 2.6.3 Modelos de mistura e geração de conjuntos de dados

De acordo com Melnykov *et al.* (2012), a função *MixSim()* é a principal do pacote e é responsável por encontrar um modelo de mistura gaussiano que atenda ao nível de sobreposição médio e/ou máximo especificado pelo usuário. O comando possui a seguinte sintaxe:

```
MixSim(BarOmega = NULL, MaxOmega = NULL, K, p, sph = FALSE, hom = FALSE,
ecc = 0.90, PiLow = 1.0, int = c(0.0, 1.0), resN = 100, eps = 1e-06,
lim = 1e06)
```

Os parâmetros da função estão listados na Tabela 2.1. Quando ambos os parâmetros *BarOmega* e *MaxOmega* são especificados, é executado o segundo algoritmo da Seção 2.6.2. Se apenas um dos parâmetros acima for fornecido, o primeiro algoritmo da Seção 2.6.2 é utilizado. O menor número permitido de componentes *K* é 2; neste caso, *BarOmega*  $\equiv$  *MaxOmega*. *MixSim()* permite a simulação de misturas com estrutura de covariância esférica ou geral, bem como com componentes homogêneos ou não homogêneos. Para melhor controle da forma dos componentes produzidos, o parâmetro *ecc* que especifica a máxima excentricidade pode ser usado.

A função responsável pela geração de dados é chamada de *simdataset()* e tem a seguinte forma:

```
simdataset(n, Pi, Mu, S, n.noise = 0, n.out = 0, alpha = 0.001,
max.out = 100000, int = NULL, lambda = NULL)
```

Os argumentos e valores retornados estão listados na Tabela 2.2. Os parâmetros *Pi*, *Mu* e *S* têm o mesmo significado que antes. O tamanho de um conjunto de dados gerado é definido como  $n + n.out$ , em que *n.out* especifica o número de valores atípicos necessários. Se o parâmetro *n.out* não for especificado, nenhum valor atípico é produzido por *simdataset*. O parâmetro *max.out* especifica o número máximo de re-simulações de valores atípicos, com o valor padrão de  $1e05$ . O parâmetro *alpha* especifica os contornos



elipsoidais além dos quais os valores atípicos precisam ser simulados. O número de dimensões para o conjunto de dados é definido como  $\dim(\text{Mu})[2] + n.\text{noise}$ . Por padrão,  $n.\text{noise} = 0$ . O intervalo *int* define um lado de um hiper-cubo para a simulação de valores atípicos. Ele também é usado para simular variáveis de ruído se  $n.\text{noise}$  for maior que 0. Tanto os valores atípicos quanto as variáveis de ruído são simulados a partir de um hiper-cubo uniforme. Quando *int* não é fornecido, os limites do intervalo são escolhidos como iguais ao menor e ao maior valor em cada dimensão dos dados.

Tabela 2.2: Resumo dos argumentos disponíveis e valores retornados pela função *simdataset()*.

Argumentos	Descrição
n	Tamanho da amostra $\bar{\omega}$
Pi	Vetor de proporções de mistura $\check{\omega}$
Mu	Vetores de média
S	Matrizes de covariância
n.noise	Número de variáveis de ruído
n.out	Número de valores atípicos
alpha	Nível para o contorno 1 - alpha para simular <i>outliers</i>
max.out	Número máximo de tentativas para simular <i>outliers</i>
int	Intervalo para simular valores atípicos e/ou variáveis de ruído
lambda	Vetor de coeficientes para uma transformação inversa de Box-Cox
Valores	Descrição
\$X	Conjunto de dados produzido
\$id	Vetor de classificação para o conjunto de dados produzido

O seguinte código ilustra como a função *simdataset()* pode ser usada para simular dados da mistura obtida por *MixSim*. Aqui, obtemos uma mistura bidimensional com 5 componentes e uma sobreposição média de 0,05. Em seguida, são simuladas 5000 observações a partir da mistura. Os seguintes comandos geram a [Figura 2.8](#):

```
set.seed(12)
Q <- MixSim(BarOmega = 0.05, K = 4, p = 2)
A <- simdataset(n = 5000, Pi = Q$Pi, Mu = Q$Mu, S = Q$S)

colors <- c("red", "blue", "orange", "purple")

simex <- A |>
  as.data.frame() |>
  mutate(id = as.factor(id)) |>
```

```
ggplot(aes(x = X.1, y = X.2, color = id)) +
  theme_classic()+
  ylab("Variável 1.")+
  xlab("Variável 2.")
  geom_point(size = 1, shape = 21) +
  scale_color_manual(values = colors,name = "Agrupamento:")
```

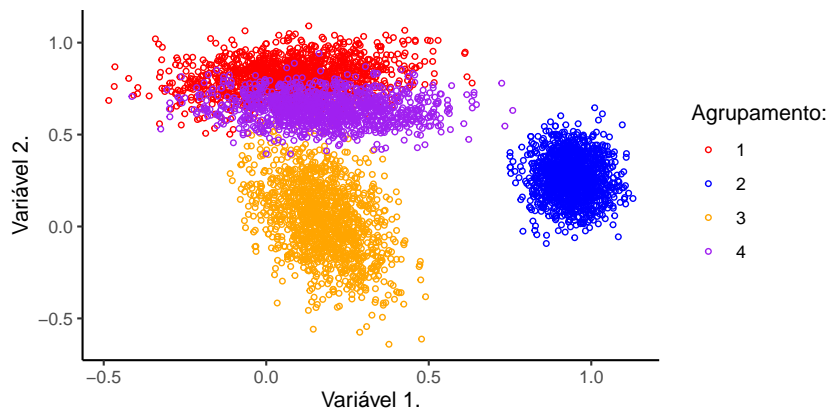


Figura 2.8: Conjuntos de dados simulados obtidos a partir de modelos de mistura gaussiana.

### 2.6.4 Índices de classificação

O pacote *MixSim* possui múltiplos índices que foram desenvolvidos para uma investigação sistemática das propriedades de algoritmos de agrupamento (Meilă 2007). Para isso, serão usadas algumas medidas para estimar o nível de similaridade entre vetores particionados.

De acordo com Melnykov *et al.* (2012), o primeiro grupo de índices compara agrupamentos contando os pares de pontos que são atribuídos ao mesmo ou diferentes agrupamentos em ambas as partições. O índice de Rand (1971) se encaixa nessa categoria, e é definido como

$$R(c_1, c_2) = \frac{N_{11} + N_{00}}{\binom{n}{2}},$$

sendo  $c_1$  e  $c_2$  o primeiro e o segundo vetor de partição, respectivamente e  $N_{11}$  o número de pares de pontos no mesmo agrupamento em  $c_1$  e  $c_2$ , e  $N_{00}$  é o número de pares em diferentes agrupamentos em  $c_1$  e  $c_2$ ;  $n$  representa o número de pontos em vetores de partição.

A modificação mais utilizada de  $R(c_1, c_2)$  envolve o ajuste com o uso de  $E(R(c_1, c_2))$ , fornecendo o índice de Rand ajustado (Hubert e Arabie 1985):

$$AR(c_1, c_2) = \frac{R(c_1, c_2) - E(R(c_1, c_2))}{1 - E(R(c_1, c_2))}.$$

Os índices  $R$  e  $AR$  têm limites superiores igual a 1 que pode ser alcançado apenas no caso do mesmo particionamento, ou seja,  $c_1 = c_2$ . O índice  $AR$ , juntamente com outros três índices provenientes de ajustes do índice de Rand estão presentes na função  $RandIndex()$  do *MixSim* (veja Melnykov *et al.* (2012) para mais detalhes).

Um segundo grupo de índices compara as partições por combinação de conjuntos. O mais conhecido índice aqui é a proporção de observações que concordam com a classificação para ambos os particionamentos de vetores. Pode-se notar que a troca de rótulo desempenha um papel importante aqui, pois o resultado é dependente do rótulo. A função  $ClassProp()$  do *MixSim* calcula essa proporção considerando todas as possíveis permutações de rótulos e escolhendo a permutação que produz a maior proporção de acordo com as duas partições.

Por fim, temos os índices que se baseiam na análise de variação de informação entre dois particionamento de vetores. Meilă (2007) desenvolveu esse índice definido como

$$VI(c_1, c_2) = H(c_1) + H(c_2) - 2I(c_1, c_2),$$

sendo  $H(c_i)$  a entropia associada a  $c_i$  definida como

$$H(c_i) = - \sum_{k=1}^{K^{(i)}} \frac{n_K^{(i)}}{n} \log \frac{n_K^{(i)}}{n}, \quad i = 1, 2.$$

A quantidade  $I(c_1, c_2)$  representa a informação mutua entre dois particionamentos e é definida como

$$I(c_1, c_2) = \sum_{k=1}^{K^{(1)}} \sum_{r=1}^{K^{(2)}} \frac{n_{kr}}{n} \log \frac{n_{kr}n}{n_k^{(1)}n_r^{(2)}},$$

sendo  $n_{kr}$  o número de observações sendo atribuídas simultaneamente ao  $k$ -ésimo e  $r$ -ésimo agrupamento sob os particionamentos  $c_1$  e  $c_2$ , respectivamente. Quando  $c_1 = c_2$ , temos que  $VI(c_1, c_2) = 0$ . O limite superior para  $VI$  é igual a  $\log(n)$ . A função de *MixSim* responsável pelo cálculo do  $VI$  é chamada de  $VarInf()$ .

## 2.7 Validação interna de agrupamentos

De acordo com [Kassambara \(2017\)](#), o termo validação de agrupamentos é usado para se referir ao procedimento de avaliação da qualidade dos resultados do algoritmo de agrupamento. Isso é importante para evitar padrões em dados aleatórios, como também na situação em que se deseja comparar dois agrupamentos algoritmos.

A validação interna de agrupamentos utiliza as informações do processo de agrupamento para avaliar a qualidade de uma estrutura de agrupamento sem referência às informações externas, isto é, sem saber o resultado previamente ao processo de agrupamento. Também pode ser usado para estimar o número de agrupamento e o algoritmo de agrupamento apropriado sem quaisquer dados externos.

As medidas de validação interna refletem frequentemente a compactação, a conexão e a separação das partições do agrupamento, medidas definidas a seguir:

- **Compactação ou coesão de agrupamentos:** Mede o quão perto estão os objetos dentro do mesmo *cluster*. Uma menor variação dentro do agrupamento é um indicador de uma boa compactação. Os diferentes índices para se avaliar o compactação dos agrupamentos é baseada em medidas de distâncias;
- **Separação:** Mede o quão bem separado um agrupamento está de outros agrupamentos. Os índices usados como medidas de separação incluem:
  - distâncias entre os centros do agrupamento;
  - as distâncias mínimas de pares entre objetos em diferentes agrupamentos.
- **Conectividade:** Corresponde até que ponto os itens são colocados no mesmo agrupamento como seus vizinhos mais próximos no espaço de dados. A conectividade pode apresentar valores entre zero e infinito, e deve ser minimizado.

### 2.7.1 Coeficiente de silhueta

A análise de silhueta mede o quão bem uma observação está agrupada e estima a distância média entre os agrupamentos. O gráfico de silhueta exibe uma medida de quão próximo cada ponto em um agrupamento está de pontos nos agrupamentos vizinhos ([Kassambara 2017](#)).

Para cada observação  $i$ , calcula-se largura da silhueta  $s_i$  como apresentado abaixo:

- (i) Calcula-se a dissimilaridade média  $a_i$  entre a observação  $i$  e todos os outros pontos pertencentes ao agrupamento que a observação  $i$  pertence;
- (ii) Para todos os outros agrupamentos  $C$  que a observação  $i$  não pertença, calcula-se a distancia (dissimilaridade) entre  $i$  e todas as observações de  $C$ ,  $d(i, C)$ . O menor valor de  $d(i, C)$ , definido como  $b_i = \min_C d(i, C)$  pode ser considerado a dissimilaridade entre a observação  $i$  e o agrupamento vizinho;
- (iii) A largura da silhueta da observação  $i$  é definida como:

$$S_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}.$$

Quanto mais próximo de 1 é o valor de  $S_i$ , melhor avaliado é o agrupamento dessa observação. Por outro lado, quanto mais próximo de  $-1$ , maior a probabilidade de que essa observação tenha sido agrupada incorretamente.

## 2.7.2 Índice de Dunn

[Kassambara \(2017\)](#) descreve o índice de Dunn como outra medida de validação interna de agrupamentos, que pode ser calculado da seguinte forma:

- (i) Para cada cluster, calcule a distância entre cada um dos objetos no agrupamento e os objetos nos outros agrupamento;
- (ii) Use o mínimo desta distância de pares como a separação entre agrupamentos (min.separação);
- (iii) Para cada agrupamento, calcule a distância entre os objetos no mesmo agrupamento;
- (iv) Use a distância intracluster máxima (ou seja, diâmetro máximo) como o intracluster compacidade;
- (v) Calcule o índice de Dunn da seguinte forma:

$$D = \frac{\text{min.separação}}{\text{max.diâmetro}}.$$

O índice de *Dunn* é maximizado quando os agrupamentos realizados no conjunto de dados estão bem compactados e separados, dessa forma o diâmetro deverá apresentar um

valor pequeno e a distância entre os agrupamentos deverá apresentar valores grandes.

Os métodos de clusterização e os índices de avaliação, descritos nesse capítulo, foram usados nos estudos envolvendo simulações de conjuntos de dados com o intuito de se criar contextos que evidencie especificidades e deficiências dos métodos em estudo. Os resultados e discussões estão presentes no [Capítulo 3](#).

## Capítulo 3

# Comparações em cenários controlados

Este capítulo tem como objetivo simular dados de agrupamentos utilizando misturas gaussianas a fim de comparar os resultados dos métodos de agrupamento *K-Means*, *DBSCAN*, *Gaussian Mixture Model*, *Fuzzy C-Means* e *Spectral*.

Para realizar a simulação, foram variados os parâmetros de sobreposição de pares, número de agrupamentos, número de dimensões e número de observações, visando abranger diferentes cenários. Em seguida, os métodos de clusterização mencionados foram aplicados aos dados simulados.

As misturas gaussianas são modelos probabilísticos, amplamente, utilizados em clusterização e outras tarefas de análise de dados. Elas assumem que os dados são gerados a partir de uma combinação de distribuições gaussianas e o parâmetro de sobreposição de pares controla o grau de sobreposição entre as distribuições gaussianas.

O número de agrupamentos é outro parâmetro importante em clusterização, que deve ser escolhido com cuidado. Um número muito grande de agrupamentos pode levar a agrupamentos com poucos objetos, enquanto um número muito pequeno de agrupamentos pode levar a agrupamentos muito amplos e pouco informativos. O número de dimensões e o número de observações também podem afetar significativamente a qualidade dos agrupamentos obtidos.

Para avaliar a qualidade dos agrupamentos obtidos, foram utilizadas medidas como índice de Rand ajustado, índice de Rand, coeficiente de silhueta, variação da informação, índice de Dunn e proporção de classificações corretas. Essas medidas permitem avaliar a similaridade entre os agrupamentos obtidos e os agrupamentos esperados, bem como a

qualidade intrínseca dos agrupamentos.

Antes de apresentarmos os resultados das simulações e aplicações dos métodos de clusterização, é importante mencionar a máquina utilizada para executar os experimentos. Todas as simulações e aplicações foram executadas em uma máquina equipada com um processador Intel® Core™ i7-7700, com 4 núcleos físicos e 8 *threads*, rodando em 4.0 GHz. Além disso, a máquina conta com 16 GB de memória RAM DDR4 com *clock* de 2400 MHz e uma placa de vídeo NVIDIA GeForce 1060 com 6GB de memória GDDR6. Essa configuração garante um bom desempenho em tarefas que exigem alto poder de processamento e memória, permitindo que as simulações e aplicações fossem executadas de forma rápida e eficiente.

Todos os códigos utilizados na geração de resultados nesse estudo estão disponíveis em <https://github.com/BadaroMath/Cluster-Comparison>.

### 3.1 Variação dos níveis de sobreposição de pares

Nesta seção, investigamos o impacto do grau de sobreposição entre as distribuições gaussianas que geraram os dados nas misturas gaussianas simuladas. O parâmetro de sobreposição de pares  $\bar{\omega}$ , explicado anteriormente na seção [Subseção 2.6.1](#), desempenha um papel crucial nesse processo, já que influencia a separação dos agrupamentos e, consequentemente, afeta o desempenho dos métodos de clusterização.

Avaliar o desempenho dos métodos em diferentes níveis de sobreposição é importante, pois, em algumas situações, a sobreposição pode ser maior, dificultando a separação dos grupos. Isso pode acontecer, por exemplo, em dados de imagens em que um mesmo objeto pode aparecer em diferentes posições. Em outras situações, a sobreposição pode ser menor, facilitando a separação dos grupos.

Cada método de clusterização apresenta particularidades técnicas e pode apresentar um desempenho diferente dependendo do nível de sobreposição dos dados. O *K-Means* pode não ser adequado para dados com formas não esféricas e tamanhos de agrupamentos distintos ([Jain 2010](#) e [Steinley 2006](#)), enquanto o *DBSCAN* é eficaz em dados com agrupamentos de formatos diferentes e com a presença de ruídos ([Jain et al. 2020](#)). O *Gaussian Mixture Model* é mais indicado para dados com distribuições complexas e pode ser mais robusto em relação ao ruído ([Bellatreche e Chakravarthy 2017](#)). O *Fuzzy C-Means*, por sua vez, permite a atribuição de uma instância a mais de um agrupamento,



sendo adequado para dados com incerteza na classificação. Por fim, o método *Spectral* é especialmente adequado para dados com estruturas de alta dimensionalidade e pode ser mais eficiente na separação de agrupamentos com formas não convencionais (Wang *et al.* 2015). Essas particularidades precisam ser consideradas ao escolher o método mais adequado para o problema em questão.

### 3.1.1 Índice de Rand Ajustado

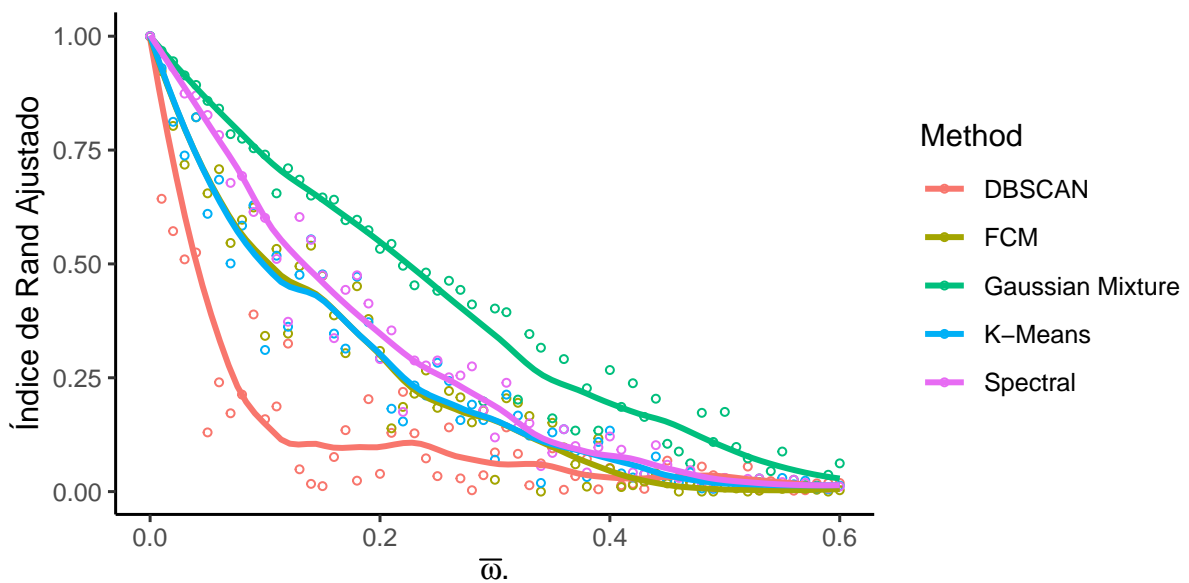


Figura 3.1: Índice de Rand Ajustado de cada método utilizado sobre diferentes níveis de sobreposição de pares.

Os resultados obtidos a partir da Figura 3.1 sugerem que o *Gaussian Mixture Model* (GM) é o método mais eficaz dentre os estudados para lidar com diferentes níveis de sobreposição de pares, particularmente, para agrupamentos gerados a partir de misturas gaussianas. Essa observação é consistente com a natureza probabilística do GMM, que é capaz de modelar misturas gaussianas com diferentes graus de sobreposição.

Por outro lado, o *DBSCAN*, que é um método que utiliza uma estratégia baseada em densidade para identificar agrupamentos, apresentou o pior desempenho. Isso ocorre porque o *DBSCAN* é menos adequado para lidar com agrupamentos de formas irregulares e densidades variáveis, o que é comum em cenários com níveis elevados de sobreposição.

*K-Means* e *Fuzzy C-Means* apresentaram resultados semelhantes, o que é esperado pela semelhança entre os métodos. Ambos são baseados em centroides e procuram encontrar

uma partição dos dados em um número pré-definido de agrupamentos. Além disso, ambos utilizam a distância Euclidiana entre os pontos e os centroides dos agrupamentos para calcular o grau de pertinência dos pontos a cada agrupamentos. Enquanto o *K-Means* utiliza uma partição rígida, ou seja, cada ponto é associado a apenas um agrupamentos, o *Fuzzy C-Means* utiliza uma partição suave, em que cada ponto pode ter graus variados de pertinência a cada agrupamentos.

### 3.1.2 Proporção de Classificações Corretas

A Proporção de Classificações Corretas é uma medida importante para avaliar os agrupamentos, pois ela mede diretamente a proporção de pontos que foram classificados corretamente em relação ao número total de pontos. Isso permite uma avaliação simples e intuitiva da qualidade da partição gerada pelo algoritmo de clusterização.

Como explicado na [Subseção 2.6.4](#), a Proporção de Classificações Corretas (*Class-Prop()* do pacote *MixSim*) é calculada comparando as classes verdadeiras dos pontos com as classes atribuídas pelo algoritmo de clusterização. Isso significa que, para calcular a medida, é necessário ter informações sobre as classes verdadeiras dos pontos, o que pode ser obtido em situações cujos dados são gerados artificialmente ou quando é possível realizar uma rotulação manual dos dados.

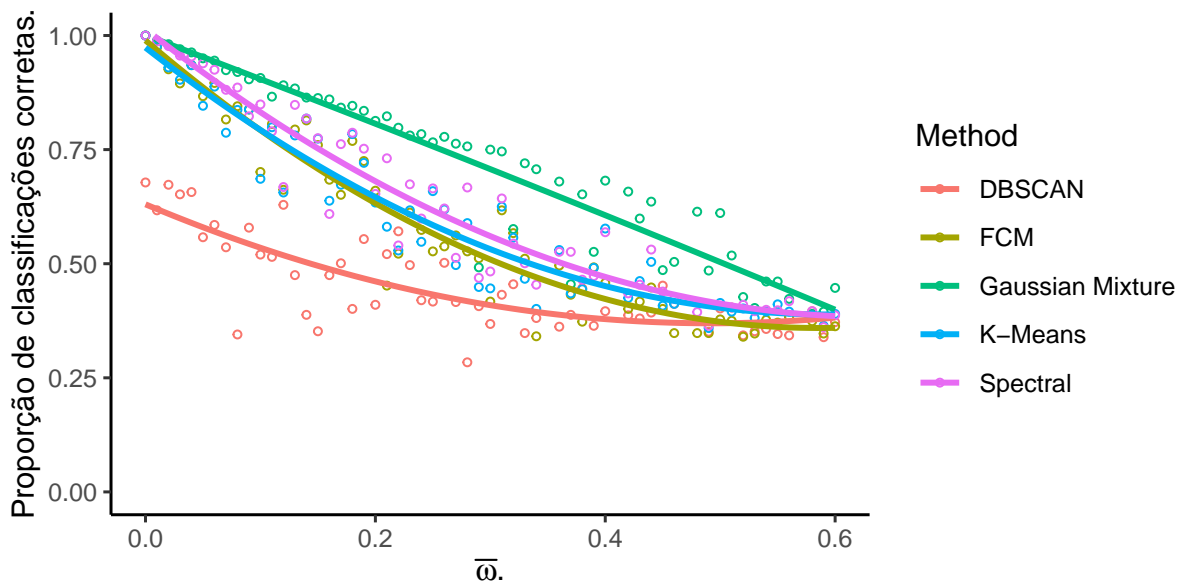


Figura 3.2: Proporção de classificações corretas de cada método utilizado sobre diferentes níveis de sobreposição de pares.

Na [Figura 3.2](#) é notado algo semelhante em relação ao desempenho dos métodos de acordo com o Índice de Rand Ajustado, mostrado na [Figura 3.1](#), porém com uma diferença importante: o *DBSCAN*, em níveis de sobreposições menores (como 0 por exemplo), também apresenta um resultado ruim. Isso ocorre porque, dentre os métodos analisados, o *DBSCAN* é o único que não pré-determina o número de agrupamentos, podendo haver pontos que não se enquadram em nenhum agrupamentos ou que são classificados como ruído pelo algoritmo.

A medida Proporção de Classificações Corretas mede a proporção de classificações corretas em relação ao número total de pontos, enquanto o Índice de Rand Ajustado (ARI) mede a similaridade entre duas partições, levando em conta a possibilidade de coincidência aleatória ([Melnykov et al. 2012](#)).

Em geral, a medida *ClassProp* pode apresentar resultados mais baixos do que o ARI, pois a primeira não leva em conta a possibilidade de coincidência aleatória na atribuição dos pontos aos agrupamentos. Por outro lado, o ARI é uma medida mais robusta que leva em conta a possibilidade de coincidência aleatória na atribuição dos pontos aos agrupamentos, sendo assim o ARI mais adequado para avaliar a qualidade de uma partição em relação aos agrupamentos reais dos dados quando temos muitas classes ([Melnykov et al. 2012](#)).

Assim, em situações em que há mais classes do que o real, é possível que a medida *ClassProp* apresente resultados mais baixos do que o ARI. Por isso, é importante utilizar uma variedade de medidas de avaliação para obter uma visão mais completa da qualidade das partições geradas pelos algoritmos de agrupamentos.

### 3.1.3 Variação de Informação

A medida de Variação de Informação é uma medida importante para avaliar a qualidade de uma partição gerada por um algoritmo de clusterização, pois ela leva em consideração a informação mútua entre as classes verdadeiras e as classes atribuídas pelo algoritmo.

De acordo com [Meilă e Shi \(2000\)](#), a VI se concentra na relação entre um ponto e seu *cluster* em cada agrupamento, em vez de contar diretamente as relações entre pares de pontos. Isso significa que o VI tem uma abordagem diferente em relação a outros critérios de comparação, mas não é melhor ou pior do que eles.

A seguir, temos uma avaliação da variação da informação para cada método em estudo

sobre diferentes níveis de sobreposição de pares.

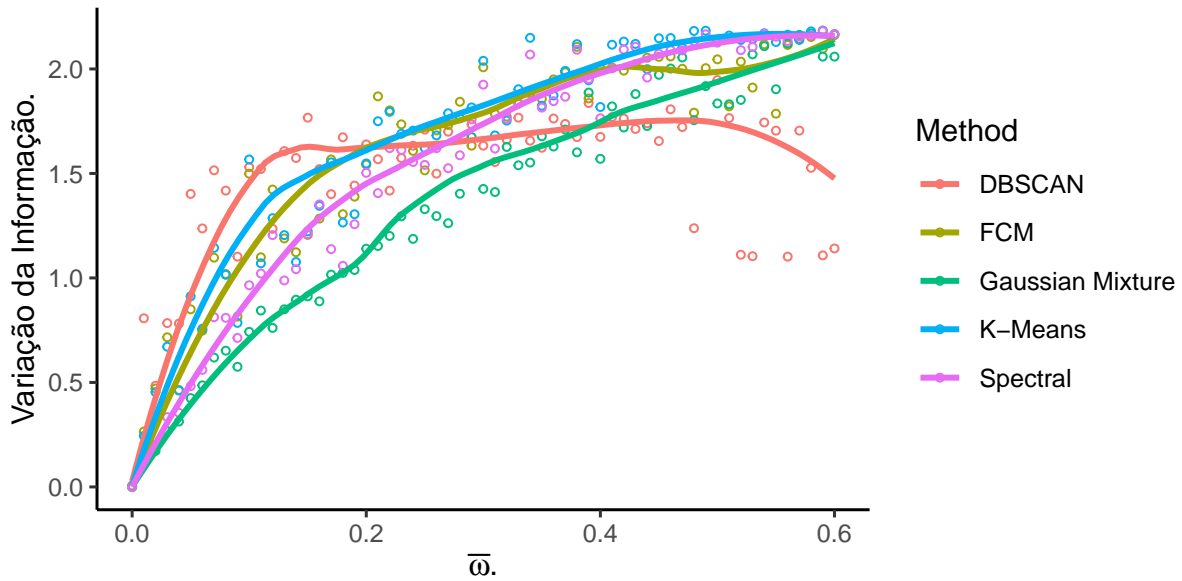


Figura 3.3: Variação da informação de cada método utilizado sobre diferentes níveis de sobreposição de pares.

Na [Figura 3.9](#) notamos que a VI se comporta de maneira inversa ao Índice de Rand Ajustado. Os valores aumentam a medida que os níveis de sobreposição de pares aumenta.

a Variação de Informação não depende diretamente do número de pontos de dados no conjunto. Isso proporciona uma base muito mais forte para comparações entre conjuntos de dados, algo que precisamos fazer se quisermos comparar algoritmos de agrupamento entre si. Também pode abrir caminho para comparar os resultados de “amostras finitas” do agrupamento padrão com os “valores populacionais” baseados em dados teoricamente infinitos.

Por essas razões, a medida de Variação de Informação é uma medida importante para avaliar a qualidade de uma partição gerada por um algoritmo de clusterização, especialmente, em situações de sobreposição entre os agrupamentos. No entanto, é importante lembrar que nenhuma medida isoladamente pode fornecer uma avaliação completa e robusta da qualidade dos agrupamentos, sendo necessário considerar várias medida em conjunto para obter uma avaliação mais completa.

Outra diferença importante é que o ARI retorna um valor entre  $-1$  e  $1$ , em que valores mais próximos de  $1$  indicam uma partição de alta qualidade, enquanto a medida de Variação de Informação retorna um valor entre  $0$  e infinito, na qual valores mais próximos de  $0$  indicam uma partição de alta qualidade ([Subseção 2.6.4](#)).

### 3.1.4 Coeficiente de Silhueta

O coeficiente de silhueta é uma medida que avalia a qualidade dos agrupamentos produzidos por diferentes métodos, levando em conta tanto a separação entre os agrupamentos quanto a coerência interna de cada agrupamento. Ele é útil para selecionar o número ótimo de agrupamentos e comparar o desempenho de diferentes métodos de agrupamento.

No contexto em que os dados são simulados, podemos utilizar o coeficiente de silhueta para avaliar sua eficiência frente a outras medidas, como o Índice de Rand Ajustado. Considerando que o coeficiente de silhueta é uma medida interna que não requer os agrupamentos verdadeiros para avaliar o desempenho do método, é importante entendermos suas limitações antes de confiar-se nos resultados baseando-se nessa medida.

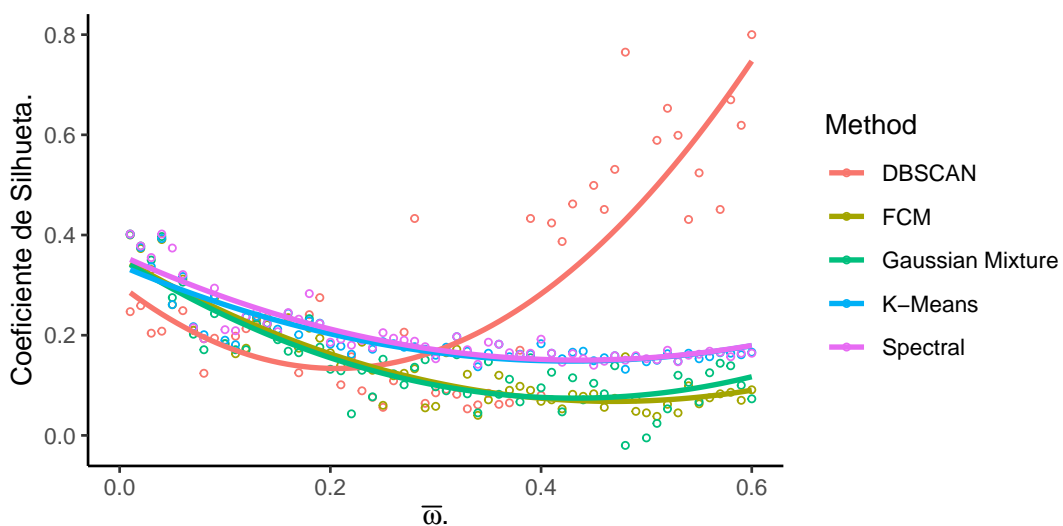


Figura 3.4: Coeficiente de Silhueta de cada método utilizado sobre diferentes níveis de sobreposição de pares.

Como visto na [Figura 3.4](#), o desempenho do *DBSCAN* parece aumentar conforme os níveis de sobreposição de pares vão aumentando, contrariando os resultados anteriores. Isso ocorre porque, em algumas situações, o *DBSCAN* pode formar agrupamentos que se estendem por várias regiões com densidades diferentes, tornando difícil o cálculo do coeficiente de silhueta.

Além disso, o coeficiente de silhueta pode ser menos eficiente para avaliar métodos de agrupamento baseados em densidade, pois uma maior ênfase pode ser dada à separação dos clusters do que à densidade de cada agrupamento.

### 3.1.5 Índice de *Dunn*

O índice de *Dunn* é uma medida que avalia a qualidade dos agrupamentos baseada na distância entre agrupamentos e na densidade de cada agrupamento. Ele é calculado como a razão entre a menor distância intercluster e a maior distância intracluster. Quanto maior o índice de *Dunn*, melhor a qualidade do agrupamento, indicando que a distância entre agrupamentos é grande em relação à densidade de cada agrupamento.

Ao contrário do coeficiente de silhueta, que avalia a qualidade dos agrupamentos com base na separação entre clusters e na coesão dentro de cada agrupamento, o índice de *Dunn* leva em consideração a relação entre todos os agrupamentos. Dessa forma, ele é capaz de avaliar a qualidade dos agrupamentos em situações em que existem agrupamentos com densidades muito diferentes.

Além disso, o índice de *Dunn* também permite comparar diretamente a qualidade de diferentes agrupamentos, ao contrário do coeficiente de silhueta, que só permite avaliar a qualidade de um agrupamento de cada vez.

Portanto, o índice de *Dunn* é uma medida importante para comparar a qualidade de diferentes agrupamentos, especialmente em situações em que existem agrupamentos com densidades muito diferentes.

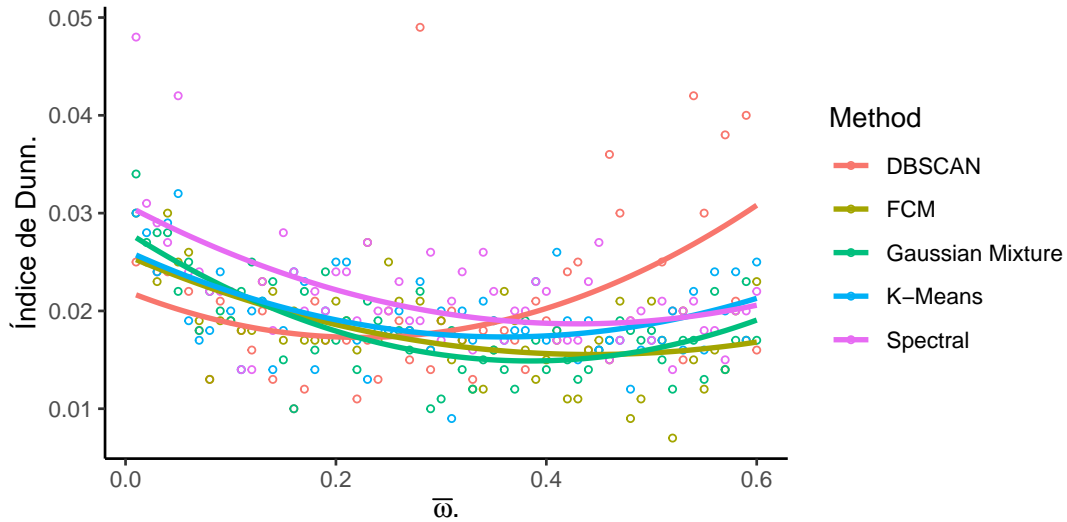


Figura 3.5: Índice de *Dunn* de cada método utilizado sobre diferentes níveis de sobreposição de pares.

Porém, como visto na [Figura 3.5](#), mais uma vez temos os resultados do *DBSCAN* superestimado. O índice de *Dunn* pode superestimar a qualidade dos resultados do *DBSCAN* devido à sua natureza de avaliação baseada na densidade dos clusters. O *DBSCAN*

é um algoritmo que se baseia na densidade dos pontos para identificar clusters, o que pode levar à criação de clusters com alta densidade de pontos, mas que podem estar muito distantes uns dos outros. Nesses casos, o índice de *Dunn* pode indicar que a qualidade dos clusters é alta, já que há uma grande distância entre eles, mas na verdade esses clusters podem não ser úteis para a análise dos dados.

### 3.1.6 Tempo de Execução dos métodos

Em termos de tempo de execução, é esperado que o *DBSCAN* tenha um tempo de execução menor em comparação com os demais métodos, pois sua complexidade computacional é linear com relação ao número de pontos de dados e a quantidade de agrupamentos formados.

Tanto o *Fuzzy C-Means* quanto o *Gaussian Mixture Model* possuem complexidades computacionais mais elevadas, pois envolvem operações matemáticas mais complexas e podem levar mais tempo para convergir. No entanto, o tempo de execução do *Fuzzy C-Means* é, geralmente, menor do que o do *Gaussian Mixture Model*, pois este último envolve a estimativa de parâmetros adicionais.

Por outro lado, o K-Means também possui uma complexidade computacional relativamente baixa, mas o tempo de execução pode ser afetado pelo número de clusters e pela dimensionalidade dos dados. O método *Spectral* pode ter um tempo de execução mais elevado em comparação com os outros métodos, pois envolve a decomposição de uma matriz de afinidade, o que pode ser computacionalmente custoso, especialmente para grandes conjuntos de dados.

A [Figura 3.6](#) ilustra as diferenças nos tempos de execução dos métodos de agrupamento mencionados, variando o número de observações e dimensões, e com diferentes valores de sobreposição de pares. É possível observar que o *Spectral* é o método mais lento entre todos, com tempos de execução superiores a 15 segundos em alguns casos. Para analisar os outros métodos com mais detalhes, removemos os pontos associados ao método *Spectral* da [Figura 3.6](#), resultando na [Figura 3.7](#).

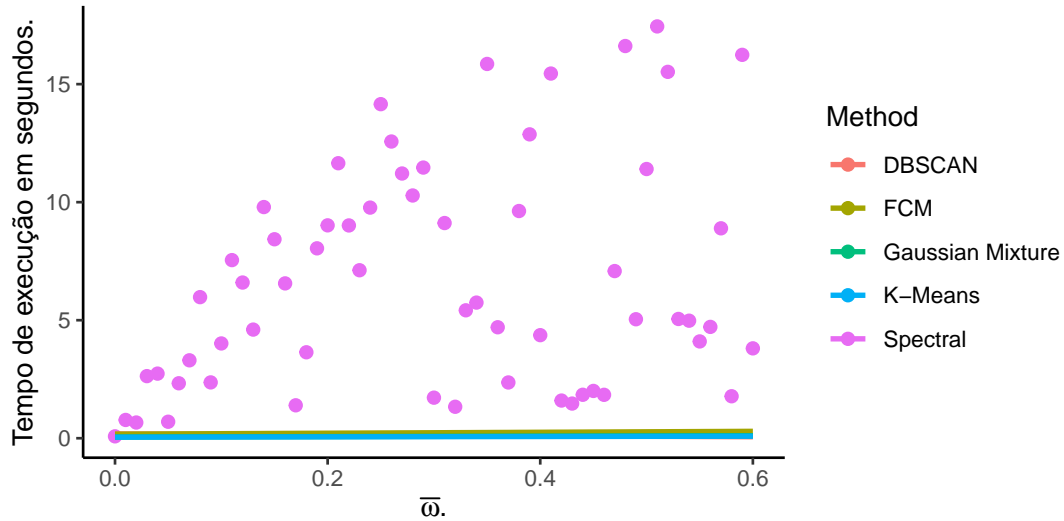


Figura 3.6: Tempo de execução em segundos de cada método utilizado sobre diferentes níveis de sobreposição de pares.

Na [Figura 3.7](#), observamos que o *K-Means* apresentou o melhor desempenho em termos de tempo de execução na maioria dos agrupamentos simulados. Em seguida, o *Gaussian Mixture Model* obteve resultados satisfatórios, seguido pelo *DBSCAN*. Por outro lado, o método *Fuzzy C-Means* apresentou o segundo pior resultado em termos de tempo de execução, indicando uma maior exigência computacional em comparação aos demais métodos avaliados.

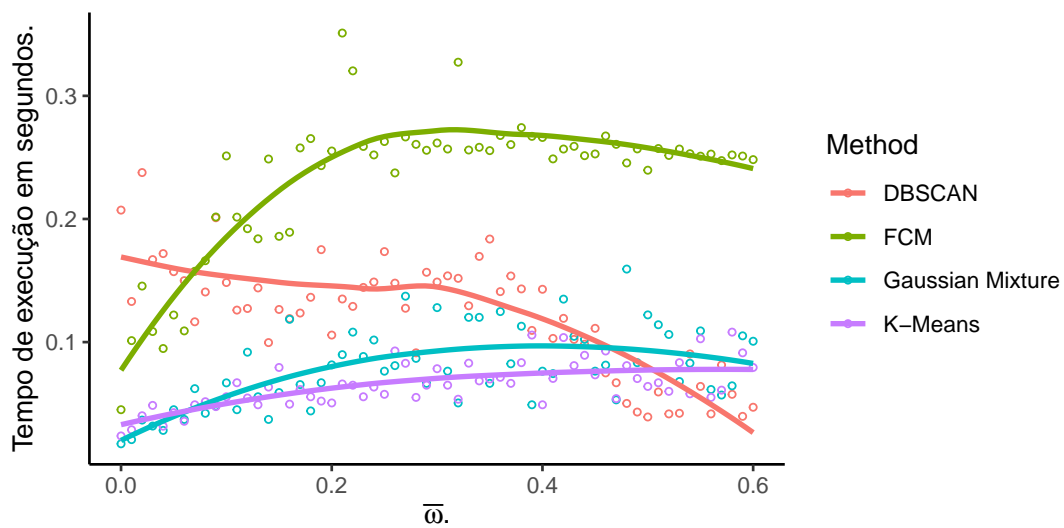


Figura 3.7: Tempo de execução em segundos de cada método utilizado, exceto pelo *Spectral*, sobre diferentes níveis de sobreposição de pares.



## 3.2 Variação do número de agrupamentos

Nesta seção, avaliamos o desempenho dos métodos de agrupamento variando o número de agrupamentos de 2 até 40, simulando conjuntos de dados com agrupamentos sem sobreposição de pares e utilizando 3 dimensões com 5 mil observações cada.

Avaliar os métodos de agrupamento em diferentes números de agrupamentos é importante porque a capacidade de detectá-los pode variar dependendo do número de clusters presentes nos dados. Alguns métodos, como o *K-Means* e o *Fuzzy C-Means*, funcionam melhor quando o número de clusters é definido a priori. Já outros métodos, como o *DBSCAN* e o *Spectral*, podem detectar o número de clusters automaticamente, mas podem ter dificuldade em encontrar clusters quando o número deles é muito grande.

Na [Seção 3.1](#), foi constatado que as medida de validação interna, como o coeficiente de silhueta e o índice de *Dunn*, não são adequadas para avaliar corretamente métodos que utilizam algoritmos baseados na densidade dos pontos, como o *DBSCAN*. Portanto, nas comparações seguintes, para analisar o desempenho dos diferentes métodos em estudo, nos concentramos nas comparações utilizando o Índice de Rand Ajustado, Índice de *Dunn* e o tempo de execução.

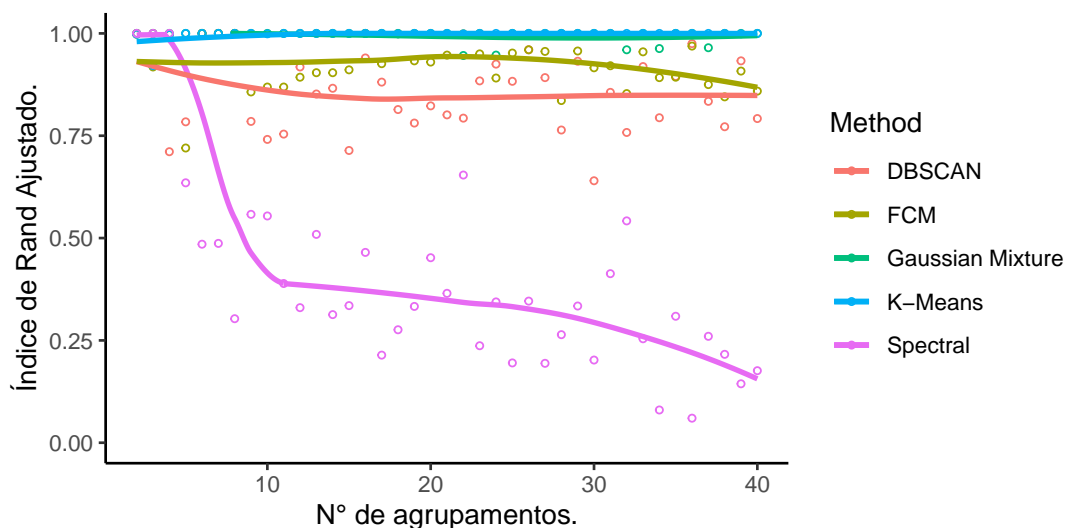


Figura 3.8: Índice de Rand de cada método utilizado sobre diferentes quantidades de agrupamentos.

Na [Figura 3.8](#), é evidente que o desempenho do método *Spectral* diminui à medida que o número de agrupamentos aumenta nos conjuntos de dados. No entanto, o desempenho do método é perfeito com até três agrupamentos. O método *DBSCAN* é o segundo pior, mantendo um desempenho constante e razoável. Por outro lado, o *Fuzzy C-Means*,

*Gaussian Mixture Model* e *K-Means* apresentaram um bom desempenho em todas as configurações testadas neste cenário.

O agrupamento *Spectral* é uma técnica poderosa, porém requer a resolução do problema de autovalores da matriz Laplaciana convertida a partir da matriz de similaridade correspondente ao conjunto de dados fornecido. Além disso, o método pode ter dificuldades em identificar agrupamentos muito pequenos ou muito próximos uns dos outros, especialmente em conjuntos de dados com alta dimensionalidade, devido ao fenômeno conhecido como “maldição da dimensionalidade” ([Bellman 1961](#)). Por isso, o método pode não desempenhar bem com um grande número de agrupamentos.

É importante considerar o tamanho do conjunto de dados e a complexidade dos cálculos necessários para a aplicação de cada método de agrupamento, uma vez que alguns métodos podem se tornar impraticáveis ou pouco eficientes em determinados cenários.

A seguir, temos uma representação do desempenho de cada método de acordo com o Índice de *Dunn*, variando o número de agrupamentos de 2 até 40.

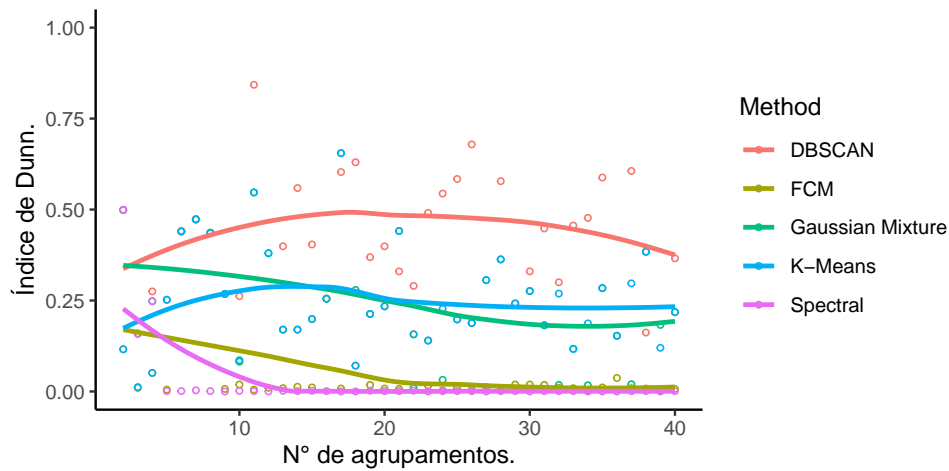


Figura 3.9: Índice de *Dunn* para cada método utilizado sobre diferentes quantidades de agrupamentos.

O Índice de *Dunn* se mantém superior para o método *DSBCAN* em todas as configurações de números de agrupamentos testadas, enquanto o *Spectral* se mantém abaixo dos demais métodos. Isso sugere que o foi eficaz em identificar o *Spectral* como sendo o pior método quando o número de agrupamentos aumentam, mas não é eficaz quando se trata do *DSBCAN*, superestimando seu desempenho.

Abaixo, temos uma ilustração do tempo de execução de cada método em estudo ao variar o número de agrupamentos.

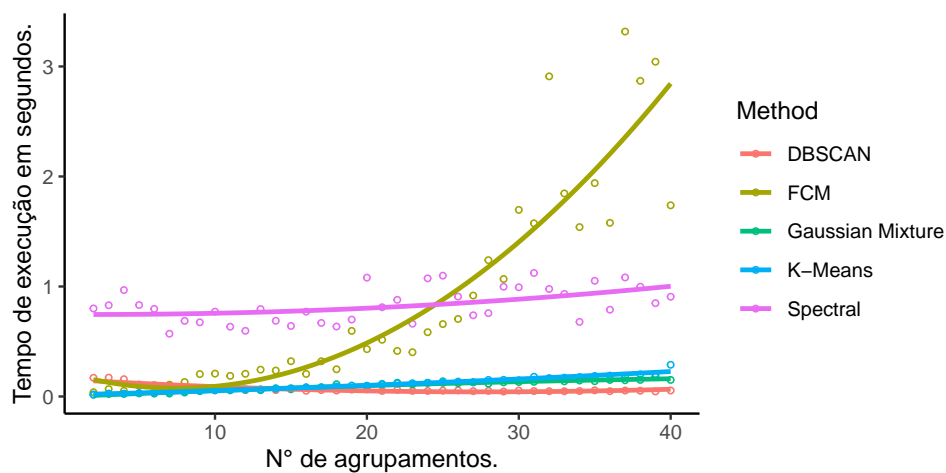


Figura 3.10: Tempo de execução em segundos para cada método utilizado sobre diferentes quantidades de agrupamentos.

Como já mencionado na [Subseção 3.1.6](#), é natural que o *Fuzzy C-Means* apresente

um tempo de execução superior quando há mais agrupamentos, pela complexidade matemática e computacional do método, que acaba demorando mais para convergir. Aqui temos o *Spectral* estável em relação ao tempo, mas ainda com tempo superior ao restante dos métodos.

### 3.3 Variação do número de variáveis

Ao aplicar a clusterização em um conjunto de dados, é importante considerar o número de variáveis que compõem esse conjunto. O número de variáveis pode afetar significativamente os resultados da clusterização, tanto em termos de eficácia quanto de eficiência.

Por exemplo, quando o número de variáveis é muito grande, alguns métodos de clusterização podem apresentar problemas de desempenho e escalabilidade. Além disso, um grande número de variáveis pode aumentar a complexidade do espaço de busca, tornando a tarefa de encontrar os clusters mais difícil (Masulli *et al.* 2015).

Por outro lado, um número muito pequeno de variáveis pode resultar em uma perda significativa de informações, o que pode levar a uma clusterização menos precisa e menos representativa dos dados. Para investigar o impacto do número de variáveis na clusterização, simulamos agrupamentos variando de 2 até 200 variáveis, mantendo o nível de sobreposição de pares médio igual a 0, em um conjunto de dados com 5 mil observações e 3 agrupamentos.

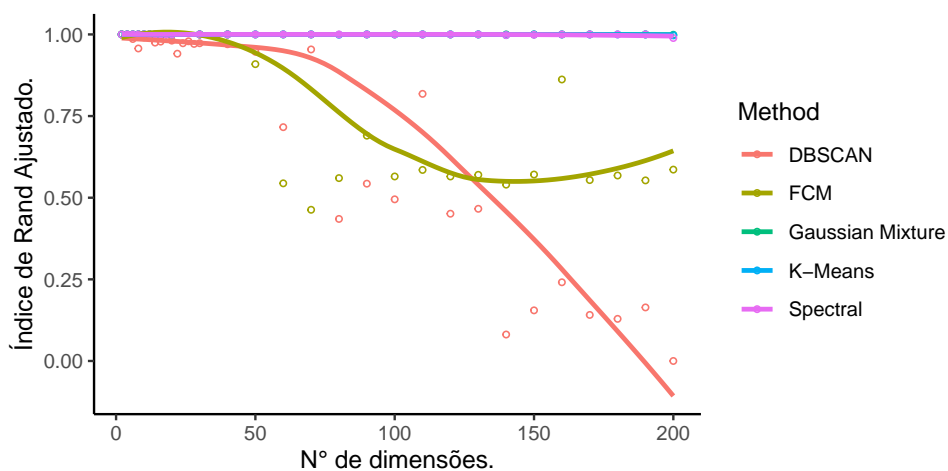


Figura 3.11: Índice de Rand Ajustado de cada método utilizado sobre diferentes quantidades de dimensões.

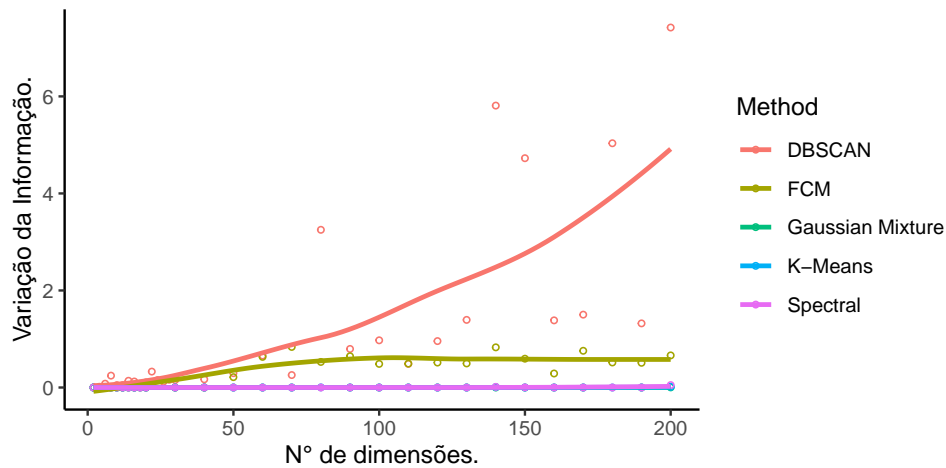


Figura 3.12: Variação da Informação de cada método utilizado sobre diferentes quantidades de dimensões.

Pelas [Figura 3.11](#) e [Figura 3.12](#), os resultados mostraram que o desempenho e a eficácia da clusterização variaram consideravelmente em relação ao número de variáveis para alguns métodos. Por exemplo, o método *DBSCAN* mostrou-se mais sensível ao número de variáveis do que outros métodos, assim como o *Fuzzy C-Means*, com o Índice de Rand Ajustado caindo consideravelmente a partir de 50 dimensões. Enquanto isso, os outros métodos desempenham muito bem até 200 dimensões.

Devido à natureza fuzzy, o *Fuzzy C-Means* é mais sensível ao número de variáveis do que outros métodos de clusterização, como *K-Means*, *Spectral* e *Gaussian Mixture Model*. Isso ocorre porque, à medida que o número de variáveis aumenta, o espaço de busca se torna mais complexo, tornando mais difícil encontrar uma solução que maximize a função de custo do *FCM*. Além disso, o *FCM* pode produzir clusters que se sobrepõem, o que pode levar a uma interpretação menos clara dos resultados. Em outras palavras, é possível que um objeto tenha pertinência significativa em mais de um agrupamento, o que pode ser difícil de interpretar ([Winkler et al. 2010](#)).

Já para o *DBSCAN*, uma das principais razões pelas quais ele é mais sensível ao número de dimensões do que outros algoritmos de clusterização é que, em dimensões mais altas, a densidade dos dados tende a se diluir muito mais rapidamente. Isso significa que, em um espaço de alta dimensionalidade, o *DBSCAN* pode não ser capaz de distinguir grupos densos de objetos de grupos esparsos. Por exemplo, quando há muitas dimensões, dois objetos que estão próximos em algumas dimensões podem estar muito distantes em outras, o que pode dificultar a detecção de agrupamentos significativos. Isso pode levar a agrupamentos impróprios ou mesmo à não identificação de agrupamentos em dados com

muitas dimensões (Boonchoo *et al.* 2018).

Além disso, em dimensões mais altas, o chamado "problema da maldição da dimensionalidade" (Clayton e Fortuna 2019) se torna mais proeminente. Esse problema se refere ao fato de que, à medida que a dimensionalidade dos dados aumenta, a quantidade de dados necessários para representar uma densidade de objetos de maneira confiável cresce exponencialmente. Isso pode levar a problemas computacionais e tornar o *DBSCAN* impraticável em espaços de alta dimensionalidade.

Portanto, é importante considerar cuidadosamente o número de variáveis ao escolher um método de clusterização e avaliar seus resultados. Uma abordagem sistemática, como a utilizada neste estudo, pode ajudar a identificar qual método é mais adequado para um determinado conjunto de dados e número de variáveis.

A seguir, temos o tempo de execução em segundos de cada método ao variar o número de dimensões entre de 2 até 200.

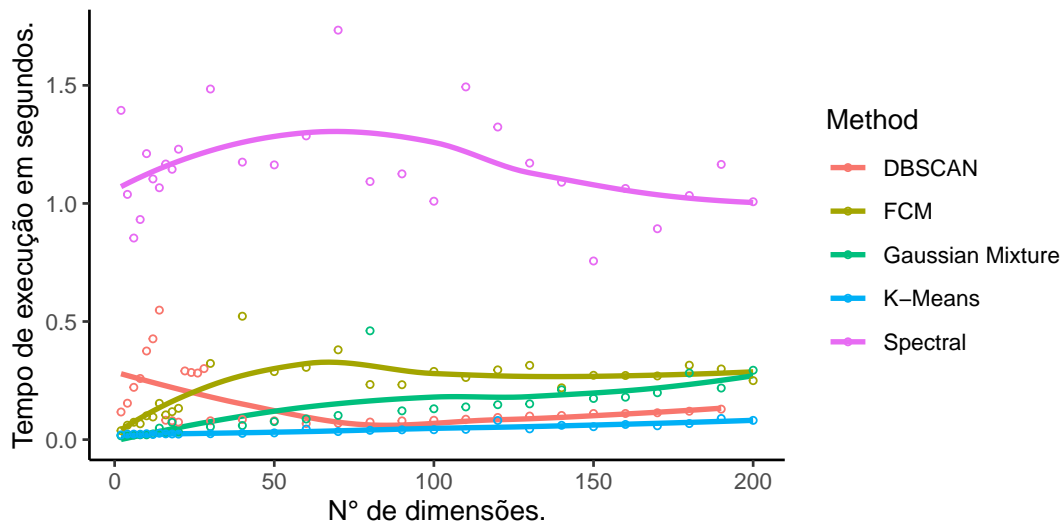


Figura 3.13: Tempo de execução em segundos de cada método utilizado sobre diferentes quantidades de dimensões.

Conforme ilustrado na Figura 3.13, no que diz respeito ao tempo de execução, os métodos avaliados não demonstraram sensibilidade significativa em relação à dimensionalidade. Observou-se constância de desempenho em todos os métodos, com o *Spectral* apresentando, em geral, o maior tempo.

### 3.4 Variação do número de observações

Dentre os diversos fatores que podem impactar a efetividade da clusterização, o número de observações é um dos mais relevantes. Para investigar essa relação, foram realizadas simulações variando o número de observações de 100 até 2 milhões, mantendo o nível de sobreposição de pares médio igual a 0, 5 dimensões e 3 clusters. Os métodos de clusterização avaliados incluem o *DBSCAN*, *K-Means*, *Fuzzy C-Means*, *Gaussian Mixture Model* e *Spectral*.

Cada um desses métodos possui características distintas que podem influenciar sua sensibilidade em relação ao número de observações. O *DBSCAN*, por exemplo, é conhecido por ser mais sensível à densidade de pontos, o que significa que sua efetividade pode diminuir quando há muitos pontos dispersos ou muito próximos. O *Spectral*, por outro lado, pode ser impactado pelo número de agrupamentos, o que pode levar a uma inadequada identificação dos grupos.

O *Fuzzy C-Means*, por sua vez, é mais suscetível à influência do número de dimensões, o que pode afetar sua capacidade de identificar corretamente os grupos. Já o *Gaussian Mixture Model* é um método probabilístico que assume a distribuição normal dos dados, podendo ser impactado por desvios significativos nessa distribuição.

A [Figura 3.14](#) mostra a quantidade máxima de observações em que foi possível simular com a configuração do computador utilizado, mencionada no início deste capítulo, considerando 5 dimensões, 3 agrupamentos e sem sobreposição de pares.

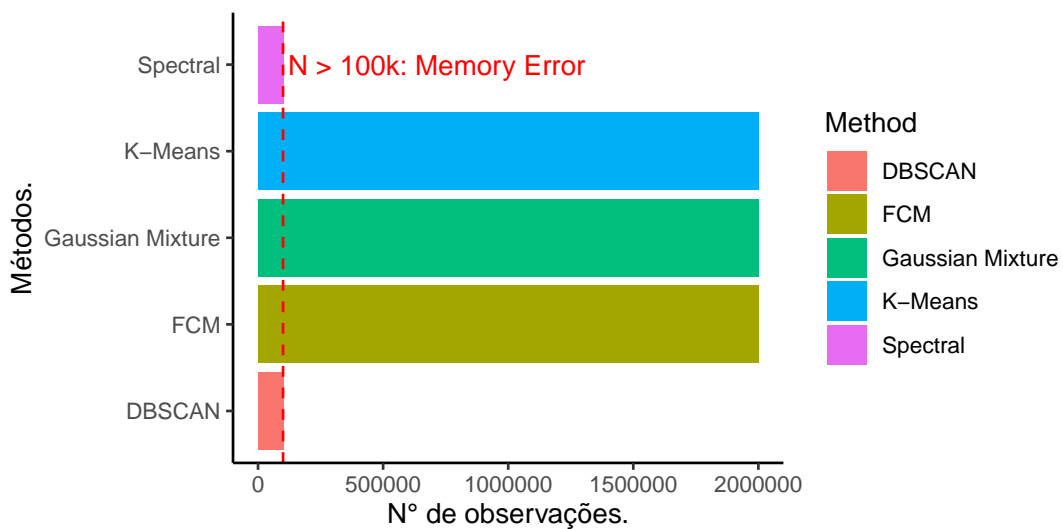


Figura 3.14: Quantidade máxima de observações em que cada método conseguiu processar durante as aplicações em dados simulados, considerando 3 agrupamentos e 5 dimensões, sem sobreposição de pares.

Tanto o *DBSCAN* quanto o *Spectral* são métodos de clusterização que podem enfrentar dificuldades ao utilizar conjuntos de dados com muitas observações devido à limitação de memória computacional. Nesse estudo, foi possível chegar a até 100 mil observações em 5 dimensões até começar a enfrentar problemas por falta de memória computacional.

No caso do *DBSCAN*, o método precisa calcular a distância entre cada par de pontos, o que pode ser computacionalmente custoso. Isso significa que, quando o conjunto de dados é muito grande, a quantidade de memória necessária para armazenar as informações necessárias pode ser maior do que a capacidade do computador. Isso pode levar a erros de memória ou a um desempenho ruim do algoritmo em termos de tempo de execução e capacidade de processamento.

Já no caso do *Spectral*, o método, que envolve o cálculo de uma matriz de afinidade, pode ser computacionalmente intensiva quando o conjunto de dados é grande. Essa matriz pode exigir uma grande quantidade de memória para ser armazenada, especialmente quando a dimensionalidade dos dados é alta. Isso pode levar a problemas de memória ou a um desempenho ruim do algoritmo, especialmente quando o conjunto de dados é muito grande.

A [Figura 3.15](#) mostra a variabilidade da medida Variação da informação para os métodos em estudo, principalmente, em relação aos números menores de observações.

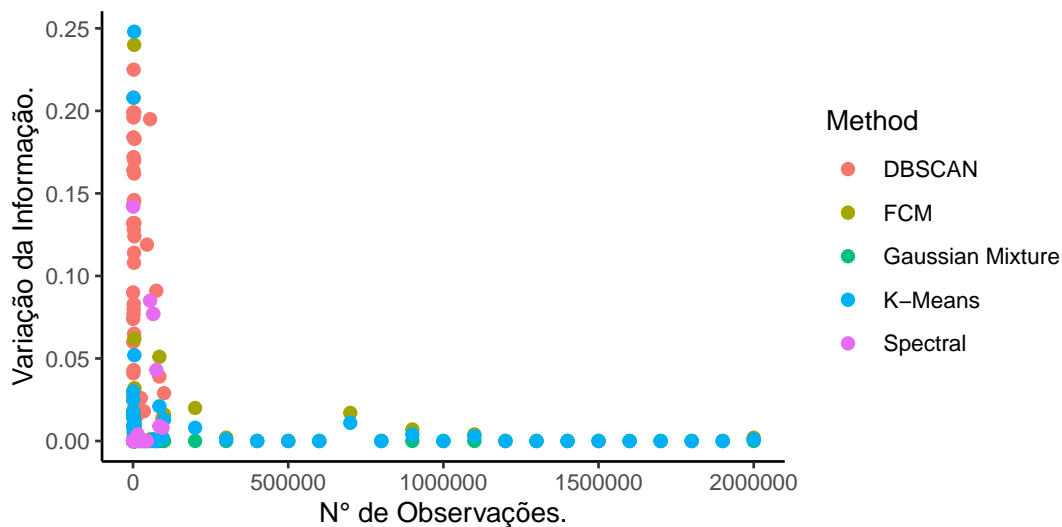


Figura 3.15: Tempo de execução em segundos de cada método utilizado sobre diferentes quantidades de observações.

Analisando a [Figura 3.15](#), os métodos *K-Means*, *Fuzzy C-Means* e *Gaussian Mixture Model* apresentaram um desempenho satisfatório para conjuntos de dados com até 2



milhões de observações, com apenas uma ligeira variação da informação em quantidades menores de pontos. A variação da informação tende a ser maior em quantidades menores de pontos, devido à maior probabilidade de formação de grupos com sobreposição ou de grupos muito próximos uns aos outros. No entanto, mesmo nesses casos, os métodos mencionados apresentaram resultados consistentes e robustos. Isso sugere que esses métodos podem ser utilizados para análise de grandes conjuntos de dados sem comprometer a qualidade dos resultados.

Em resumo, tanto o *DBSCAN* quanto o *Spectral* podem enfrentar problemas de memória computacional ao utilizar conjuntos de dados com muitas observações. Isso ocorre porque esses métodos exigem o armazenamento de uma grande quantidade de informações ou cálculos intensivos em termos de memória, o que pode ser difícil de gerenciar em computadores com capacidade limitada.

### 3.5 Variações combinadas de parâmetros

Semelhante ao que foi analisado na [Figura 3.1](#), abaixo temos uma avaliação dos métodos de agrupamento a partir do Índice de Rand Ajustado para níveis de sobreposição média de pares variando de 0 até 5%, porém agora com 50 dimensões, ao invés de 5.

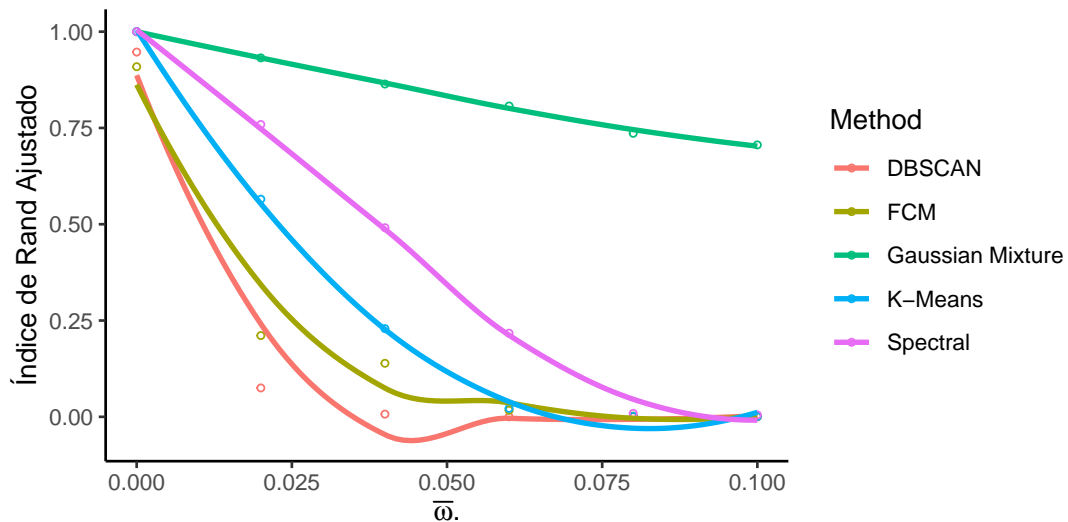


Figura 3.16: Índice de Rand Ajustado de cada método utilizado sobre diferentes níveis de sobreposição de pares, considerando 50 dimensões e 3 agrupamentos, com 5mil observações.

Observamos na [Figura 3.1](#) um comportamento muito semelhante ao da [Figura 3.1](#), com o Gaussian Mixture se destacando ainda mais dos demais em relação aos demais.

Isso pode indicar uma maior facilidade do método em lidar com sobreposição de pares quando há muitas variáveis.

A seguir temos um caso semelhante, porém agora com um alto número de agrupamentos, ao invés de dimensões.

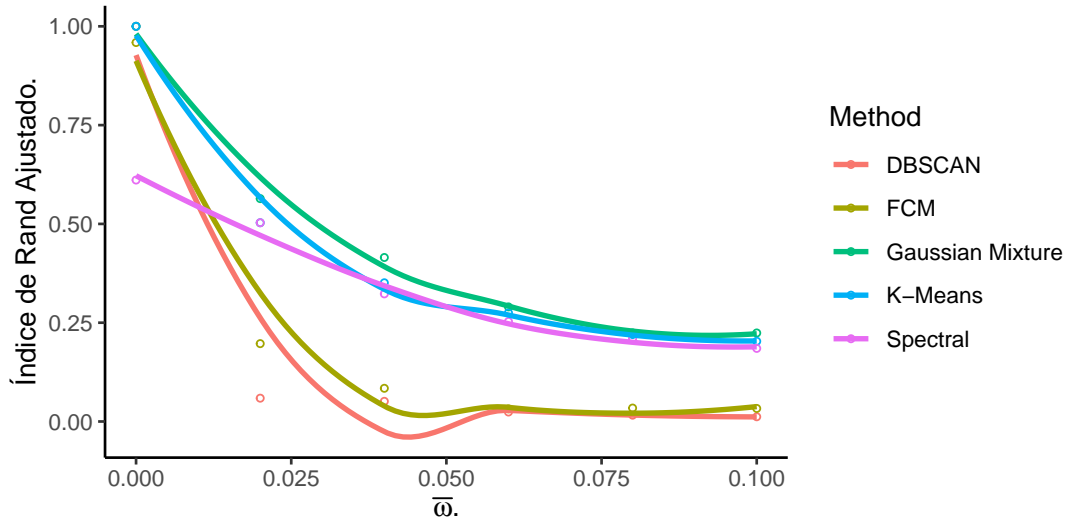


Figura 3.17: Índice de Rand Ajustado de cada método utilizado sobre diferentes níveis de sobreposição de pares, considerando 5 dimensões e 30 agrupamentos, com 5mil observações.

Na [Figura 3.17](#) notamos que o Fuzzy C-Means apresenta uma piora no geral, se aproximando mais do DBSCAN em todos os valores de sobreposição média de pares. Além disso, o Gaussian Mixture já não dista muito da performance dos outros métodos nesse caso.

Com o aumento do número de agrupamentos, há uma diminuição da informação, o que pode explicar uma queda nas performances de alguns métodos.

Vamos agora avaliar como os métodos se comportam variando o número de agrupamentos em conjuntos de alta dimensionalidade.

Assim como observado na [Seção 3.3](#), temos na [Figura 3.18](#) que o Fuzzy C-Means apresenta uma performance baixa quando há muitas dimensões, devida a complexidade matemática e computacional para atribuir os graus de pertinência de cada observação a cada um dos agrupamentos.

O método Spectral também apresenta uma performance abaixo se comparada aos outros métodos, assim como notado na [Seção 3.2](#).

A seguir, é possível avaliar como os métodos se comportam ao variar o número de dimensões quando há uma grande quantidade de agrupamentos.

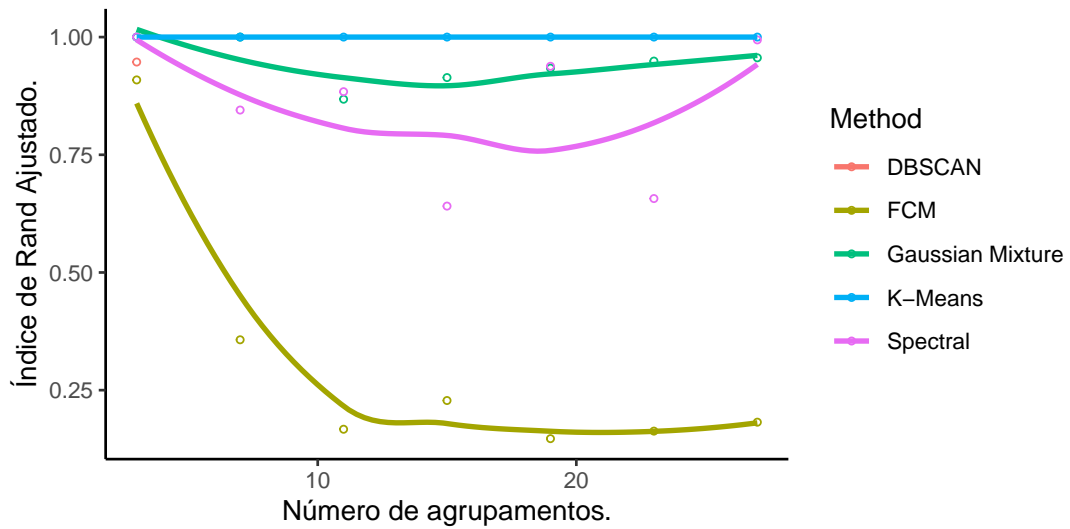


Figura 3.18: Índice de Rand Ajustado de cada método utilizado sobre diferentes números de agrupamentos, considerando 50 dimensões e 5mil observações.

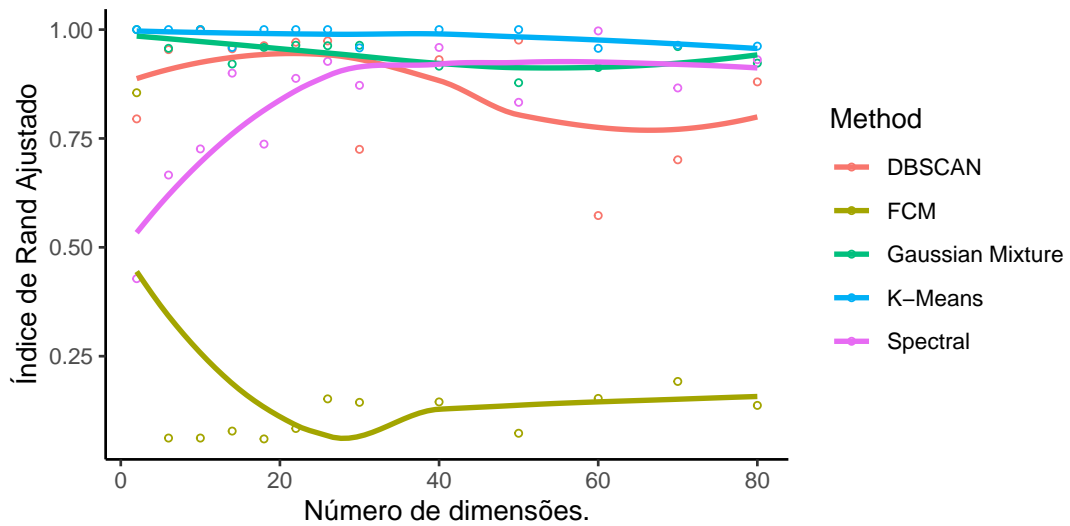


Figura 3.19: Índice de Rand Ajustado de cada método utilizado sobre diferentes números de dimensões, considerando 30 agrupamentos e 5mil observações.

Conforme se eleva o número de dimensões na [Figura 3.19](#), observa-se que o desempenho do algoritmo Fuzzy C-Means decresce, enquanto que o método Spectral apresenta um aumento no desempenho. Tal fenômeno no Fuzzy C-Means pode ser atribuído ao efeito mencionado na seção [Seção 3.3](#). Por outro lado, o desempenho crescente do método Spectral se deve ao acréscimo de informação proporcional ao aumento das dimensões.

Analisaremos a seguir o desempenho a partir do ARI ao variar o número de dimensões, considerando um nível de sobreposição médio de 5%.

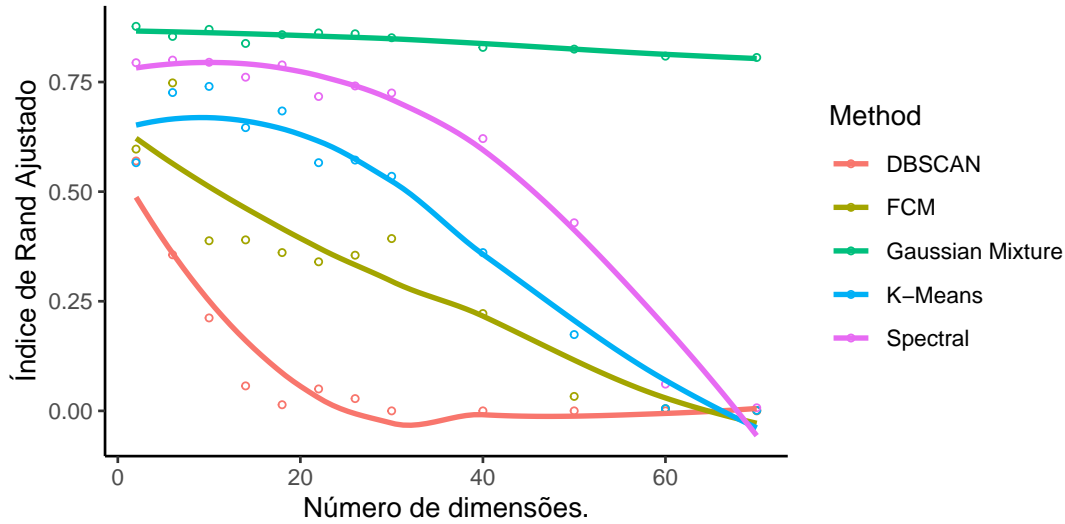


Figura 3.20: Índice de Rand Ajustado de cada método utilizado sobre diferentes números de dimensões, considerando 3 agrupamentos, 5mil observações e  $\bar{\omega} = 5\%$ .

Em contraste ao que observamos na Figura 3.16, notamos na Figura 3.20 novamente que o Gaussian Mixture desempenha melhor em conjuntos de alta dimensionalidade quando há sobreposição de pares. Em contrapartida, os outros métodos decrescem seu desempenho a medida que o número de dimensões aumenta, com o Índice de Rand Ajustado convergindo para zero.

### 3.6 Conclusões preliminares

Considerando todos os cenários testados, avaliamos, de modo geral, como foi o desempenho de cada método em estudo para cada uma das medidas de avaliação de clusterização que utilizamos nesse capítulo. A seguir, apresentamos o gráfico BoxPlot e uma tabela com medidas resumo para ilustrar a variabilidade do Índice de Rand Ajustado para cada um dos métodos em estudo.

Tabela 3.1: Medidas resumo para o Índice de Rand Ajustado por método utilizado em todos os cenários simulados.

Método	Min.	1° Quartil	Mediana	3° Quartil	Max.	Média	SD
<i>DBSCAN</i>	0.00	0.14	0.88	0.98	1.00	0.64	0.41
<i>FCM</i>	0.00	0.54	0.95	1.00	1.00	0.73	0.36
<i>Gaussian Mixture</i>	0.01	0.65	1.00	1.00	1.00	0.81	0.32
<i>K-Means</i>	0.00	0.56	1.00	1.00	1.00	0.78	0.37
<i>Spectral</i>	0.01	0.26	0.87	1.00	1.00	0.64	0.39

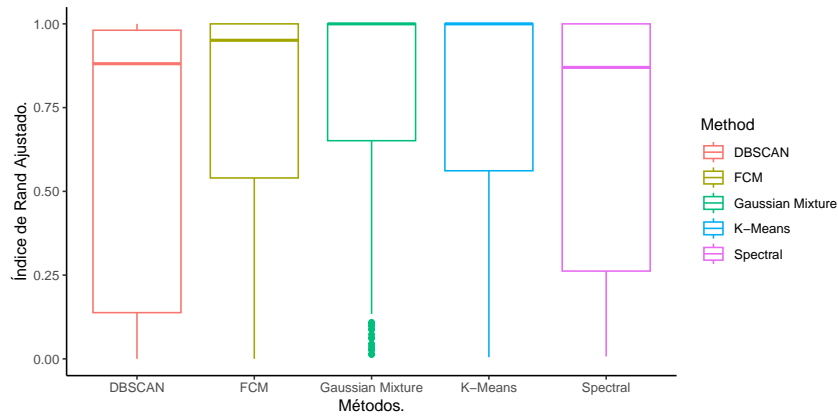


Figura 3.21: Boxplot para o Índice de Rand Ajustado considerando todos os cenários simulados.

Pela [Figura 3.21](#) e pela [Tabela 3.1](#), analisando o comportamento do Índice de Rand Ajustado, notamos que os Métodos *Fuzzy C-Means*, *Gaussian Mixture Model* e *K-Means* apresentam desempenho semelhantes, mantendo as medianas próximas a 1, o que indica que mais de 50% das execuções apresentaram bons resultados. Além disso, de modo geral, o método que se apresentou melhor foi o *Gaussian Mixture Model*, que possui mediana igual a 1 e primeiro quartil maior que os outros métodos, indicando menor desvio padrão do índice.

Por outro lado, temos o *Spectral* e o *DBSCAN* com desempenhos piores em relação aos outros métodos, apresentando medianas mais distantes de 1 e primeiro quartil abaixo de 0.3. No caso do *DBSCAN*, é ainda pior, pois nem mesmo o terceiro quartil está próximo de 1 assim como estão nos outros métodos. Isso corrobora com as avaliações feitas nas seções anteriores, indicando as deficiências que esse método possui ao se deparar com alguns cenários de clusterização.

É possível chegar a essa mesma conclusão analisando a [Figura 3.22](#) e a [Tabela 3.2](#), que avalia como os métodos se comportaram por meio da Variação da Informação.

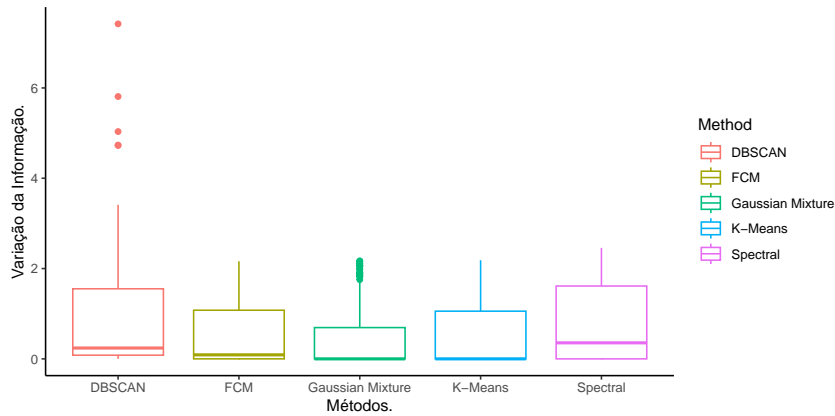


Figura 3.22: Boxplot para a Variação da Informação considerando todos os cenários simulados.

Tabela 3.2: Medidas resumo para a Variação da Informação por método utilizado em todos os cenários simulados.

Método	Min.	1° Quartil	Mediana	3° Quartil	Max.	Média	SD
<i>DBSCAN</i>	0.00	0.08	0.24	1.55	7.42	0.80	1.11
<i>FCM</i>	0.00	0.00	0.09	1.08	2.16	0.55	0.76
<i>Gaussian Mixture</i>	0.00	0.00	0.00	0.69	2.17	0.42	0.69
<i>K-Means</i>	0.00	0.00	0.00	1.06	2.18	0.49	0.81
<i>Spectral</i>	0.00	0.00	0.36	1.61	2.46	0.78	0.85

Observa-se o *DBSCAN* e o *Spectral* com maior Variação da Informação média, e medianas que distam dos demais métodos. Em contrapartida, temos o *K-Means*, *Fuzzy C-Means* e *Gaussian Mixture Model* com medianas iguais ou próximas a zero, indicando que no mínimo 50% dos resultados estiveram com quase nenhuma Variação da Informação.

Algo relevante foi constatado na [Subseção 3.1.5](#) e na [Subseção 3.1.4](#) sobre os índices de validação interna: eles não avaliam com precisão métodos que se baseiam em densidade (como o *DBSCAN*).

Na sequência, temos um BoxPlot e uma tabela com medidas resumo do Coeficiente de Silhueta

	Method	Min.	1° Quartil	Mediana	3° Quartil	Max.	Média	SD
1	DBSCAN	0.01	0.02	0.25	0.47			
2	FCM	0.00	0.01	0.02	0.18	0.58	0.12	0.19
3	Gaussian Mixture	0.01	0.02	0.03	0.38	0.66	0.19	0.21
4	K-Means	0.01	0.02	0.09	0.38	0.66	0.20	0.21
5	Spectral	0.00	0.00	0.02	0.03	0.61	0.13	0.21

Tabela 3.3: Medidas resumo para o Coeficiente de Silhueta por método utilizado em todos os cenários simulados.

Na [Figura 3.23](#) notamos que o *DBSCAN* possui desempenho semelhante aos outros

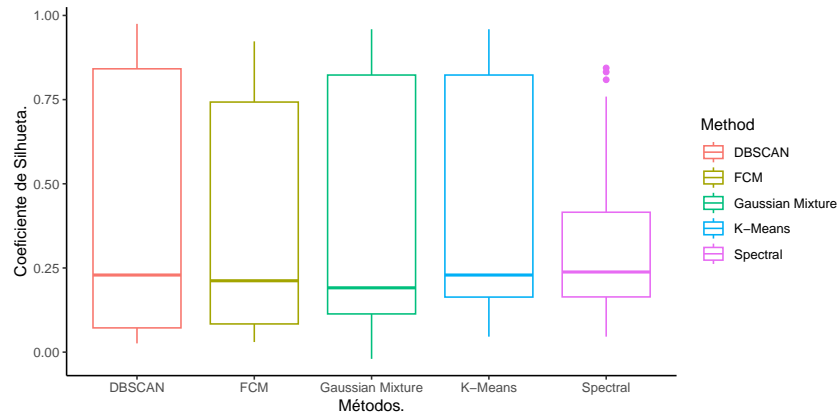


Figura 3.23: Boxplot para o Coeficiente de Silhueta considerando todos os cenários simulados.

métodos, exceto pelo *Spectral*, contrariando o Índice de Rand Ajustado e a Variação da Informação. Como estes são medidas que utilizam os rótulos verdadeiros dos agrupamentos, são mais confiáveis para se avaliar clusterizações.

O mesmo ocorre quando avaliamos pelo Índice de *Dunn*, que parece, inclusive, superestimar o *DBSCAN*, apresentando mediana superior em relação aos demais métodos. O comportamento do Índice de *Dunn*, nos cenários simulados, é mostrado na Figura 3.24 e na Tabela 3.4.

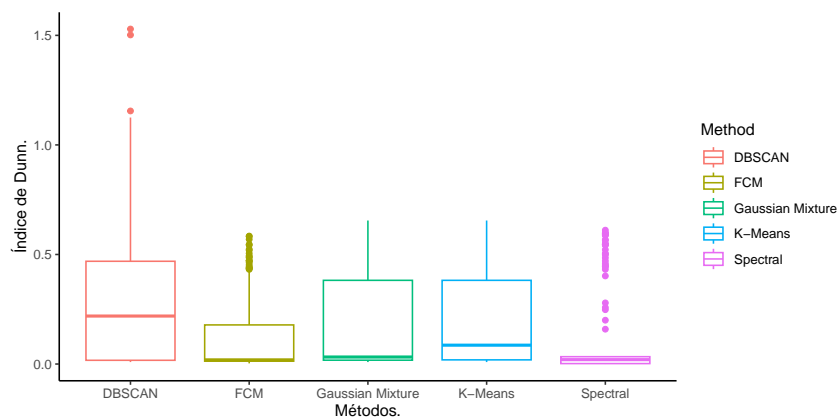


Figura 3.24: Boxplot para o Índice de *Dunn* considerando todos os cenários simulados.

Tabela 3.4: Medidas resumo para o Índice de *Dunn* por método utilizado em todos os cenários simulados.

Método	Min.	1° Quartil	Mediana	3° Quartil	Max.	Média	SD
<i>DBSCAN</i>	0.01	0.02	0.25	0.47			
<i>FCM</i>	0.00	0.01	0.02	0.18	0.58	0.12	0.19
<i>Gaussian Mixture</i>	0.01	0.02	0.03	0.38	0.66	0.19	0.21
<i>K-Means</i>	0.01	0.02	0.09	0.38	0.66	0.20	0.21
<i>Spectral</i>	0.00	0.00	0.02	0.03	0.61	0.13	0.21

Além disso, há ainda alguns pontos *outliers*, indicando bom agrupamento de acordo com o índice. Porém, o Índice de *Dunn* e o Coeficiente de Silhueta possuem alguns problemas quando o *DBSCAN* cria, aleatoriamente, agrupamentos pequenos, de até mesmo 1 único ponto dependendo dos parâmetros de densidades utilizados, fazendo com que a distância *intracluster* fique muito pequena, ou até mesmo 0 em alguns casos, fazendo com que o índice de *Dunn* fique muito grande.

Apesar de ser uma medida relativa e que está relacionada à capacidade computacional associada ao ambiente de execução, a comparação do tempo de execução é interessante para avaliarmos, indiretamente, a complexidade matemática e computacional de cada método.

Comparando os tempos de execução, por meio da [Figura 3.25](#) e da [Tabela 3.5](#), nota-se o método *Spectral* mais lento que o restante, com alguns pontos outliers que passam de 50 segundos, cujo primeiro quartil é superior ao terceiro quartil dos outros métodos, indicando que, no mínimo, 75% das execuções do *Spectral* foram mais lentas se comparada com os outros métodos.

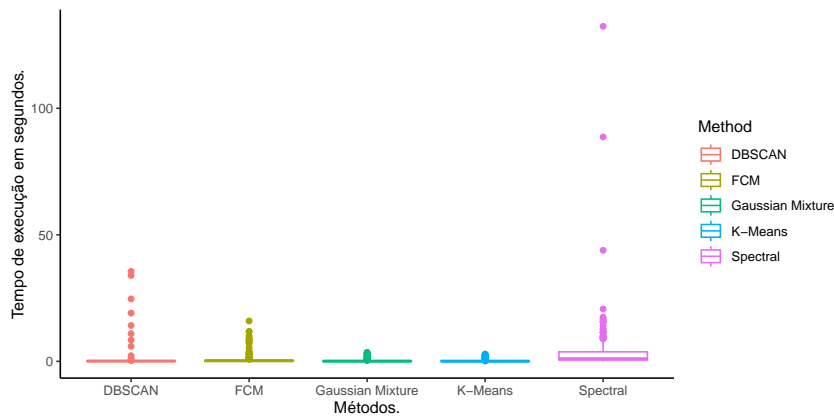


Figura 3.25: Boxplot para o tempo de execução em segundos considerando todos os cenários simulados.

Tabela 3.5: Medidas resumo para o tempo de execução em segundos por método utilizado em todos os cenários simulados.

Método	Min.	1° Quartil	Mediana	3° Quartil	Max.	Média	SD
<i>DBSCAN</i>	0.00	0.06	0.10	0.15	35.57	0.92	4.39
<i>FCM</i>	0.01	0.06	0.25	0.32	15.96	0.94	2.30
<i>Gaussian Mixture</i>	0.01	0.02	0.07	0.13	3.53	0.25	0.62
<i>K-Means</i>	0.01	0.03	0.06	0.09	2.92	0.21	0.51
<i>Spectral</i>	0.04	0.54	1.04	4.48	2230.68	24.04	172.10



## 3.7 Comparação dos métodos em agrupamentos não convexos

Nesta seção, realizamos uma comparação entre os métodos de agrupamento em conjuntos de dados não convexos. Diferentemente da seção anterior, na qual foram utilizadas misturas gaussianas, aqui usamos conjuntos de dados mais complexos e heterogêneos, disponíveis no *toy dataset*, da biblioteca *Scikit Learn* (Pedregosa *et al.* 2011) e os conjuntos de dados “Compound” (Zahn 1971), “Aggregation” (Gionis *et al.* 2007), “Pathbased” (Chang e Yeung 2008), “S2” (Fränti e Virtajoki 2006), “Flame” (Fu e Medico 2007) e “Face” (Ultsch e Pasewaldt 2005).

O objetivo dessa avaliação é verificar como os métodos se comportam em agrupamentos com formas não convexas e quais apresentam melhor desempenho nesses tipos de dados. Para isso, foi realizada uma análise quantitativa e qualitativa dos resultados, incluindo medidas de avaliação de agrupamento e visualizações gráficas dos resultados obtidos.

### 3.7.1 Comparando diferentes algoritmos de agrupamento por meio do *Toy Datasets*

Em Pedregosa *et al.* (2011), há exemplos que mostram as características de diferentes algoritmos de agrupamentos em conjuntos de dados que são “interessantes”, mas ainda em duas dimensões. Com exceção do último conjunto de dados, os parâmetros de cada um desses pares de conjunto de dados-algoritmo foram ajustados para produzir bons resultados de clusterização. Alguns algoritmos são mais sensíveis aos valores dos parâmetros do que outros.

O último conjunto de dados é um exemplo de situação “nula” para clusterização: os dados são homogêneos e não há uma boa clusterização. Para este exemplo, o conjunto de dados nulo usa os mesmos parâmetros que o conjunto de dados *blobs*, representado na Figura 3.26, o que representa uma falta de correspondência nos valores dos parâmetros e na estrutura dos dados. Embora esses exemplos forneçam alguma intuição sobre os algoritmos, essa intuição pode não se aplicar a dados com muitas dimensões.

Como esperado, métodos de agrupamentos que se baseiam em densidade lidam melhor em agrupamentos que não possuam formatos convexas, como é o caso do *circles* e do *moons*. A Figura 3.26 ilustra como o *Spectral* e o *DBSCAN* lidam melhor com esses

casos. Por outro lado, observamos que o *DBSCAN* desempenha mal quando se trata de agrupamentos com sobreposição de pares, como ocorre no *varied*.

Ainda na [Figura 3.26](#), é possível ver o grau de pertinência do *Fuzzy C-Means* atuar de acordo com o gradiente das cores, em que à medida que se aproxima do centroide definido, maior é o grau de pertinência. Porém, ainda é um método que utiliza particionamento como o *K-Means* e, por isso, também não lida bem com agrupamentos não convexos.

Nos casos em que foram criados agrupamentos convexos (*varied*, *aniso* e *blobs*), os métodos *K-Means*, *Gaussian Mixture Model* e *Fuzzy C-Means* apresentaram resultados satisfatórios.

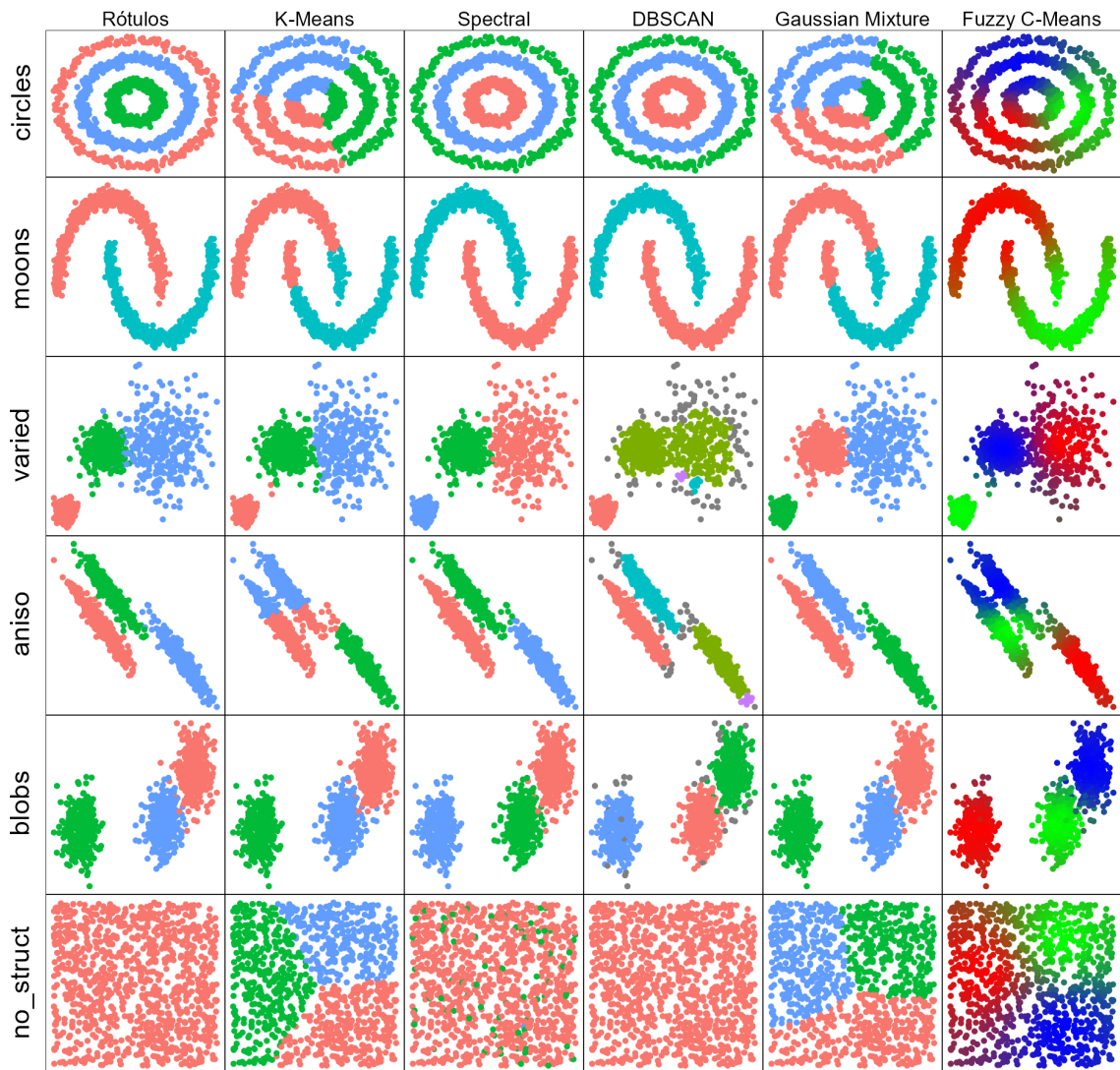


Figura 3.26: Comparação dos métodos em estudo aplicados respectivamente nos conjuntos de dados gerados por *noisy\_circles*, *noisy\_moons*, *varied*, *aniso*, *blobs* e *no\_structure* do *Toy Datasets*.

Fonte: Disponível em

[https://scikit-learn.org/stable/auto\\_examples/cluster/plot\\_cluster\\_comparison.html](https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html)  
 (“Rótulos” na figura representa os rótulos reais dos agrupamentos).

Em algumas situações, os rótulos de atribuição dos agrupamentos podem apresentar divergências em relação aos resultados de outros agrupamentos, como evidenciado na Figura 3.26 e na Figura 3.27. Isso acontece porque muitos algoritmos de clusterização usam técnicas iterativas para definir os agrupamentos, em que os objetos são inicialmente atribuídos a um agrupamento e depois ajustados de forma iterativa até que os agrupamentos finais sejam obtidos. Métodos lineares, como o *DBSCAN*, também podem apresentar essa divergência, dependendo da inicialização do algoritmo. Durante esse processo, é possível que um objeto seja atribuído a um agrupamento diferente em diferentes

iterações, devido às regras de atribuição e aos parâmetros definidos pelo algoritmo.

Apesar de terem sido rotulados de maneira diferente, as classificações podem estar corretas, o que é levado em consideração pelos índices utilizados neste estudo, como o Índice de Rand Ajustado e a Proporção de Classificações Corretas.

### 3.7.2 Comparando diferentes algoritmos de agrupamento com *Clustering Datasets*

Nas aplicações a seguir, foram utilizados conjuntos de dados com formatos, densidades e número de componentes diferentes com objetivo de ilustrar algumas exceções no comportamento dos métodos de agrupamento. Nos resultados apresentados na [Subseção 3.7.1](#), notamos como o *DBSCAN* e o *Spectral* lidam melhor com formatos não convexos de agrupamento. Na [Figura 3.27](#), mostramos a clusterização considerando alguns conjuntos de dados de [Ultsch e Pasewaldt \(2005\)](#), que apresentam formas e configurações interessantes.

Na primeira linha da [Figura 3.27](#), visualizamos os resultados de cada método para o conjunto de dados *Compound*, que possui formatos e densidades distintas. O *DBSCAN* obteve um bom resultado, apesar de algumas classificações terem sido feitas como ruído. Já o *Spectral* não obteve êxito em classificar corretamente os agrupamentos, assim como o *K-Means*, *Gaussian Mixture Model* e *Fuzzy C-Means*.

Logo em seguida, o *Aggregation* também apresenta formatos diversos, mas sem variar a densidade. Nesse conjunto de dados, o *Spectral* e o *DBSCAN* se destacaram, seguido do *Gaussian Mixture Model*. *Fuzzy C-Means* e *K-Means* não conseguiram classificar corretamente os agrupamentos, devido ao particionamento que é realizado pelo método.

Já no *PathBased*, temos uma situação diferente do usual: são formas não convexas em que o *Spectral* e o *DBSCAN* não obtiveram um bom desempenho. Isso elucida que existem fatores como densidade e sobreposição, que impactam no resultado de uma clusterização.

Para o *S2*, que são agrupamentos convexos, todos os métodos tiveram um resultado satisfatório, exceto pelo *DBSCAN*, que sofreu com a sobreposição de pares presente no conjunto de dados.

Por fim, em relação aos conjuntos de dados *Flame* e *Face*, o *Spectral* e o *DBSCAN* lidaram muito bem com formatos diferentes nos agrupamentos, enquanto que os outros tiveram dificuldade e não conseguiram classificar corretamente.

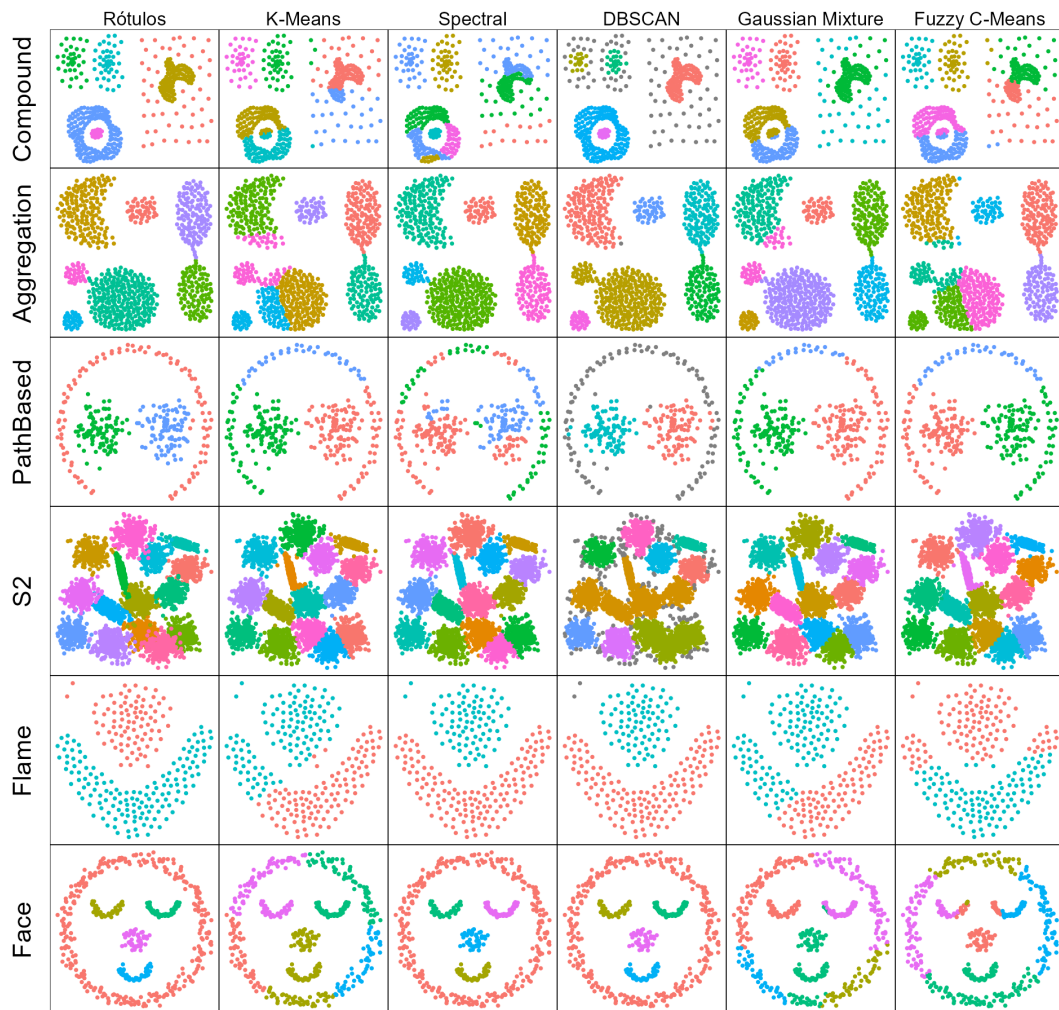


Figura 3.27: Comparação dos métodos em estudo aplicados nos conjuntos de dados *Compound*, *Aggregation*, *PathBased*, *S2*, *Flame* e *Face*, obtidos a partir do repositório [Ultsch e Pasevaldt \(2005\)](#) e do [elbamos \(2021\)](#) (“Rótulos” na figura representa os rótulos reais dos agrupamentos).



# Capítulo 4

## Considerações Finais

Diante do desafio de resumir a informação contida em um conjunto de dados multivariado, a criação de agrupamentos se apresenta como uma alternativa importante. Este trabalho se dedicou a comparar diferentes métodos de agrupamento, buscando avaliar sua qualidade em diferentes cenários e fornecer subsídios para a escolha do método mais adequado para determinado conjunto de dados.

A análise foi conduzida com os algoritmos *DBSCAN*, *Fuzzy C-Means*, *Gaussian Mixture Model*, *Spectral* e *K-Means*, os quais foram avaliados com o auxílio de índices de avaliação da qualidade de agrupamentos. A comparação dos resultados e a validação dos agrupamentos são fundamentais para verificar a consistência e a qualidade das análises de agrupamentos, permitindo a tomada de decisões mais seguras e confiáveis em diferentes áreas de pesquisa.

Durante a realização das simulações, pudemos nos deparar com alguns problemas como extrapolação de memória e/ou capacidade de processamento, principalmente, em grandes conjuntos de dados. Apesar de não ter sido tratado nesse estudo, é importante também considerar a otimização dos processos e a redução da informação antes de ser imputada no algoritmo de clusterização.

As conclusões obtidas a partir dos resultados mostrados no [Capítulo 3](#), mesmo que limitadas aos cenários e contextos criados, buscam ilustrar, por meio de simulações, de que forma os métodos são utilizados na busca de obter agrupamentos e o quão eficiente eles são, dependendo do cenário em que é considerado.

Apesar das conclusões obtidas na [Seção 3.6](#), nem sempre conseguimos obter informações sobre os agrupamentos que buscamos no conjunto de dados, restando apenas algumas medidas de validação interna como o Índice de *Dunn* e o Coeficiente de Silhueta, que nem

sempre são eficazes em avaliar a qualidade da clusterização, mas auxiliam na busca por agrupamentos e na definição de alguns parâmetros de iniciação de algoritmos. Alguns algoritmos, como o *DBSCAN* e o *Spectral*, necessitam de parâmetros relacionados à densidade e/ou similaridade para a formação dos grupos, que podem ser escolhidos com o auxílio de medidas de validação.

Os resultados mostraram que não há um método universalmente melhor e que a escolha do método mais adequado dependerá das características do conjunto de dados e dos objetivos da análise. Porém, alguns métodos se mostraram mais robustos em diferentes cenários, como o *DBSCAN*, para dados densamente agrupados, e o *Gaussian Mixture Model*, para dados com sobreposição de grupos.

Em resumo, a escolha do método de agrupamento adequado é crucial para uma análise multivariada efetiva. A comparação e validação dos resultados são fundamentais para garantir a confiabilidade da análise e a interpretação correta dos grupos formados. Esperamos que este trabalho possa contribuir para a escolha e aplicação adequada de métodos de agrupamento em diferentes contextos de análise multivariada.



# Referências Bibliográficas

- Arthur, D. e Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. Em *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, páginas 1027–1035. SIAM.
- Bellatreche, L. e Chakravarthy, S. (2017). *Big Data Analytics and Knowledge Discovery: 19th International Conference, DaWaK 2017, Lyon, France, August 28–31, 2017, Proceedings*. Lecture Notes in Computer Science. Springer International Publishing. ISBN 9783319642833.
- Bellman, R. (1961). Adaptive control processes: a guided tour. *Princeton University Press*.
- Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA, first edition.
- Boonchoo, T., Ao, X. e He, Q. (2018). An efficient density-based clustering algorithm for higher-dimensional data. *CoRR*, **abs/1801.06965**.
- Bouveyron, C., Celeux, G., Murphy, T. B. e Raftery, A. E. (2019). *Model-Based Clustering and Classification for Data Science: With Applications in R*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.
- Chang, H. e Yeung, D.-Y. (2008). Robust path-based spectral clustering. *Pattern Recognition*, **41**(1), 191–203.
- Clayton, T. e Fortuna, T. (2019). The curse of dimensionality in machine learning and data science: An overview. *arXiv preprint arXiv:1907.02523*.
- Davies, R. B. (1980). The distribution of a linear combination of  $\chi^2$  random variables. *Applied Statistics*, **29**, 323–333.

- Donath, W. E. e Hoffman, A. J. (1973). Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, **17**(5), 420–425.
- Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, **3**(3), 32–57.
- elbamos (2021). Clusteringdatasets. <https://github.com/elbamos/clusteringdatasets>. Accessed: [10-March-2023].
- Ester, M., Kriegel, H.-P., Sander, J. e Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. páginas 226–231. AAAI Press.
- Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, **23**, 298–305.
- Forgy, E. (1965). Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics*, **21**, 768–780.
- Fraley, C. e Raftery, A. E. (2000). Model-based clustering, discriminant analysis, and density estimation. *JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION*, **97**, 611–631.
- Fränti, P. e Virtajoki, O. (2006). Iterative shrinking method for clustering problems. *Pattern Recognition*, **39**(5), 761–765.
- Fu, L. M. e Medico, E. (2007). Flame, a novel fuzzy clustering method for the analysis of dna microarray data. *BMC bioinformatics*, **8**(1), 3.
- Gionis, A., Mannila, H. e Tsaparas, P. (2007). Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, **1**(1), 1–30.
- Hennig, C. e Liao, T. F. (2013). How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **62**(3), 309–369.
- Hubert, L. e Arabie, P. (1985). Comparing partitions. *Journal of Classification*, **2**(1), 193–218.
- Jafelice, R., Barros, L. e Bassanezi, R. (2012). *Teoria dos conjuntos fuzzy com aplicações*. ISBN ISSN 2236-5915.

- Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, **31**(8), 651–666.
- Jain, L., Peng, S., Alhadidi, B. e Pal, S. (2020). *Intelligent Computing Paradigm and Cutting-edge Technologies: Proceedings of the First International Conference on Innovative Computing and Cutting-edge Technologies (ICICCT 2019), Istanbul, Turkey, October 30-31, 2019*. Learning and Analytics in Intelligent Systems. Springer International Publishing. ISBN 9783030385019.
- Kassambara, A. (2017). *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning. Edition 1*. STHDA.
- Masulli, F., Petrosino, A. e Rovetta, S. (2015). *Clustering High-Dimensional Data: First International Workshop, CHDD 2012, Naples, Italy, May 15, 2012, Revised Selected Papers*. Lecture Notes in Computer Science. Springer Berlin Heidelberg. ISBN 9783662485774.
- Meilă, M. e Shi, J. (2000). Learning segmentation by random walks. Em *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS’00, página 837–843, Cambridge, MA, USA. MIT Press.
- Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, **98**(5), 873–895.
- Melnykov, V. e Maitra, R. (2010). Finite mixture models and model-based clustering. *Statistics Surveys*, **4**(none), 80 – 116.
- Melnykov, V., Chen, W.-C. e Maitra, R. (2012). Mixsim: An r package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software, Articles*, **51**(12), 1–25.
- Pearson, K. e Henrici, O. M. F. E. (1894). Iii. contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. (A.)*, **185**, 71–110.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. e Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, **66**(336), 846–850.
- Ruspini, E. H. (1977). A theory of fuzzy clustering. Em *1977 IEEE Conference on Decision and Control including the 16th Symposium on Adaptive Processes and A Special Symposium on Fuzzy Set Theory and Applications*, páginas 1378–1383.
- Shinnou, H. e Sasaki, M. (2008). Spectral clustering for a large data set by reducing the similarity matrix size. Em *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).
- Steinley, D. (2006). K-means clustering: a half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, **59**(1), 1–34.
- Tan, P.-N., Steinbach, M. e Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley, Boston, MA, USA, first edition.
- Tran, T. N., Drab, K. e Daszykowski, M. (2013). Revised dbscan algorithm to cluster data with dense adjacent clusters. *Chemometrics and Intelligent Laboratory Systems*, **120**, 92–96.
- Ultsch, A. e Pasewaldt, M. M. (2005). Clustering benchmark datasets (<http://cs.joensuu.fi/sipu/datasets/>). [Online; accessed 12-March-2023].
- Wang, S., Gu, J. e Chen, F. (2015). Clustering high-dimensional data via spectral clustering using collaborative representation coefficients. Em D.-S. Huang, K.-H. Jo, e A. Hussain, editors, *Intelligent Computing Theories and Methodologies*, páginas 248–258, Cham. Springer International Publishing. ISBN 978-3-319-22186-1.
- Weiss, Y. (1999). Segmentation using eigenvectors: a unifying view. Em *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, páginas 975–982 vol.2.
- Winkler, R., Klawonn, F. e Kruse, R. (2010). Fuzzy c-means in high dimensional spaces. *International Journal of Fuzzy System Applications*, **11**.
- Wu, J. (2012). *Advances in K-Means Clustering: A Data Mining Thinking*. Springer.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, **8**(3), 338–353.

Zahn, C. T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, **100**(1), 68–86.