

DBSCAN

Density-Based Spatial Clustering of Applications with Noise

Davi Esmeraldo da Silva Albuquerque

Daniel Paranagua Fontenelle Fraga

5 de fevereiro de 2025

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) foi proposto em 1996 no Instituto de Ciência da Computação da Universidade de Munique pelos autores:

- **Martin Ester** - Professor de Ciência da Computação na **Simon Fraser University, Canadá**.
- **Hans-Peter Kriegel** - Cientista da Computação Alemão e Professor na **Ludwig Maximilian University de Munique, Alemanha**.
- **Jörg Sander** - Professor de Ciências da Computação na **University of Alberta, Canadá**.
- **Xiaowei Xu** - Professor de Ciência da Informação na **University of Arkansas at Little Rock, Estados Unidos**.



Introdução ao DBSCAN

O **DBSCAN** foi introduzido como uma resposta às limitações dos algoritmos de clustering existentes à época (Algoritmos de particionamento e hierárquicos (Kaufman Rousseeuw 1990)). As principais motivações foram:

- Identificação de Clusters de Forma Arbitrária sem qualquer informação preliminar sobre os grupos.
- Número de Clusters Não Definido a Priori
- Robustez a Outliers nos dados.
- Lidar bem com grandes bases de dados
- Lidar bem com dados espaciais como esféricos, alongados, não lineares, etc.



O **DBSCAN** é um algoritmo de agrupamento baseado em densidade. Desse modo, agrupa pontos densamente conectados em regiões do espaço de dados, formando clusters.

Características principais:

- Identifica clusters como regiões densas separadas por áreas de baixa densidade.
- Identifica **outliers** como pontos em regiões de baixa densidade (vizinhos muito distantes).
- Algoritmo de agrupamento **não-paramétrico**.



- Esses algoritmos de agrupamento são amplamente utilizados na prática com aplicações que vão desde :
 - A descoberta de valores discrepantes em conjuntos de dados para **Prevenção de Fraudes** (Breunig, Kriegel, Ng e Sander 2000)
 - **Descoberta de padrões em dados de Streaming** (Chen e Tu 2007; Cao, Ester, Qian e Zhou 2006)
 - **Sinais Ruidosos** (Kriegel e Pfeifle 2005; Ester et al. 1996; Tran, Wehrens e Buydens 2006; Hinneburg e Keim 1998; Duan, Xu, Guo, Lee e Yan 2007)



- Esses algoritmos de agrupamento são amplamente utilizados na prática com aplicações que vão desde :
 - **Dados de Expressão Genética** (Jiang, Pei e Zhang 2003)
 - **Bancos de Dados Multimídia** (Kisilevich, Mansmann e Keim 2010)
 - **Tráfego rodoviário** (Li, Han, Lee e Gonzalez 2007).
 - **Dados Geoespaciais** (Chen, Ji, e Wang, 2014).

Dados geoespaciais frequentemente possuem pontos ruidosos devido a erros de estimativa em sensores habilitados por GPS.

Clusters podem assumir formas únicas influenciadas pelo espaço físico onde os dados foram capturados.



Parâmetros:

- ε (**epsilon**): Representa o raio de vizinhança ao redor de cada ponto para considerar densidade de agrupamento.
- **minPts**: Número mínimo de pontos dentro da distância ε para formar um cluster.

Vizinhança ϵ : Região ao redor de um ponto p com raio ϵ .

$$N_{\epsilon}(p) = \{q \in D \mid \text{dist}(p, q) \leq \epsilon\}$$

Onde:

- **$N_{\epsilon}(\mathbf{p})$:** Representa o conjunto de pontos que estão na vizinhança ϵ do ponto p . É o conjunto de vizinhos diretos de p dentro de uma distância máxima ϵ .
- **$\text{dist}(\mathbf{p}, \mathbf{q})$:** Função de distância que mede a proximidade entre p e q . Pode ser qualquer métrica apropriada, como a distância Euclidiana, Manhattan ou Cosine.
- **$\{\mathbf{q} \in \mathbf{D} \mid \text{dist}(\mathbf{p}, \mathbf{q}) \leq \epsilon\}$:** Conjunto de todos os pontos q em D que estão a uma distância $\leq \epsilon$ de p .



Distância Euclidiana (Padrão no DBSCAN):

- Mede a distância linear entre dois pontos no espaço.
- Recomendada quando as variáveis estão na mesma escala.

Impacto da Função de Distância:

- A escolha da métrica impacta diretamente o valor de ε (raio de vizinhança) e os resultados.
- Distâncias inadequadas podem levar a clusters distorcidos ou à classificação incorreta de ruídos.
- Em altas dimensões, os dados tendem a ficar mais espalhados no espaço, tornando difícil distinguir pontos próximos de pontos distantes. Dificulta a escolha ideal dos parâmetros e consequentemente reduz a eficácia do DBSCAN. (Curse of Dimensionality)



Classificação dos pontos:

- **Pontos Núcleo (Core Point):** Contém pelo menos **minPts** pontos em sua vizinhança (ε).
- **Pontos de Borda (Border Points):** Estão na vizinhança de um núcleo, mas não são núcleo. Um ponto é considerado borda quando não possui MinPts dentro de um raio Eps de distância, porém está inserido dentro de um raio Eps de um outro ponto no qual é núcleo.
- **Ruído (Outlier/Noise):** Pontos que não são acessíveis a partir de qualquer outro ponto. Desse modo, pontos que não são pontos de borda ou núcleo.

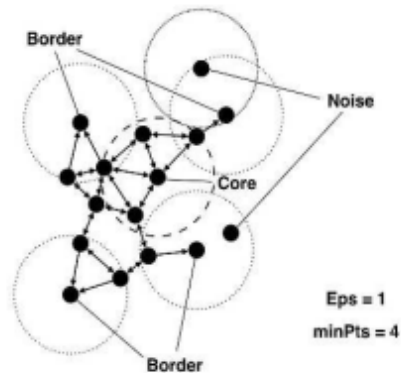


Resumo do processo:

- 1 Escolher um ponto arbitrário p no conjunto de dados.
- 2 Determinar a sua vizinhança $\epsilon (N_\epsilon(p))$.
- 3 Identificar se p é um ponto núcleo (tem ao menos **minPts** pontos em sua vizinhança):
 - Se for um ponto núcleo:
 - Criar um novo cluster (se ainda não pertence a um).
 - Expandir o cluster conectando todos os pontos densamente alcançáveis.
 - Se não for um ponto núcleo:
 - Classificar p como ponto de borda (se conectado a um núcleo) ou ruído (outlier, se não conectado).
- 4 Repetir até que todos os pontos sejam classificados como núcleo, borda ou ruído.



Exemplo



Descrição:

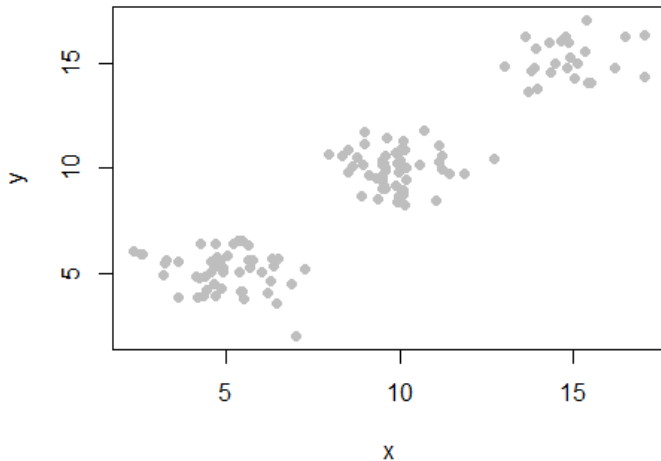
- Três grupos de dados gerados com distribuições normais diferentes.
- Os dados são bidimensionais (x e y) e foram gerados aleatoriamente.
- Cada grupo representa um possível cluster.

Código:

```
set.seed(42)
data <- rbind(
  matrix(rnorm(100, mean = 5, sd = 1), ncol = 2),
  matrix(rnorm(100, mean = 10, sd = 1), ncol = 2),
  matrix(rnorm(50, mean = 15, sd = 1), ncol = 2))
plot(data, col = "gray", pch = 19, main = "Dados de Exemplo")
```



Dados de Exemplo



O algoritmo K-means tem como objetivo dividir uma amostra em K clusters, minimizando a distância entre os pontos e seus respectivos centróides.

Funcionamento:

- Inicializa K centróides aleatórios.
- Atribui cada ponto ao centróide mais próximo.
- Atualiza os centróides calculando a média dos pontos atribuídos a cada cluster.
- Repete o processo até que a mudança nos centróides seja mínima.

Exemplo Didático - Resultado K-means

- O K-means ($K=3$) identificou 3 clusters, com os seguintes tamanhos:
 - Cluster 1: 50 pontos
 - Cluster 2: 25 pontos
 - Cluster 3: 50 pontos

- Centroides dos clusters (x, y):
 - Cluster 1: (4.96, 5.10)
 - Cluster 2: (9.85, 9.98)
 - Cluster 3: (14.86, 15.15)

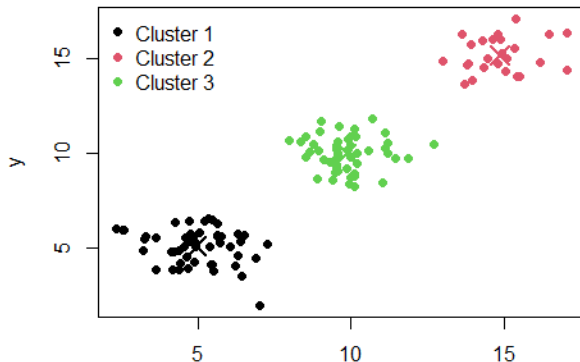


Exemplo Didático - Resultado K-means

O algoritmo K-means:

- Atribuiu cada ponto ao cluster mais próximo (baseado na distância euclidiana).
- Calculou os centroides, representados pelas cruzes, para cada cluster.

Clusters identificados pelo K-means



Parâmetros Selecionados Arbitrariamente

- $Eps = 1.5$: Define o raio de vizinhança.
- $MinPts = 5$: Define o número mínimo de pontos necessários para formar um cluster.

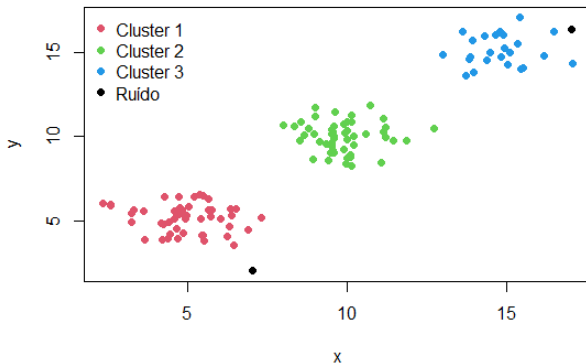
Resultados Esperados

- Identificar 3 clusters principais.
- Alguns pontos sendo considerados **ruído**.

Exemplo Didático - DBSCAN

- Pontos em preto são classificados como **ruído** devido à baixa densidade em suas vizinhanças.

Clusters identificados pelo DBSCAN



Exemplo Didático - Ajuste Iterativo dos Hiperparâmetros

Visualização Shiny

Impacto da Escolha do Parâmetro ε

A escolha dos parâmetros é crucial para o bom desempenho do DBSCAN.

Impacto do valor de ε :

■ Eps pequeno:

- Muitos pontos serão classificados como ruído, especialmente os mais distantes.
- Reduz o risco de incluir muitos outliers dentro de clusters.
- Ideal para detectar pequenos grupos densos, mas pode fragmentar clusters reais.

■ Eps alto:

- Poucos clusters, mas muito generalizados.
- Pode resultar na fusão de clusters distintos em um único cluster.



Impacto da Escolha do Parâmetro MinPts

A escolha de `minPts` é essencial para equilibrar a **sensibilidade** do DBSCAN ao ruído e à formação de clusters.

Impacto do valor de `minPts`:

■ `MinPts` pequeno:

- Permite a formação de clusters pequenos e densos.
- Pode incluir ruído em clusters, reduzindo a qualidade dos agrupamentos.

■ `MinPts` alto:

- Exige maior densidade para formar clusters.
- Evita clusters formados por pontos ruidosos ou esparsos.
- Ideal para conjuntos de dados com muito ruído ou grande dimensionalidade.
- Pode ignorar clusters pequenos e relevantes.



Impacto dos Parâmetros do DBSCAN

Eps	MinPts	Comportamento
Baixo	Baixo	Clusters pequenos e instáveis.
Baixo	Alto	Apenas os pontos mais densos são agrupados em clusters. Grande número de pontos sendo classificados como <i>ruído</i> , especialmente em regiões menos densas.
Alto	Baixo	O modelo tende a formar clusters grandes, com muitos pontos no mesmo grupo, mas com poucos clusters distintos.
Alto	Alto	Clusters bem definidos, mas com uma densidade maior necessária para formar um cluster. Pontos em regiões menos densas provavelmente serão classificados como <i>ruído</i> .



Método para Escolha de minPts :

■ Regras práticas:

- $\text{minPts} \geq D + 1$, onde D é o número de dimensões do conjunto de dados.
- Recomendação: $\text{minPts} = 2 \cdot D$ para:
 - Dados ruidosos.
 - Conjuntos de dados grandes.

Recomendação prática: Ajustar ε e minPts iterativamente para obter clusters bem definidos e minimizar ruído.

Método da Distância k-Vizinha (k-Distance Plot)

Objetivo: Determinar o valor ideal de ε no DBSCAN. Esse gráfico ajuda a identificar um limiar natural para a densidade dos clusters, distinguindo melhor os pontos pertencentes a clusters dos que serão considerados ruído.

Passos:

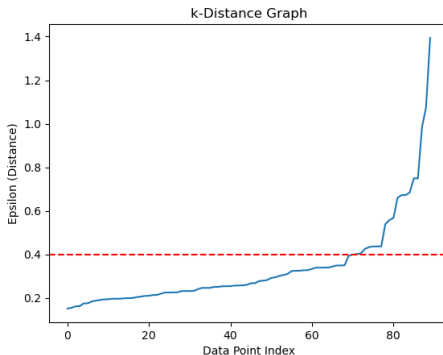
- 1 Calcular a distância do k -ésimo vizinho para cada ponto.
- 2 Ordenar essas distâncias de forma crescente.
- 3 Construir um gráfico onde:
 - Eixo X: Índices dos pontos ordenados.
 - Eixo Y: Distância para o k -ésimo vizinho.
- 4 Identificar o **ponto de inflexão** ("cotovelo da curva"), que representa um bom valor para ε .



Interpretação do Gráfico

Regiões do gráfico:

- **Crescimento Lento:** Pontos bem conectados dentro de clusters.
- **Ponto de Inflexão ("Cotovelo"):** Melhor escolha para ϵ , pois separa clusters do ruído.
- **Crescimento Acentuado:** Representa pontos isolados e ruídos, com distâncias maiores para seus vizinhos.



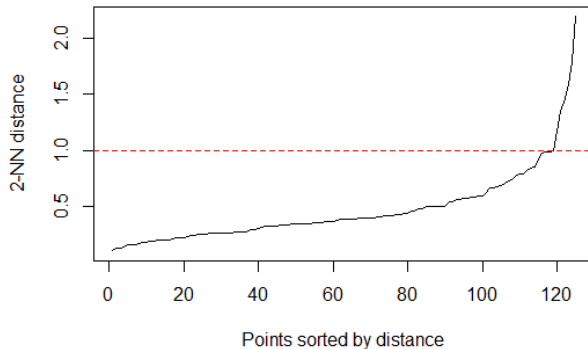
Método para Escolha de ε :

- Utilize o gráfico de distância k (k-distance):
 - Ordene as distâncias para o $k = \text{minPts} - 1$ vizinho mais próximo.
 - Plote as distâncias e procure o "cotovelo" (*elbow*), que representa a transição natural entre densidades.

Exemplo Didático - DBSCAN

Parâmetros Selecionados com base em:

- $\text{MinPts} = \text{dim} + 1 = 3$
- $\text{Eps} \rightarrow k\text{-NN distance para } k = \text{minPts} - 1 = 2$

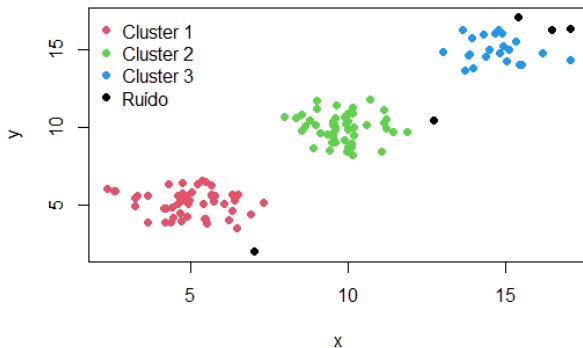


Exemplo Didático - DBSCAN

Parâmetros Selecionados

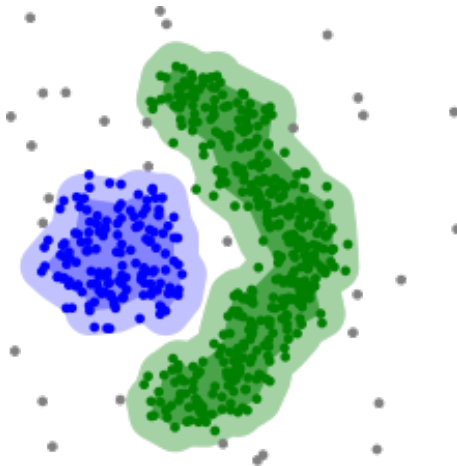
- Eps = 2
- MinPts = 3

Clusters identificados pelo DBSCAN



Exemplo

O DBSCAN pode encontrar clusters não linearmente separáveis.



■ Pacote **dbscan**:

- Implementação rápida devido ao uso de Estruturas Espacial *K-D Tree* ou *R-Tree* acelera a busca, reduzindo o custo computacional.
- Somente Distância Euclidiana
- Suporte para:
 - DBSCAN
 - HDBSCAN
 - OPTICS

■ Pacote **fpc**:

- Suporte para funções de distância arbitrárias via matrizes de distância.
- Mais lento devido ao uso do interpretador R.



Principais Vantagens:

■ Não requer o número de clusters a priori:

- Diferentemente do k-means, o número de clusters é determinado automaticamente com base na densidade.
- Insensível à inicialização (como em k-means).

■ Identifica clusters de formas arbitrárias:

- É capaz de encontrar clusters não convexos e até mesmo clusters cercados por outros clusters.

■ Robustez a ruídos:

- Possui uma noção explícita de "ruído" (*noise*) e é resistente a outliers.



Principais Vantagens:

- Requer apenas dois parâmetros (ϵ e **MinPts**), sendo amplamente insensível à ordem dos pontos no banco de dados.
- Pode lidar com grandes bases de dados.
- Ideal para dados espaciais com clusters complexos.

Desvantagens do DBSCAN

■ Não é totalmente determinístico:

- Pontos de borda (*border points*) podem pertencer a diferentes clusters dependendo da ordem de processamento dos dados.

■ Dependência da métrica de distância:

- O algoritmo é sensível à métrica utilizada no cálculo de proximidade geralmente a distância Euclidiana.
- No caso de dados de alta dimensionalidade, o "Efeito da Dimensionalidade" (**Curse of Dimensionality**) pode tornar a distância Euclidiana menos efetiva.

■ Dificuldade na Escolha dos parâmetros

- Se os dados ou a escala não forem bem compreendidos, definir um valor apropriado para os parâmetros pode ser desafiador.



DBSCAN: Um dos algoritmos de clustering mais usados e citados

- Reconhecimento acadêmico e prático significativo.
- Citado entre os 25 artigos mais influentes em mineração de dados, segundo a **Microsoft Academic Search** (posição #24, 2010).
- Em 2014, recebeu o **Test of Time Award**, concedido pelo ACM SIGKDD (premiação para algoritmos com impacto duradouro em teoria e prática).
- Em julho de 2020, o artigo de revisão: *"DBSCAN Revisited: Why and How You Should (Still) Use DBSCAN"* foi listado entre os 8 artigos mais baixados da revista **ACM Transactions on Database Systems (TODS)**.

O DBSCAN inspirou diversas variações que foram desenvolvidas para lidar com desafios específicos, como clusters de densidade variável ou alta dimensionalidade.

Variações:

- HDBSCAN
- OPTICS
- DENCLUE

O que é HDBSCAN?

- **HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise)** é uma extensão do algoritmo DBSCAN.
- Constrói uma hierarquia de clusters baseada na **densidade** dos dados, permitindo uma análise mais detalhada da estrutura dos dados.

Vantagens do HDBSCAN

- **Descoberta automática de Clusters:** O HDBSCAN não requer a especificação prévia do número de clusters no conjunto de dados, tornando-o adequado para conjuntos de dados com densidades variáveis e estruturas complexas.
- **Identificação de Clusters com formas e tamanhos variados:** O HDBSCAN é capaz de identificar clusters de formas e tamanhos variados, incluindo clusters que são não convexos e possuem formas irregulares.
- **Detectar ruídos automaticamente:** O HDBSCAN possui um mecanismo eficaz para identificar pontos de dados que não pertencem a nenhum cluster, classificando-os como ruído. Isso é particularmente útil em conjuntos de dados que contêm outliers ou pontos de dados anômalos.



- **Desempenho em dados de alta dimensionalidade:** Embora o HDBSCAN seja melhor que o DBSCAN em situações com alta variabilidade de densidade, ele ainda pode sofrer no caso de alta dimensionalidade. Em espaços de alta dimensão, a distância entre pontos tende a se tornar mais uniforme, o que dificulta a identificação de clusters. O algoritmo pode, nesse caso, falhar em detectar a estrutura real dos dados.



- **Exigência Computacional para grandes conjuntos de dados:** O HDBSCAN pode ser mais computacionalmente custoso do que outros métodos de clusterização, especialmente em grandes conjuntos de dados, por conta da complexidade do cálculo de hierarquia exigir mais tempo de processamento e memória.

- **Conhecimento prévio:** Para usar HDBSCAN em bases mais complexas, de maneira eficaz, é necessário algum conhecimento prévio dos dados. O usuário precisa entender as características de densidade e as variações que o algoritmo pode encontrar.

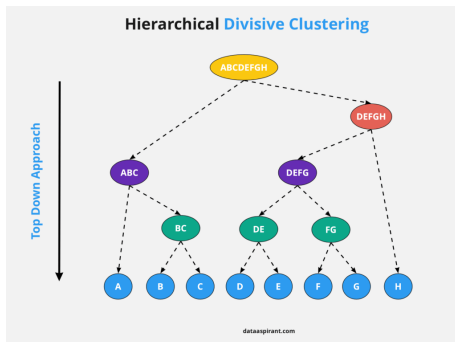
Como o HDBSCAN funciona?

- Sklearn: É uma biblioteca de aprendizado de máquina de código aberto para a linguagem de programação Python.

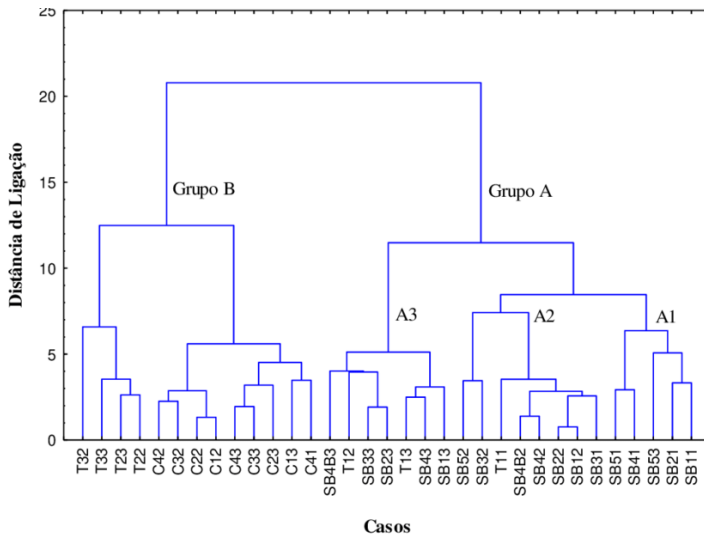


Como o HDBSCAN funciona?

- 1 Calcula a **distância de alcance mútua**, ajustando as distâncias para capturar a densidade local.
- 2 Constrói uma **árvore de clusterização** com base nas distâncias (CDHI).
- 3 Simplifica a árvore, removendo os pontos com baixa densidade.
- 4 Determina clusters estáveis em diferentes níveis hierárquicos.



Dendrograma



■ Áreas comuns de aplicação:

- **Análise de Dados Geoespaciais:** Para identificar regiões densamente povoadas ou áreas de interesse em dados geográficos.
- **Segmentação de Clientes:** Para agrupar clientes com comportamentos ou características semelhantes, auxiliando em estratégias de marketing direcionadas.
- **Detecção de Padrões em Grandes Conjuntos de Dados:** Para descobrir estruturas ocultas ou padrões em grandes volumes de dados, como em análises de redes sociais ou dados financeiros.

Parâmetros Seleccionados Arbitrariamente:

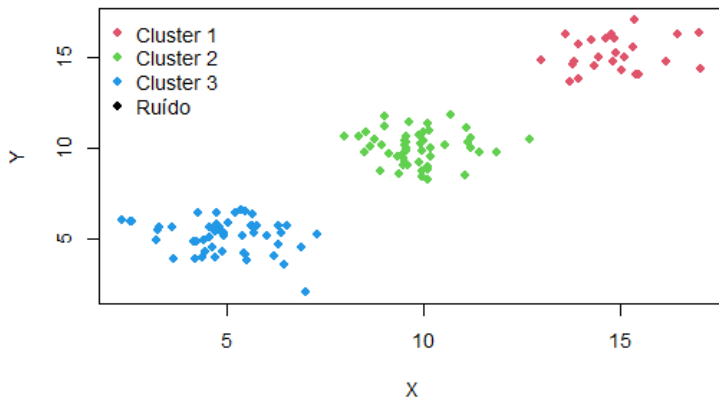
- **MinPts** = 5: Define o número mínimo de pontos necessários para formar um cluster.
- **Observação:** A escolha do parâmetro minPts afeta a quantidade de pontos considerados como ruídos.

Resultados Esperados:

- Identificar três clusters bem definidos.
- Se houver, considerar os pontos dispersos como ruídos.
- Permitir clusters de diferentes densidades.

Exemplo Didático: HDBSCAN

Clusters Identificados pelo HDBSCAN



- O HDBSCAN é uma ferramenta robusta e eficiente para análise de dados mais complexos.
- Oferece uma maior flexibilidade em relação ao DBSCAN, sendo ideal para datasets com formas ou densidades variáveis.
- Excelente para cenários onde a detecção de ruído é crítica.

O que é OPTICS?

- **OPTICS (Ordering Points To Identify the Clustering Structure)** é um algoritmo de clusterização baseado em densidade, assim como o DBSCAN e o HDBSCAN.
- Foi projetado para lidar com dados de densidade variável.
- Não define diretamente os clusters, mas permite derivá-los posteriormente.

Vantagens do OPTICS

- **Sem necessidade de um valor fixo de ε :** O DBSCAN exige que o usuário defina um único valor de ε , que determina a distância máxima para definir uma vizinhança. O OPTICS, por outro lado, ordena os pontos, permitindo a formação de clusters com densidades variadas.
- **Geração de uma “Ordem de Acessibilidade”:** Além de encontrar clusters, o OPTICS também gera um gráfico chamado "reachability plot", que mostra a estrutura hierárquica dos clusters nos dados.
- **É mais flexível que o DBSCAN:** O OPTICS pode extrair clusters de diferentes escalas de densidade, algo que o DBSCAN não consegue fazer com um único parâmetro ε .



- **Complexidade computacional:** O OPTICS tem um desempenho melhor que o DBSCAN em termos de flexibilidade e em lidar com diferentes densidades de clusters, mas pode ser computacionalmente mais caro e lento para conjuntos de dados grandes, pois o algoritmo ordena os pontos e exige a construção de uma estrutura de ordem dos dados antes de formar os clusters.

- **Sensibilidade ao Parâmetro minPts :** Embora o OPTICS seja considerado mais robusto que o DBSCAN em relação ao parâmetro eps , ele ainda é sensível ao valor de minPts . Se esse valor for muito pequeno, o algoritmo pode dividir clusters grandes ou até identificar ruídos erroneamente. Se for muito grande, o OPTICS pode não detectar clusters menores.

Principais conceitos:

- 1 A **distância do núcleo** (core distance) de um ponto é a menor distância necessária para incluir o número mínimo de pontos (*minPts*) dentro do seu raio.
- 2 A **distância de alcançabilidade** (reachability distance) mede a densidade relativa de um ponto com relação a outro, sendo usado para ordenar os pontos de forma que os clusters possam ser identificados.

Parâmetros principais:

- ϵ : distância máxima para considerar um ponto vizinho (opcional).
- *minPts*: número mínimo de pontos necessários para formar um cluster.

■ Áreas comuns de aplicação:

- Análise de Dados Espaciais: por exemplo, agrupamento de cidades com base na população.
- Segmentação de Clientes: Para agrupar clientes com comportamentos ou características semelhantes, auxiliando em estratégias de marketing direcionadas.
- Detecção de Anomalias: Identificando pontos isolados em grandes bases de dados.

Parâmetros Seleccionados Arbitrariamente:

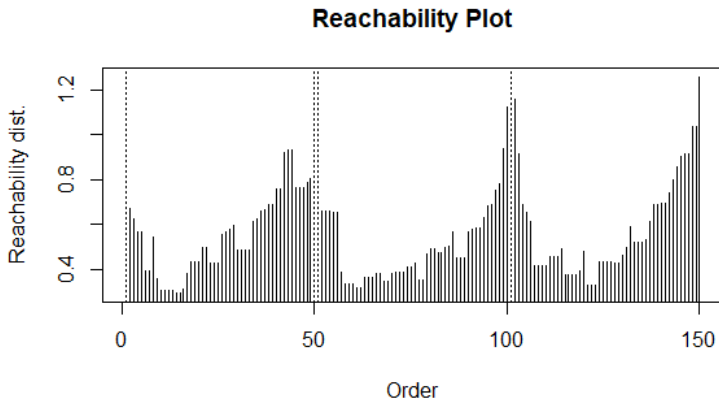
- $Eps = 1.5$: Define o raio de vizinhança.
- $MinPts = 5$: Define o número mínimo de pontos necessários para formar um cluster.

Resultados Esperados:

- Gerar três clusters com diferentes densidades
- Analisar a distância de acessibilidade.
- Usar o OPTICS para ordenar os pontos e identificar clusters com limiares variados.

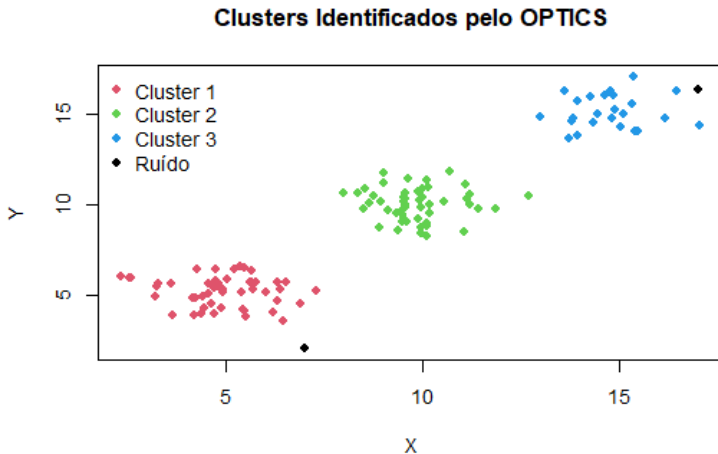
Exemplo Didático: OPTICS

- Os vales (regiões mais baixas) representam clusters densos, enquanto os picos indicam áreas de baixa densidade ou separação entre clusters.
- As linhas tracejadas verticais indicam possíveis divisões naturais dos clusters.





Exemplo Didático: OPTICS


- Diferente do HDBSCAN, aqui há ruídos explícitos .





- O OPTICS é uma excelente alternativa ao DBSCAN, especialmente quando há clusters de diferentes densidades. Ele permite encontrar estruturas hierárquicas nos dados e ajusta automaticamente o parâmetro ϵ , tornando-o mais flexível.

 HAHSLER, Michael; PIEKENBROCK, Cameron; DANKO, Luke. dbscan: Fast Density-based Clustering with R. *Journal of Statistical Software*, v. 91, n. 1, p. 1–30, 2019.

 ESTER, Martin; KRIEGEL, Hans-Peter; SANDER, Jörg; XU, Xiaowei. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*. Portland, Oregon: AAAI Press, 1996. p. 226–231.

 ESTER, Martin; KRIEGEL, Hans-Peter; SANDER, Jörg; XU, Xiaowei. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*. Portland, Oregon: AAAI Press, 1996. p. 226–231.

 ANKERST, Mihael; BREUNIG, Markus M.; KRIEGEL, Hans-Peter; SANDER, Jörg. OPTICS: Ordering Points to Identify the Clustering Structure. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*. Philadelphia, PA: ACM, 1999. p. 49–60.

 HAHSLER, Michael; PIEKENBROCK, Cameron; DANKO, Luke. dbscan: Fast Density-based Clustering with R. *Journal of Statistical Software*, v. 91, n. 1, p. 1–30, 2019.

