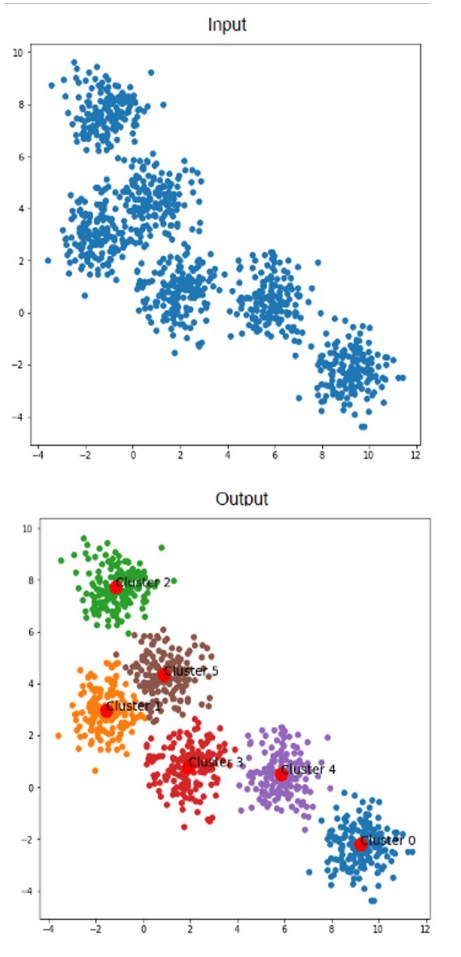
Clusterização

Sistemas Inteligentes

Clusterização

- Clusterização é uma técnica que agrupa pontos de dados similares, tal que pontos de um mesmo grupo são mais similares entre si do que com pontos de outros grupos.
- O grupo de pontos de dados similares é chamado de cluster.



Tipos de Clusterização

Existem quatro principais tipos de técnicas de clusterização:

- **Particionamento**
- **Densidade**
- **Hierárquica**
- **Fuzzy**

Particionamento – K-Means

1. Primeiro, é escolhida a quantidade de classes / grupos, que têm seus respectivos pontos centrais aleatoriamente posicionados.
2. Cada ponto de dados é classificado calculando a distância entre ele e cada centro do grupo e, em seguida, ele é classificado no grupo cujo centro está mais próximo dele.
3. Com base na classificação dos pontos, é recalculado o centro do grupo tomando a média de todos os valores do grupo.
4. Repita os passos 2 e 3 para um determinado número de iterações ou até que os centros do grupo não mudem muito entre as iterações.
   - Você também pode optar por inicializar aleatoriamente os centros de grupo algumas vezes e, em seguida, selecionar a execução que parece ter fornecido os melhores resultados.

Particionamento – K-Means

- Vantagem:
  - Possui complexidade linear O(n).
- Desvantagem:
  - É necessário definir quantos grupos / classes existem.
    - Isso nem sempre é trivial e, idealmente, seria interessante se ele descobrisse para nós, porque o objetivo é obter alguns insights dos dados.
  - Ele começa escolhendo aleatoriamente os centros dos clusters e, portanto, pode produzir resultados de clustering diferentes em execuções diferentes do algoritmo.
    - Assim, os resultados podem não ser repetíveis e carecem de consistência. Outros métodos de cluster são mais consistentes.

Particionamento – K-Means

Algoritmo de Lloyd

- Fácil implementação
- Estratégia Gulosa
- Refinamento Iterativo
- Centróides iniciais podem ser escolhidos aleatoriamente
  - Estocástico – Não determinístico
  - Nem sempre gera o mesmo resultado de saída
- Um ponto dentro do grupo significa que ele está mais perto do centro do grupo do que de qualquer outro grupo

Particionamento – K-Means

Algoritmo de Lloyd

Entrada: conjunto de dados S e número de clusters K

Saída: K centróides dos grupos

Selecione *K* pontos para utilizar como centróides iniciais *C*

Repita

    Criar K grupos vazios a partir dos centróides *C*

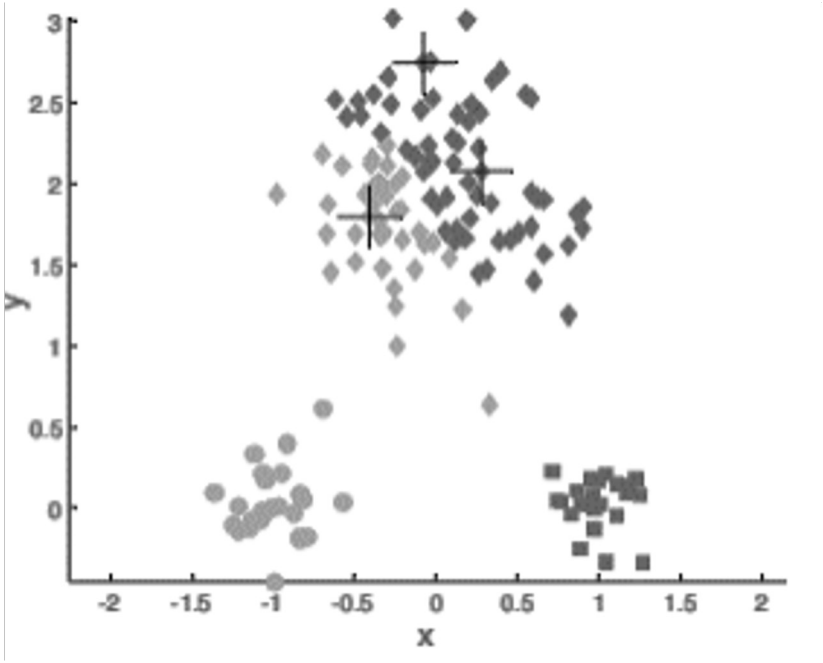
    Adicionar cada dado de *S* ao grupo com centróide mais próximo

    Recalcular cada centróide de *C* pela média dos dados do grupo

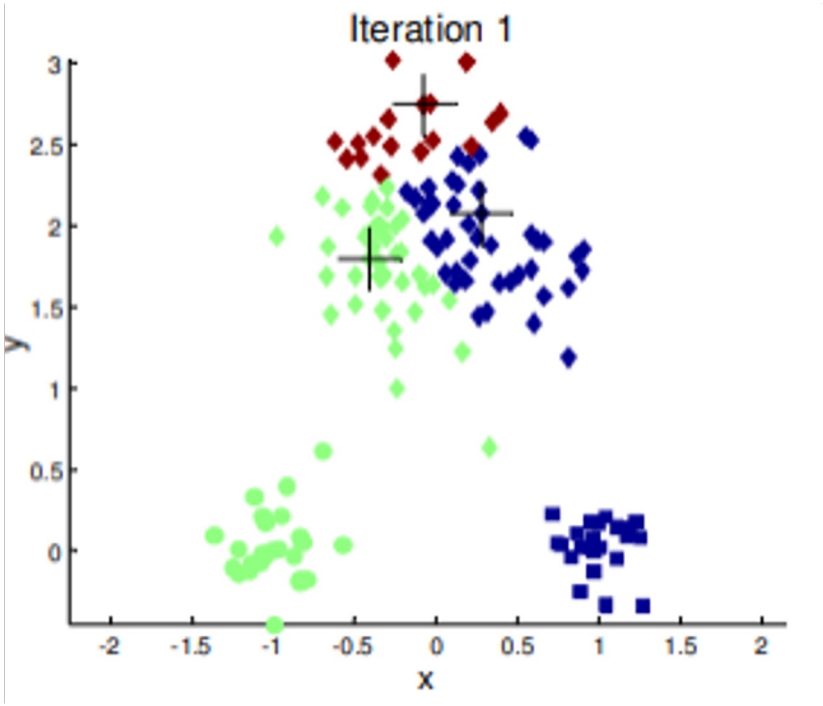
Enquanto os centróides mudarem

Retorne os centróides *C*

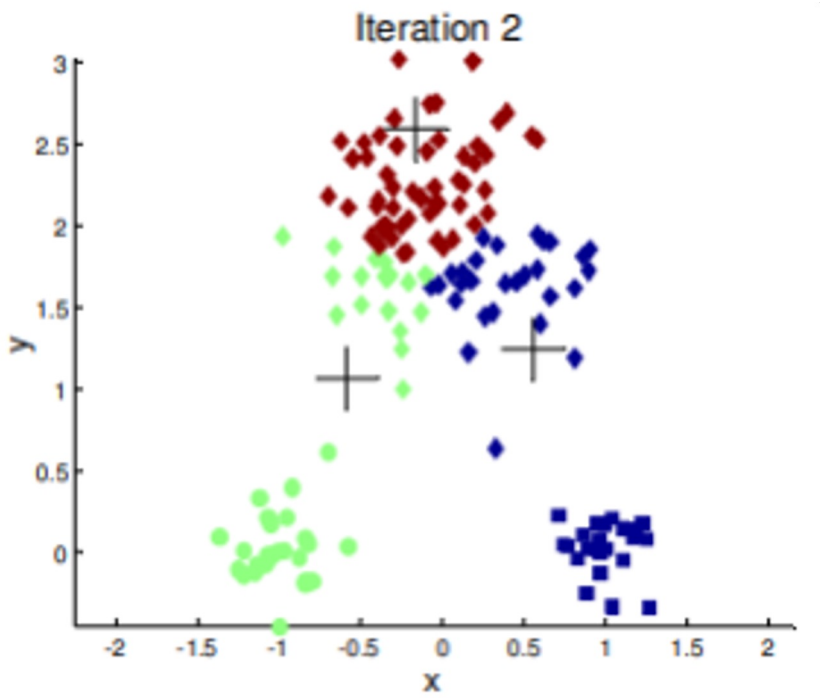
Particionamento – K-Means



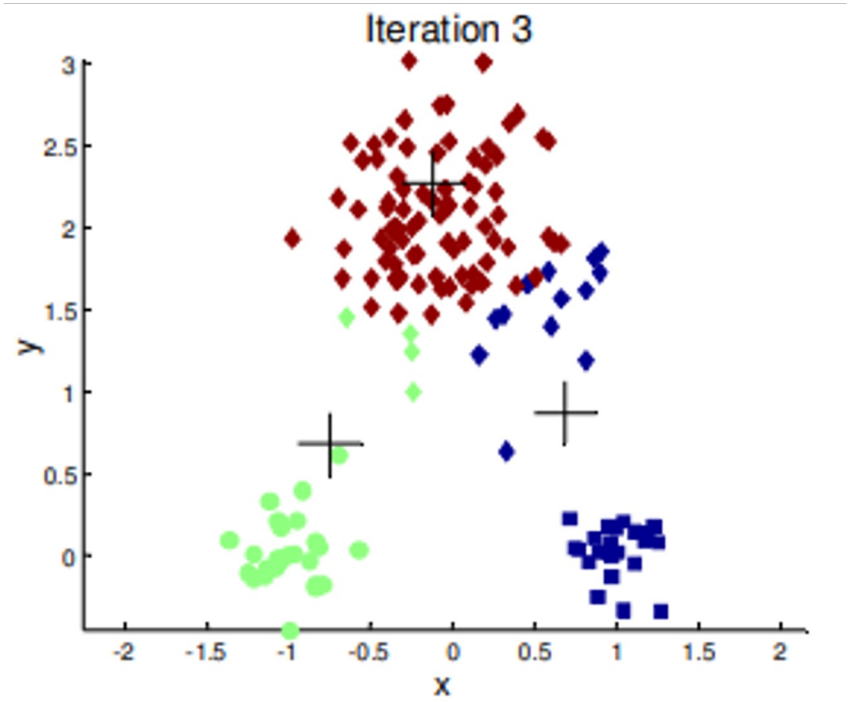
Particionamento – K-Means



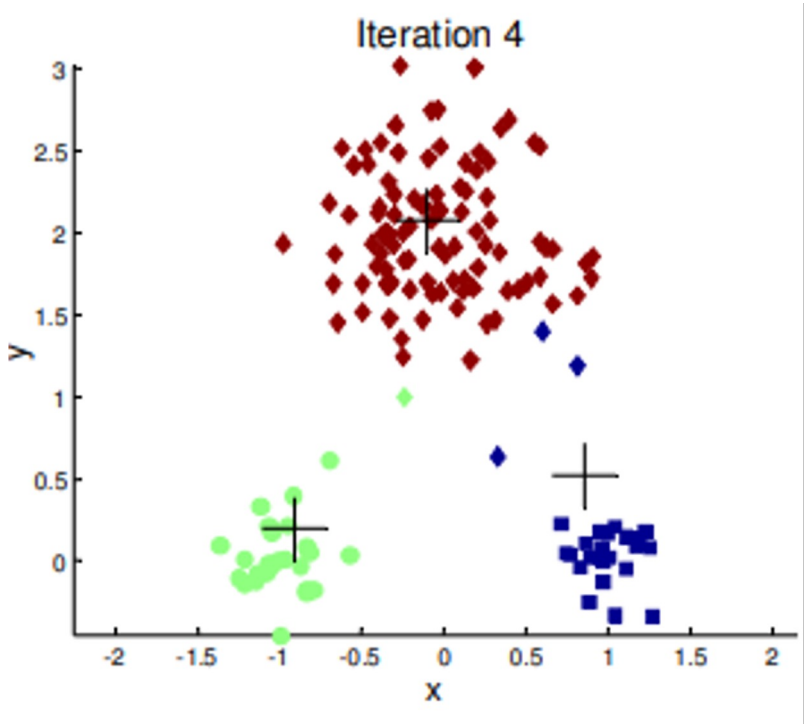
Particionamento – K-Means



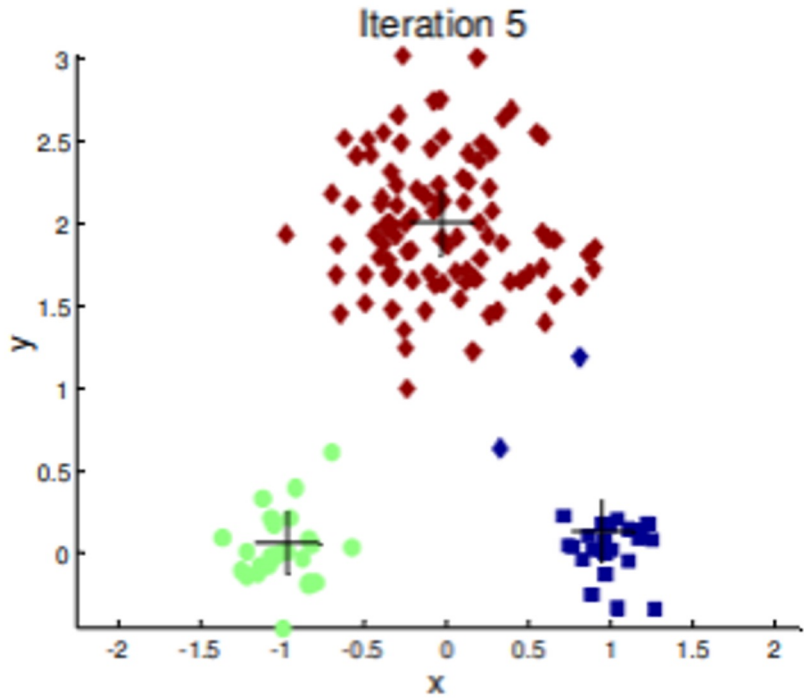
Particionamento – K-Means



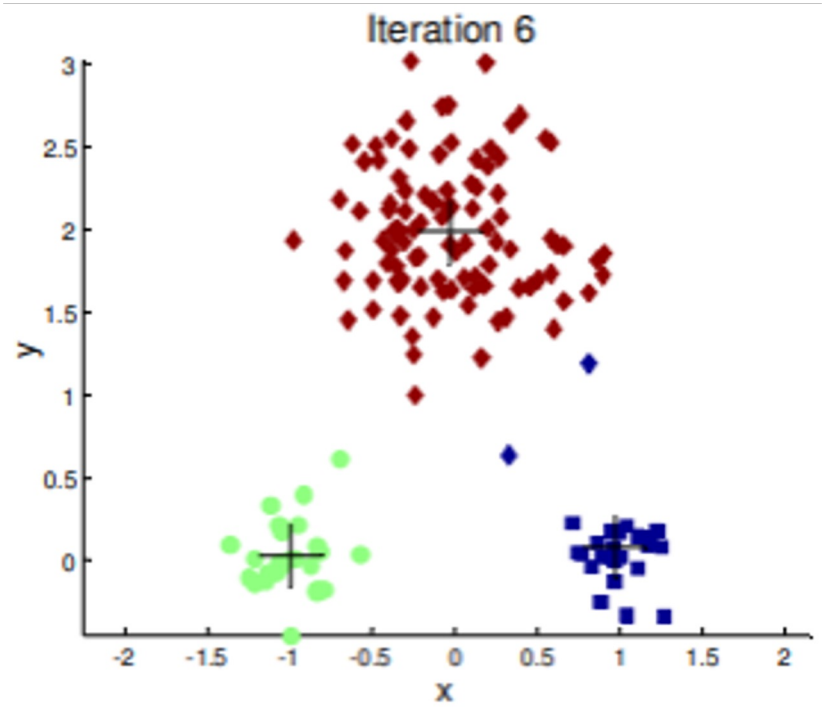
Particionamento – K-Means



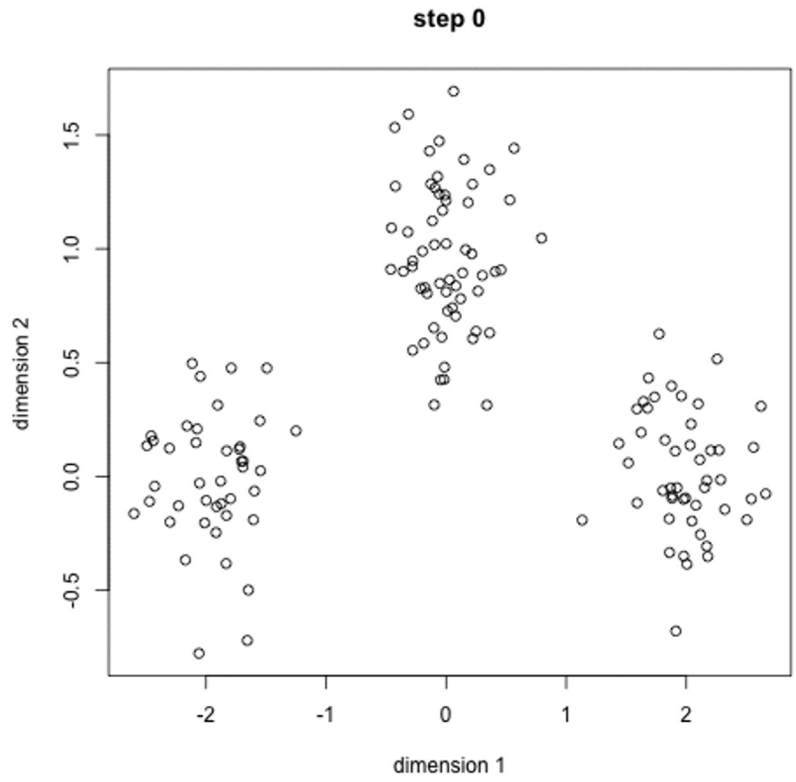
Particionamento – K-Means



Particionamento – K-Means



Clusterização por Particionamento



Particionamento – K-Means

- A convergência geralmente é rápida (número de iterações é pequeno)
- O critério de parada pode ser substituído por pouca diferença

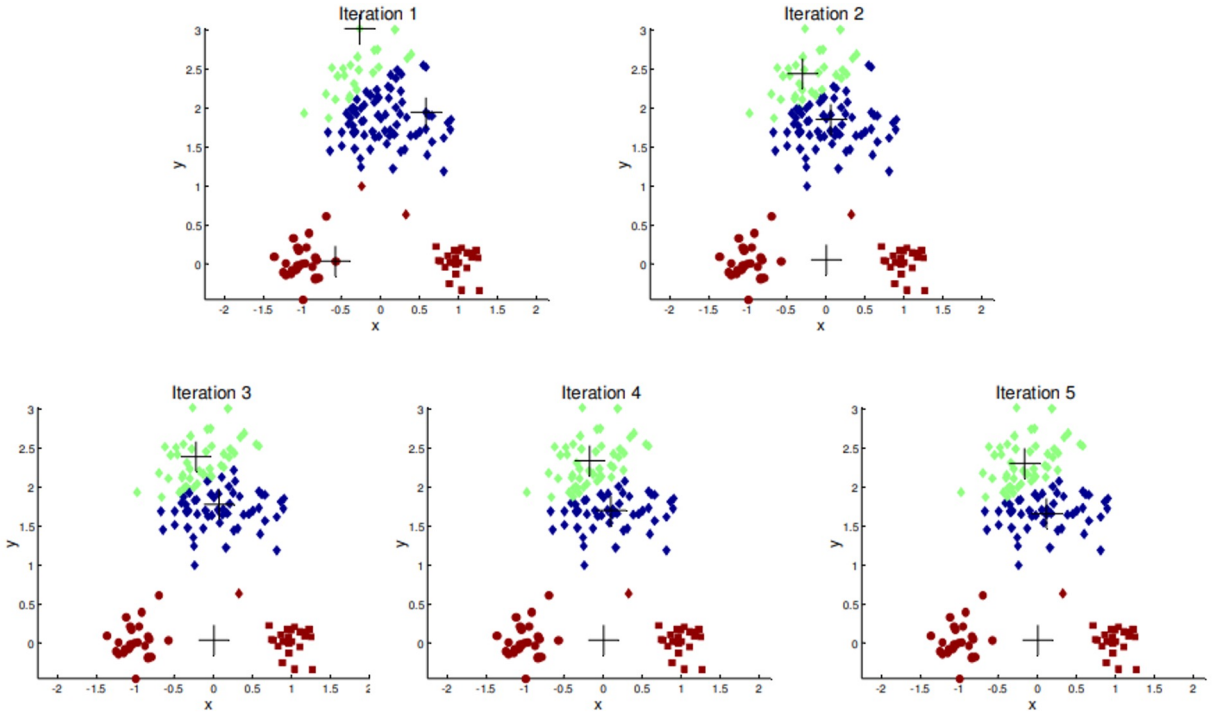
- Complexidade de tempo O(n * K * I * d)
  - n = número de instâncias
  - K = número de grupos
  - I = número de iterações
  - d = dimensão do dado (complexidade de comparação)

Particionamento – K-Means

- Possíveis problemas
  - Grupos de tamanho diferentes
  - Grupos de densidades diferentes
  - Formas não globulares
  - Pouco tolerante a ruídos
  - Má escolha dos centróides iniciais

Particionamento – K-Means

- Má escolha dos centróides iniciais



Determinação do Número Ótimo de Clusters

- Determinar o número ótimo de clusters de um conjunto de dados é fundamental para a aplicação de algoritmos de particionamento.
- Entretanto, não há uma forma única para se responder esta pergunta.
- A definição do número ótimo de clusters é subjetiva e depende do método usado para medir a similaridade e os parâmetros utilizados no particionamento.
- Uma solução simples consiste em inspecionar o dendrograma produzido usando a clusterização hierárquica para ver se ele sugere um número particular de clusters.

Determinação do Número Ótimo de Clusters

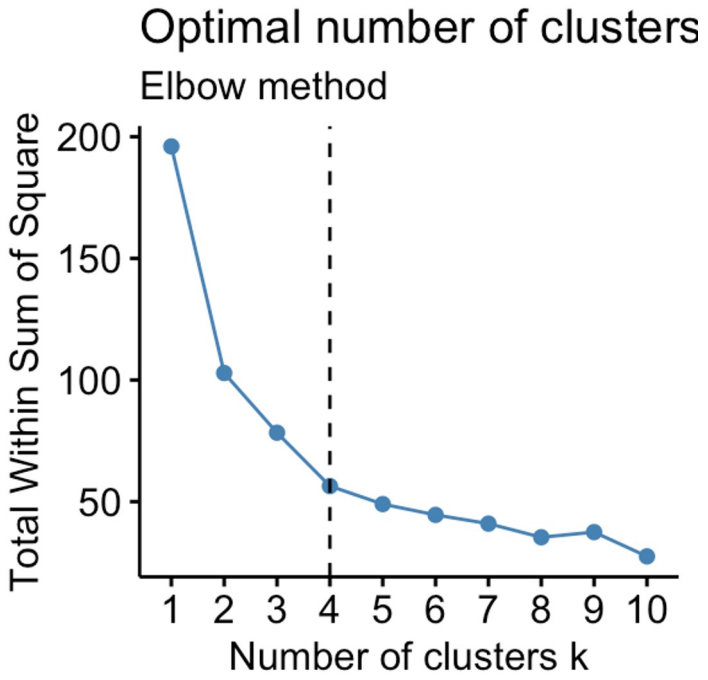
- Os métodos para determinação do número ótimo de clusters podem ser divididos em:
  - **Métodos diretos**: consistem em otimizar um critério, como a soma dos quadrados das distâncias dentro do cluster ou a silhueta média. Com exemplo tem-se os métodos *elbow* e *silhouette*.
  - **Métodos de teste estatístico**: consistem em comparar a evidência com a hipótese nula. Com exemplo tem-se o método *gap statistic*

Determinação do Número Ótimo de Clusters

- Método de cálculo da dispersão (*elbow*):
  - A ideia é selecionar como número de clusters aquele que se adicionado outro cluster afeta pouco a dispersão intracluster.
  - A localização de uma dobra (cotovelo) no gráfico é geralmente considerada como um indicador do número apropriado de grupos.
  - A dispersão pode ser calculada pela soma do erro quadrático ou pela variância.

$$E = \sum_{j=1}^{k} \sum_{x_i \in C_j} d\left(x_i, \bar{x}^{(j)}\right)^2$$

$$var = \sqrt{\frac{1}{n_x} \sum_{j=1}^{k} \sum_{x_i \in C_j} d(x_i, \bar{x}^{(j)})}$$



Determinação do Número Ótimo de Clusters

- *Elbow*

  https://towardsdatascience.com/k-means-clustering-with-scikit-learn-6b47a369a83c

- *Silhouette*

  https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

- *Gap statistic*

  https://medium.com/mlearning-ai/deciding-number-of-clusters-using-gap-statistics-davies-bouldin-index-calinski-harabasz-index-2ce9acfb6118

Particionamento – K-Medoids

- K-Medoids é outro algoritmo de agrupamento similar ao K-Means.
- Entretanto, ao invés de recalcular os pontos centrais do grupo usando a média, ele usa a mediana do grupo.
- Portanto, ele é menos sensível a outliers (devido ao uso da Mediana).
- Mas, ele é muito mais lento para conjuntos de dados maiores, pois é necessário ordená-los a cada iteração para calcular a mediana.

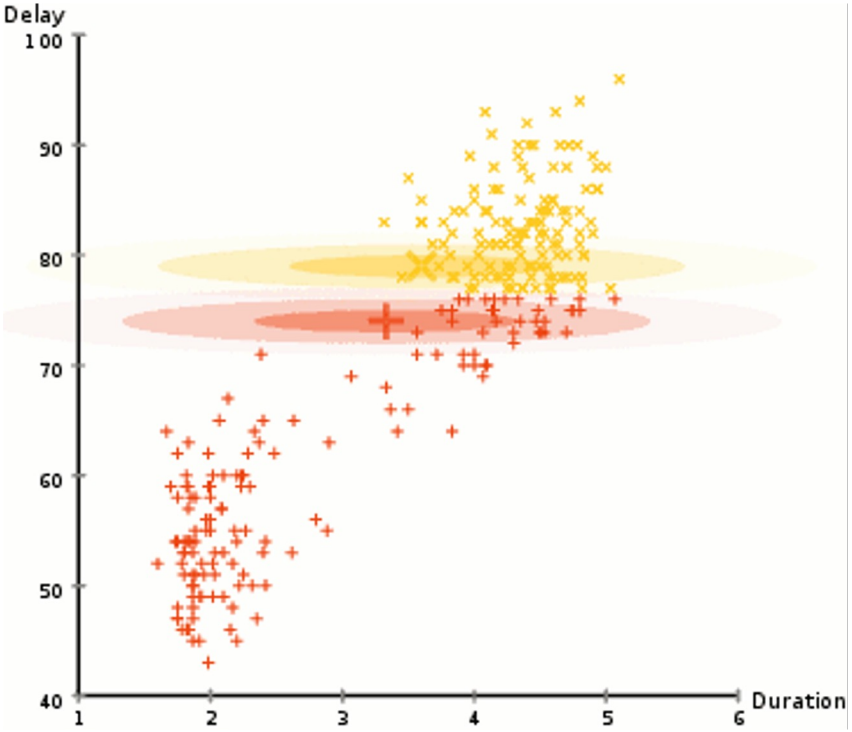
Particionamento – K-Medoids

```
Algoritmo Medoids
Entrada: conjunto de dados S
Saída: o centróide do conjunto

1.  menorCusto = Infinity
2.  menorPonto = Null
3.  Para cada ponto si em S
4.     custoSi = 0
5.     Para cada ponto sj em S
6.       custoSi = custoSi + custo(si, sj)
7.     Se custoSi < menorCusto
8.       menorCusto = custoSi
9.     menorPonto = si
10.  retorna menorPonto
```

Clusterização utilizando Gaussian Mixture Models (GMM)

- Para encontrar os parâmetros do Gaussiano para cada cluster (a média e o desvio padrão), é usado um algoritmo de otimização chamado Expectation – Maximization (EM).
- Portanto, de forma iterativa as elipses gaussianas são ajustadas aos clusters.
- Em seguida, é feito o agrupamento dos pontos usando os GMMs.



Clusterização utilizando Gaussian Mixture Models (GMM)

1. Começa-se selecionando o número de clusters (como o K-Means faz) e inicializando aleatoriamente os parâmetros de distribuição gaussiana para cada cluster.
2. Dadas as distribuições gaussianas para cada cluster, calcula-se a probabilidade de cada ponto de dados pertencer a um cluster específico. Quanto mais próximo um ponto está do centro do Gaussiano, mais provável é que ele pertença a esse agrupamento.
3. Com base nessas probabilidades, calcula-se um novo conjunto de parâmetros para as distribuições gaussianas de modo que maximize as probabilidades de pontos de dados dentro dos clusters.
   - Esses novos parâmetros são calculados usando uma soma ponderada das posições dos pontos de dados, em que os pesos são as probabilidades do ponto de dados pertencer a esse cluster específico.
4. As etapas 2 e 3 são repetidas iterativamente até a convergência, onde as distribuições não mudam muito de iteração para iteração.

Clusterização utilizando Gaussian Mixture Models (GMM)

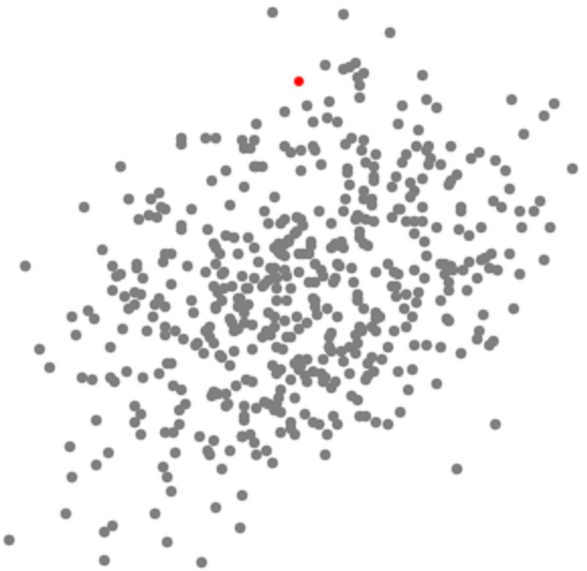
- Vantagens:
  - Os GMMs são muito mais flexíveis em termos de covariância do cluster do que as K-médias; devido ao parâmetro de desvio padrão, os clusters podem assumir qualquer forma de elipse, em vez de ficarem restritos a círculos.
  - Uma vez que os GMMs usam probabilidades, eles podem ter vários clusters por ponto de dados. Portanto, se um ponto de dados está no meio de dois clusters sobrepostos, podemos simplesmente definir sua classe dizendo que ele pertence X por cento à classe 1 e Y por cento à classe 2. Ou seja, os GMMs suportam associação mista.

Clusterização utilizando Gaussian Mixture Models (GMM)

- GMM assume que os dados têm distribuição Gaussiana;
  - Que é menos restritiva do que as médias circulares usadas pelo K-Means.
- Dessa forma, têm-se dois parâmetros para descrever a forma dos clusters: a média e o desvio padrão.
- Tomando um exemplo em duas dimensões, isso significa que os clusters podem assumir qualquer tipo de forma elíptica (uma vez que temos um desvio padrão nas direções x e y).
- Assim, cada distribuição gaussiana é atribuída a um único cluster.

Densidade – Clusterização Mean-Shift

- O agrupamento mean-shift é um algoritmo baseado em janela deslizante que tenta encontrar áreas densas de pontos de dados.
- Seu objetivo é localizar os centróides de cada grupo atualizando os candidatos pela média dos pontos dentro da janela deslizante.
- Essas janelas candidatas são então filtradas em um estágio de pós-processamento para eliminar duplicatas próximas, formando o conjunto final de centróides e seus grupos correspondentes.

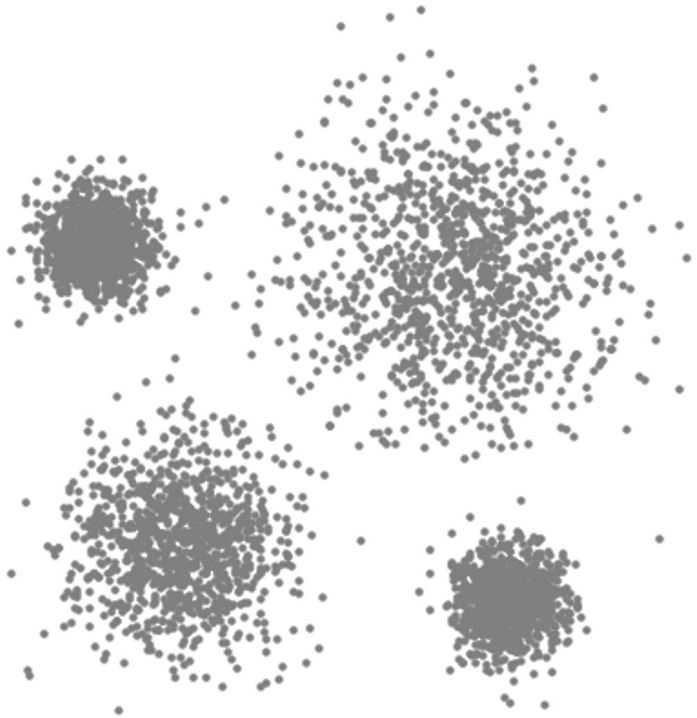


Densidade – Clusterização Mean-Shift

1. O algoritmo começa com uma janela deslizante circular de raio r centrada em um ponto C (selecionado aleatoriamente). Ele é um algoritmo de hill-climbing que desloca o cluster iterativamente para uma região de densidade mais alta em cada etapa até a convergência.
2. A cada iteração, a janela deslizante é deslocada para uma região de maior densidade, deslocando o centróide para a média dos pontos dentro da janela. Portanto, ao mudar para a média dos pontos na janela, ele se moverá gradualmente em direção às áreas de maior densidade de pontos.
3. O deslocamento da janela deslizante é feito de acordo com a média até que não haja uma direção na qual uma mudança possa acomodar mais pontos dentro do cluster.
4. As etapas de 1 a 3 são executadas com muitas janelas deslizantes até que todos os pontos fiquem dentro de uma janela. Quando várias janelas deslizantes se sobrepõem, a janela que contém a maioria dos pontos é preservada. Os pontos de dados são então agrupados de acordo com a janela deslizante na qual residem.

Densidade – Clusterização Mean-Shift

- Todo o processo com todas as janelas deslizantes é mostrado ao lado.
- Cada ponto preto representa o centróide de uma janela deslizante e cada ponto cinza é um ponto de dados.



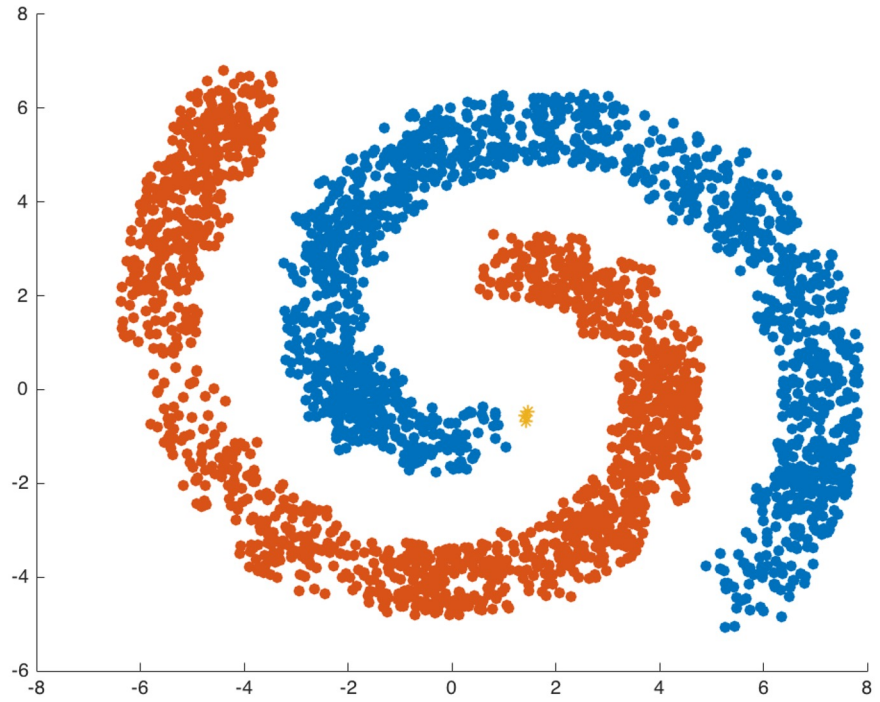
Densidade – DBSCAN

- *Density-based spatial clustering of applications with noise* (DBSCAN)

- Baseado em densidade (vizinho mais próximo, semelhança transitiva):
  - Um grupo é definido por regiões de alta densidade, separados por regiões de baixa densidade
  - Densidade independe do formato dos grupos: podem ser irregulares ou entrelaçados
  - Resiliente à ruídos / *outliers* (regiões de baixa densidade não são relevantes)

- Um ponto dentro do grupo significa que ele está mais perto de outro ponto do grupo do que de qualquer ponto de outro grupo

Densidade – DBSCAN



Densidade – DBSCAN

- Não necessita de número de clusters
- Fácil implementação
- Como definir uma região de alta densidade?
  - Um círculo de raio *D* com pelo menos outros *E* pontos dentro dele.

Densidade – DBSCAN

Algoritmo DBSCAN

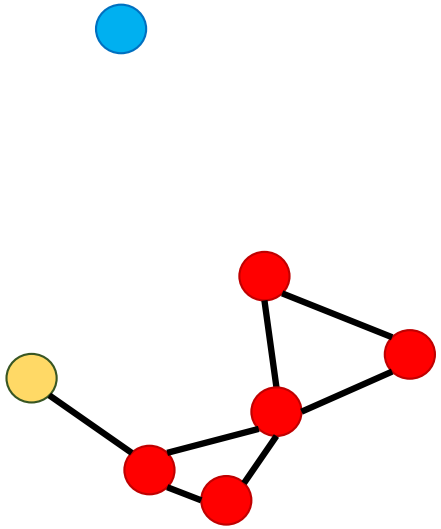
Entrada: conjunto de dados S, tamanho mínimo do cluster *E* e distância máxima intra-cluster *D*

Saída: Grupos criados

1. Enquanto há pontos em *S* que ainda não foi selecionado
2. Criar novo grupo *C* vazio
3. Selecionar um ponto c_i que ainda não foi selecionado
4. Se c_i possui menos de E pontos próximos
5. Considere como ruído e volte ao passo 1
6. Senão
7. Adicione c_i ao grupo C
8. Faça recursivamente o passo 4 para todos os vizinhos de c_i
9. *Retorne os grupos criados*

Densidade – DBSCAN

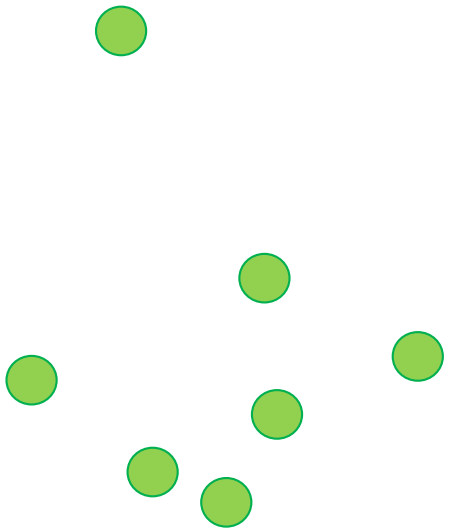
- Vermelho: Núcleo do Grupo
- Amarelo: Borda do Grupo
- Azul: Ruído



MinPts = 2
MaxDIst = 5

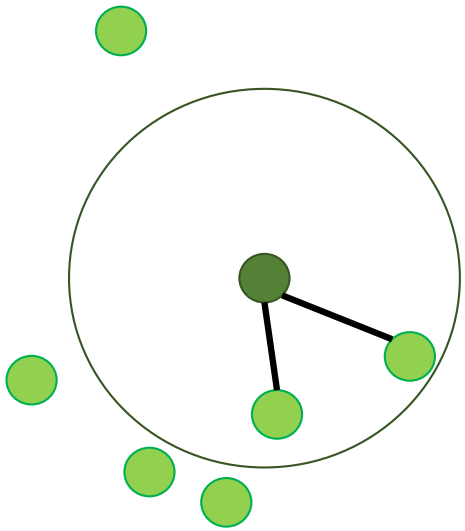
Densidade – DBSCAN

- Verde: A Agrupar
- Vermelho: Núcleo do Grupo
- Amarelo: Borda do Grupo
- Azul: Ruído



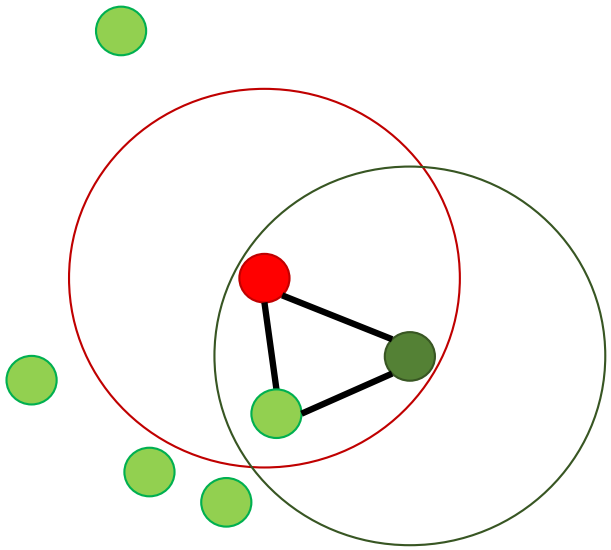
Densidade – DBSCAN

- Verde: A Agrupar
- Vermelho: Núcleo do Grupo
- Amarelo: Borda do Grupo
- Azul: Ruído



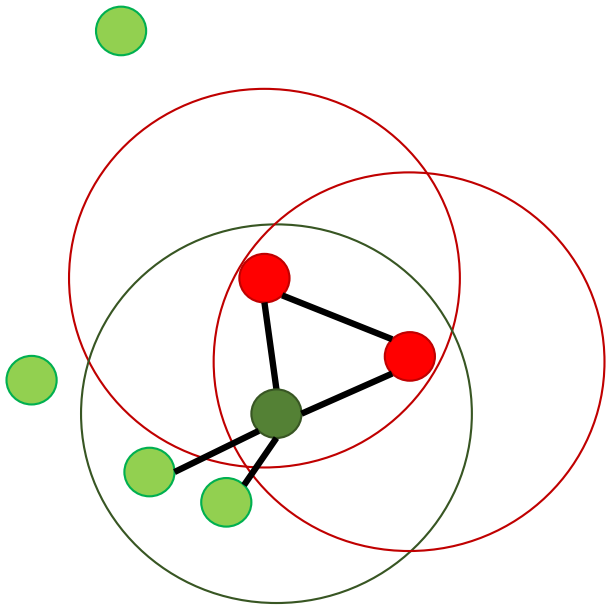
Densidade – DBSCAN

- Verde: A Agrupar
- Vermelho: Núcleo do Grupo
- Amarelo: Borda do Grupo
- Azul: Ruído



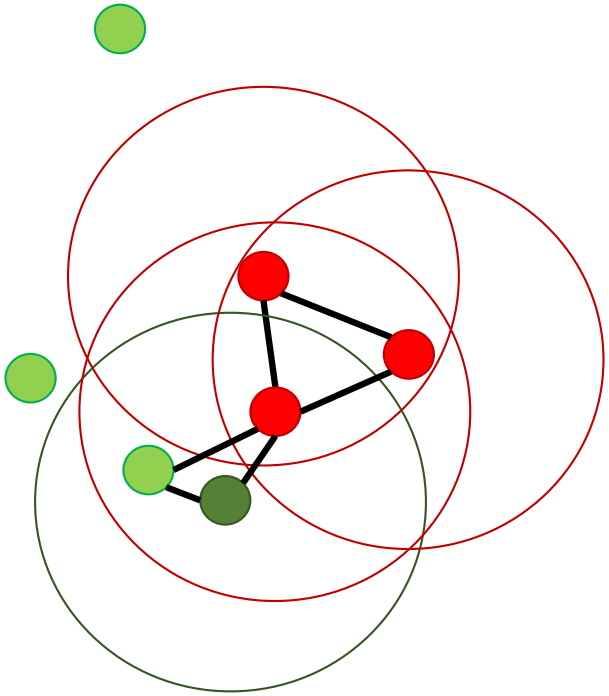
Densidade – DBSCAN

- Verde: A Agrupar
- Vermelho: Núcleo do Grupo
- Amarelo: Borda do Grupo
- Azul: Ruído



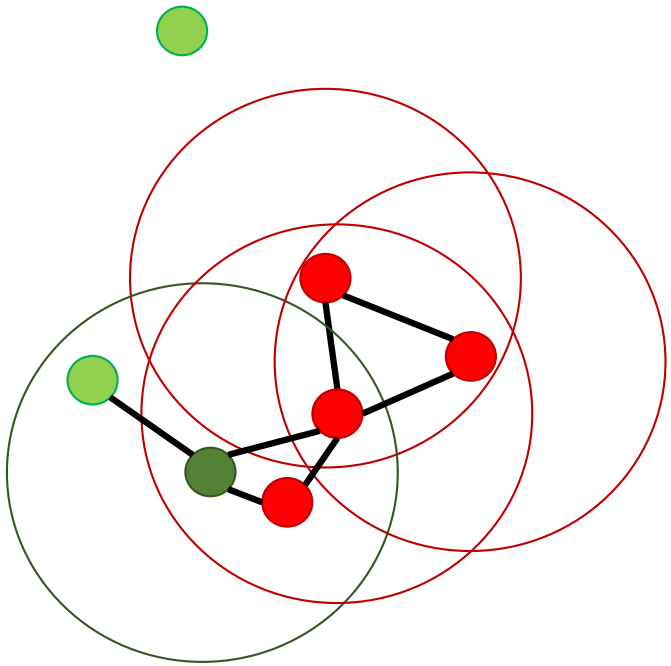
Densidade – DBSCAN

- Verde: A Agrupar
- Vermelho: Núcleo do Grupo
- Amarelo: Borda do Grupo
- Azul: Ruído



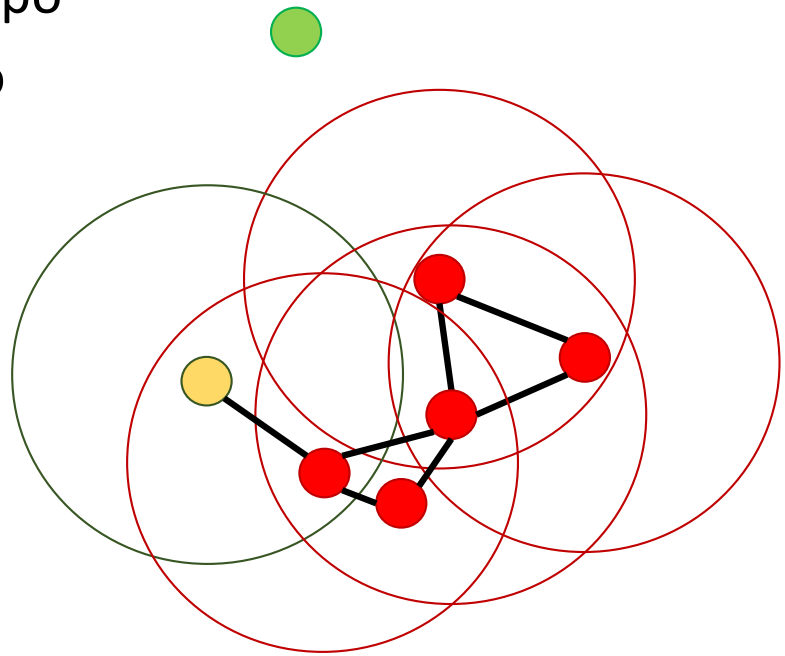
Densidade – DBSCAN

- Verde: A Agrupar
- Vermelho: Núcleo do Grupo
- Amarelo: Borda do Grupo
- Azul: Ruído



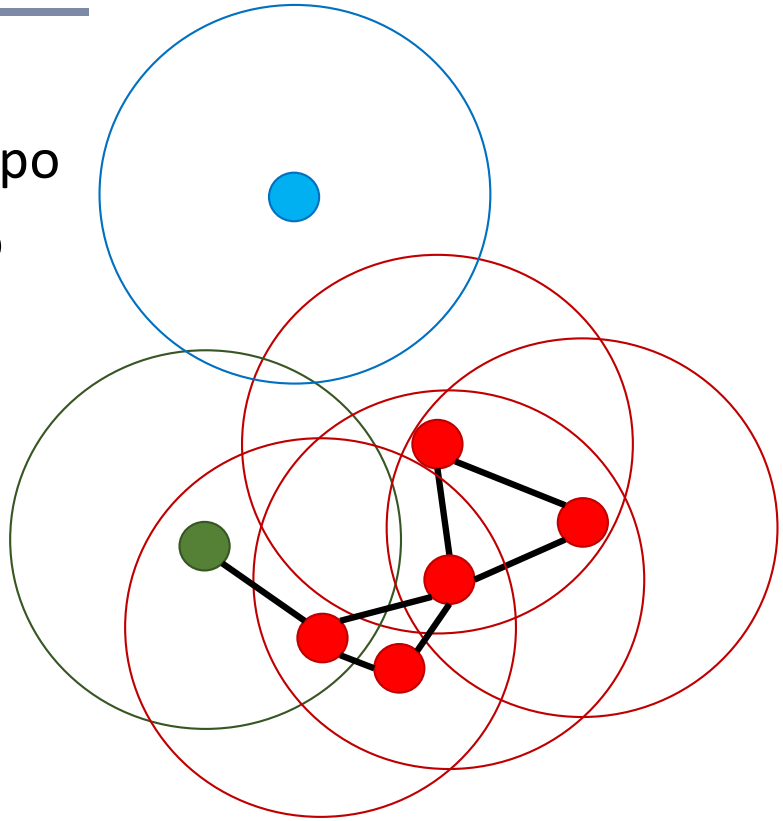
Densidade – DBSCAN

- Verde: A Agrupar
- Vermelho: Núcleo do Grupo
- Amarelo: Borda do Grupo
- Azul: Ruído

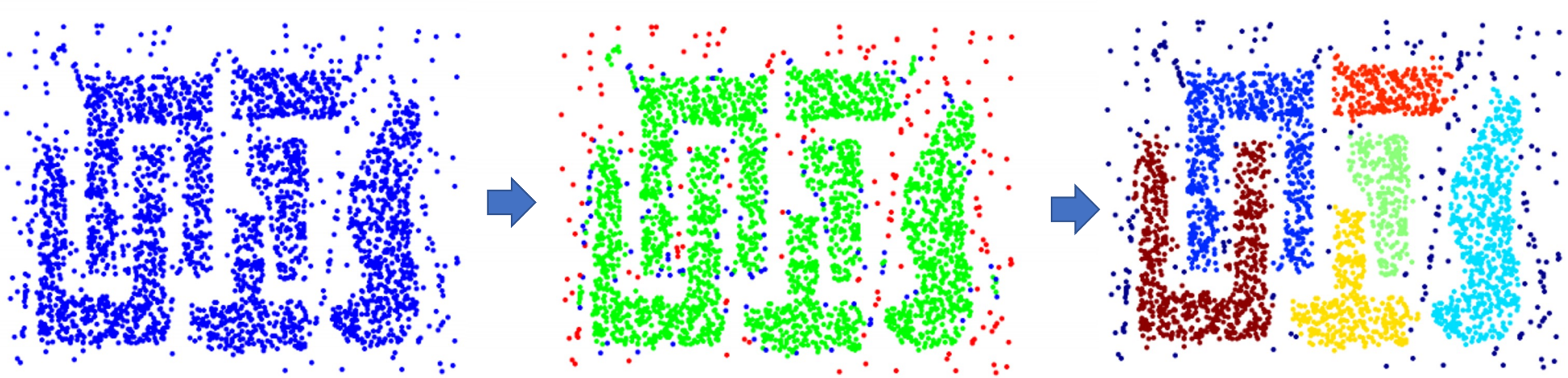


Densidade – DBSCAN

- Verde: A Agrupar
- Vermelho: Núcleo do Grupo
- Amarelo: Borda do Grupo
- Azul: Ruído

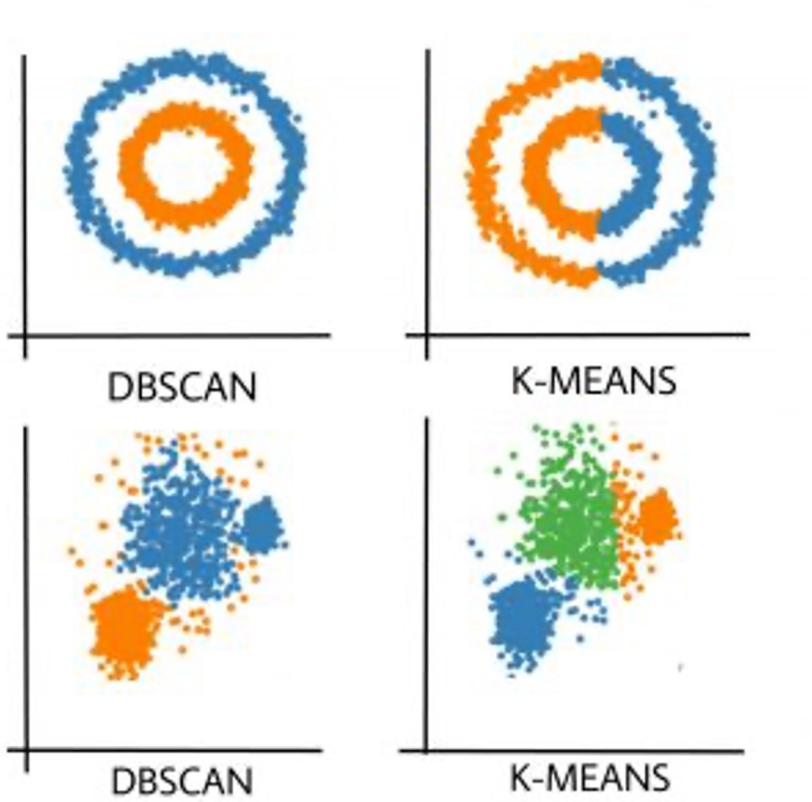


Densidade – DBSCAN



MinPts = 4
MaxDIst = 10

Densidade – DBSCAN



Densidade – DBSCAN

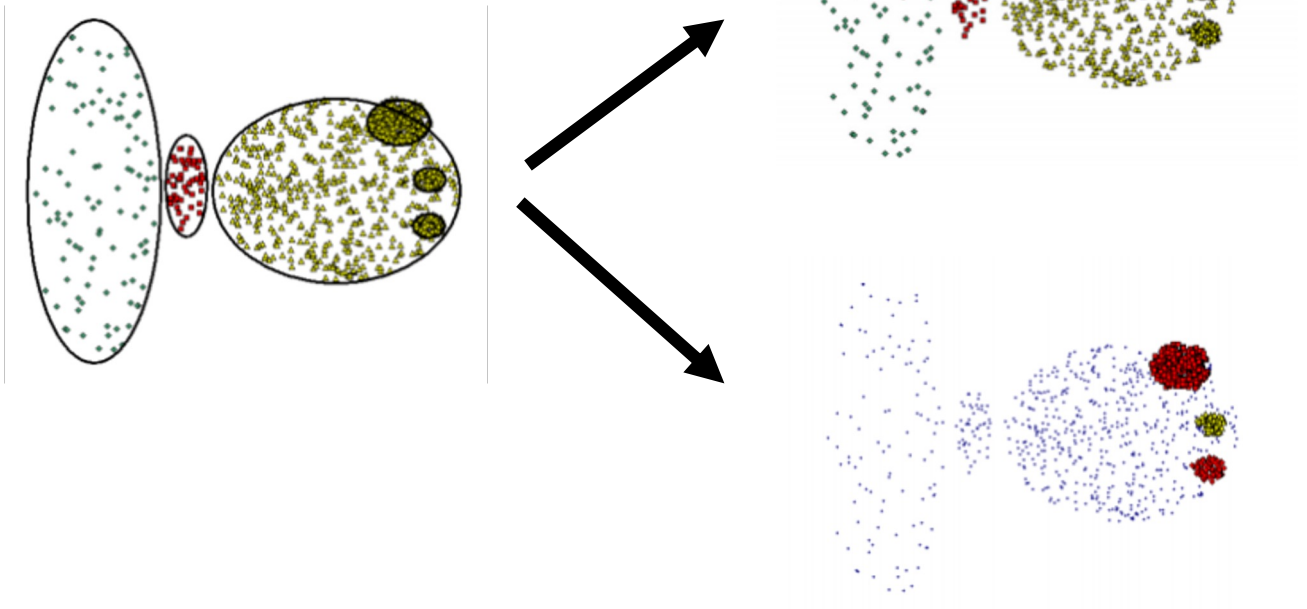
- Determinístico
- Complexidade de tempo O(n * n) para implementação ingênua
  - n = número de instâncias
- Pode ser otimizado para alcançar O(n * log(n)) com as estruturas adequadas

Densidade – DBSCAN

- Vantagens:
  - Não requer um número pré-definido de clusters.
  - Ele também identifica outliers como ruídos
  - Além disso, ele pode encontrar clusters de tamanhos e formatos arbitrários muito bem.
- Desvantagem:
  - Não funciona tão bem quando os clusters são de densidade variável.
    - Isso ocorre porque a configuração do limite de distância ε e minPoints para identificar os pontos de vizinhança irá variar de cluster para cluster quando a densidade varia.
    - Esta desvantagem também ocorre com dados dimensionais muito elevados, uma vez que novamente o limite de distância ε torna-se difícil de estimar.

Densidade – DBSCAN

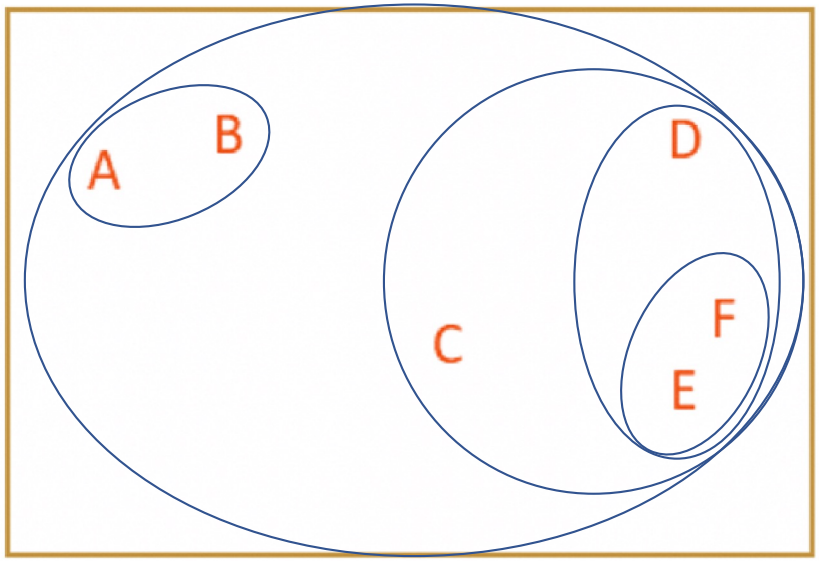
- Possíveis problemas:
  - Densidade Variante



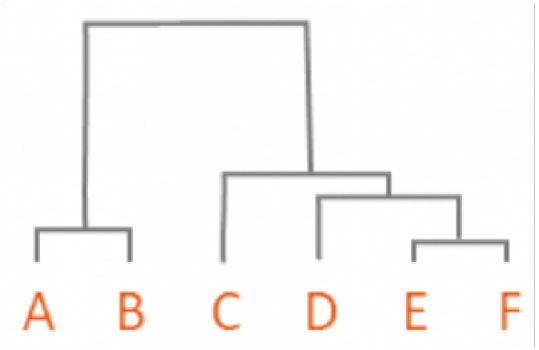
Hierárquico

- Os algoritmos de agrupamento hierárquico se enquadram em 2 categorias: *top-down* ou *bottom-up*.
  - Algoritmos *bottom-up* tratam cada ponto de dados como um único cluster no início e depois mesclam (ou aglomeram) pares de clusters sucessivamente até que todos os clusters tenham sido mesclados em um único cluster que contém todos os pontos de dados.
  - O agrupamento hierárquico *top-down* a raiz da árvore é o único cluster que reúne todas as amostras, sendo as folhas os aglomerados com apenas uma amostra.

Hierárquico



Agrupamentos



Dendrograma

Hierárquico

- Representação por meio de árvore
- Visualizável por meio de um dendrograma
  - Um diagrama de árvore que armazena a sequência de junções/separações
- Contrução gulosa
  - Com alta complexidade de tempo $O(n^3)$
- Necessita de poucos parâmetros:
  - Não necessita de número de clusters antecipadamente
  - Número de cluster pode ser definido por humano ou métrica

Hierárquico

- Abordagem Aglomerativa
  - AGNES – *Agglomerative Nesting*
  - Construção *Bottom-Up*
  - Inicia com *N* grupos e sucessivamente realiza a junção
- Abordagem Divisiva
  - DIANA – *Divisive Analysis*
  - Construção *Top-Down*
  - Inicia com um grupo e sucessivamente realiza divisões

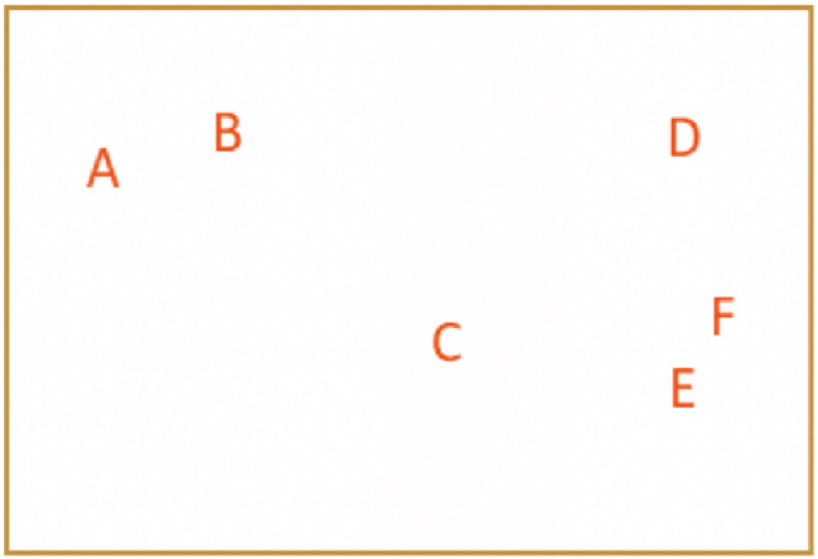
Hierárquico

Algoritmo AGNES
Entrada: conjunto de dados S
Saída: Estrutura hierárquica de agrupamento

1. Converter cada ponto *s* em S em um grupo com apenas o ponto *s*
2. Computar uma matriz de distância/similaridade entre os pontos
3. Enquanto existe mais de um grupo a ser processado
4. Encontrar os grupos *Sa* e *Sb* que possuem a distância mínima
5. Criar um novo grupo *Sc* com os grupos *Sa* e *Sb* como filhos
6. Remover *Sa* e *Sb* da matriz e do conjunto a processar
7. Inserir o grupo *Sc* na matriz e no conjunto a processar
8. Retorna a raiz da árvore gerada (o último grupo restante)

Hierárquico

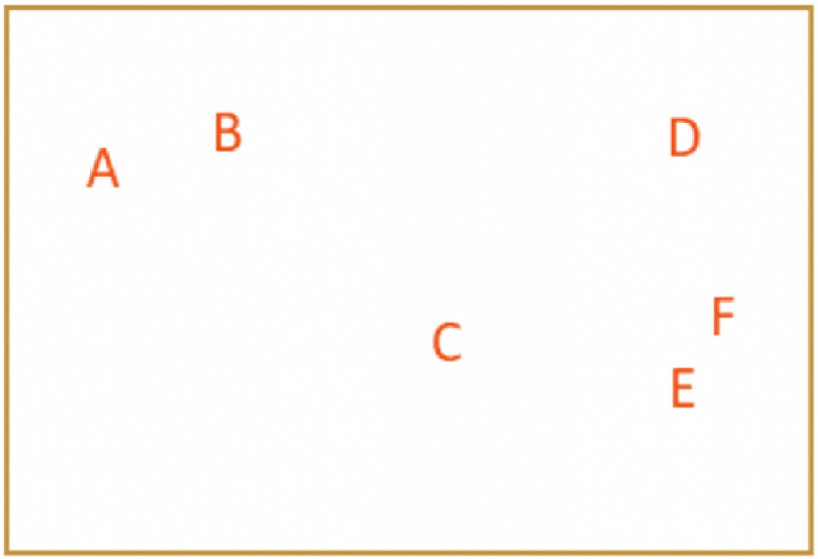
B	16				
C	47	37			
D	72	57	40		
E	77	64	30	31	
F	79	66	35	23	10
	A	B	C	D	E



Matriz de Distância Agrupamentos Dendrograma

Hierárquico

B	16				
C	47	37			
D	72	57	40		
E	77	64	30	31	
F	79	66	35	23	10
	A	B	C	D	E



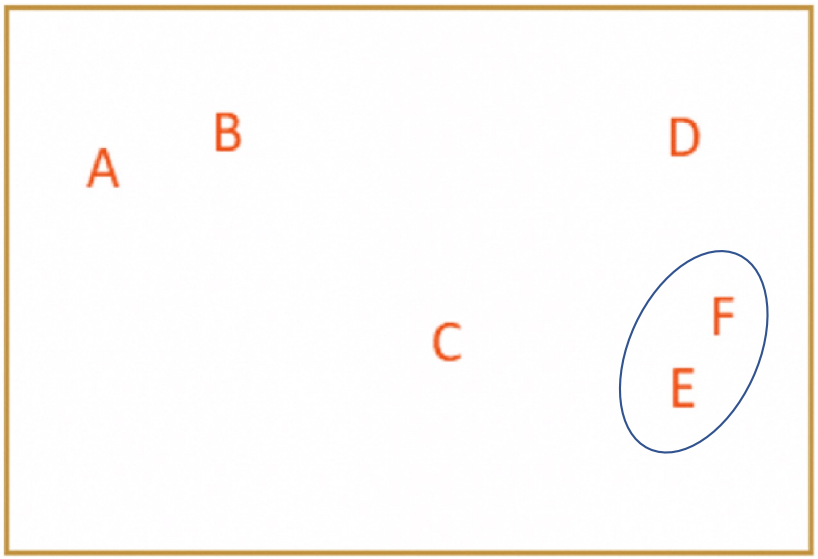
Matriz de Distância

Agrupamentos

Dendrograma

Hierárquico

B	16			
C	47	37		
D	72	57	40	
E,F	77	64	30	23
	A	B	C	D



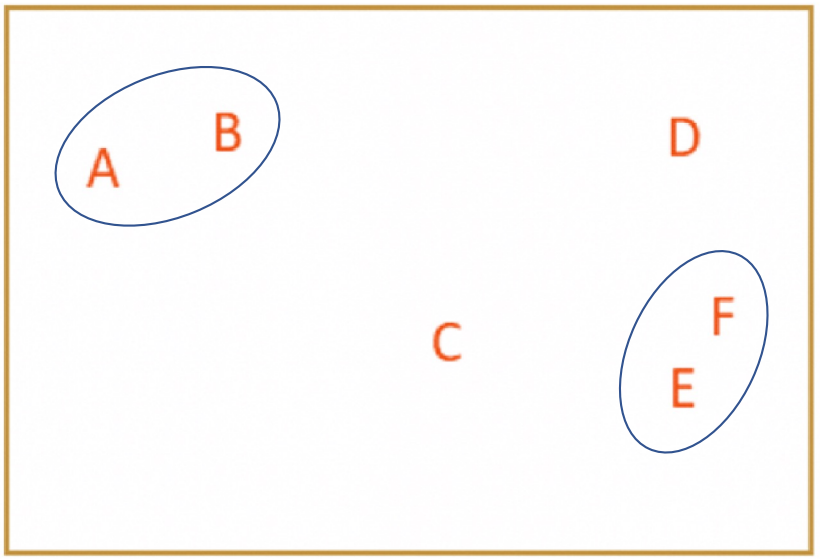
Matriz de Distância

Agrupamentos

Dendrograma

Hierárquico

C	37		
D	57	40	
E,F	64	30	23
	A,B	C	D



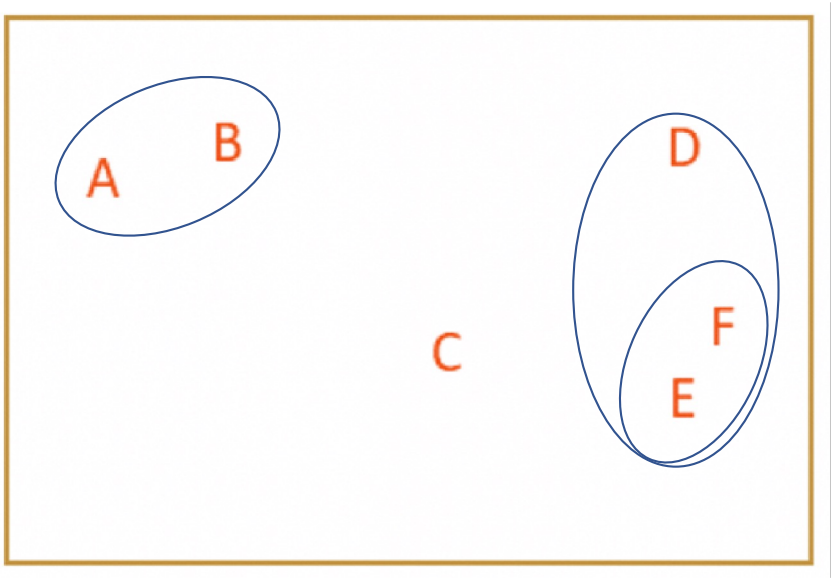
Matriz de Distância

Agrupamentos

Dendrograma

Hierárquico

C	37	
D,E,F	57	30
	A,B	C



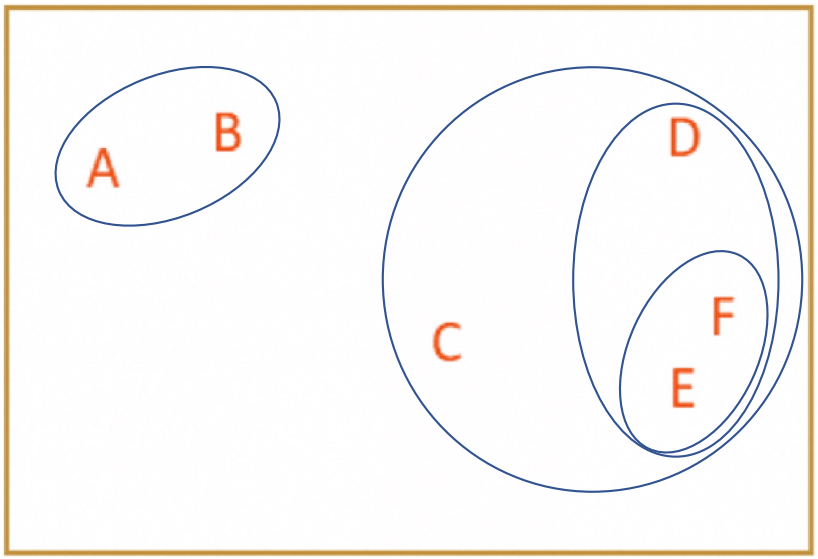
Matriz de Distância

Agrupamentos

Dendrograma

Hierárquico

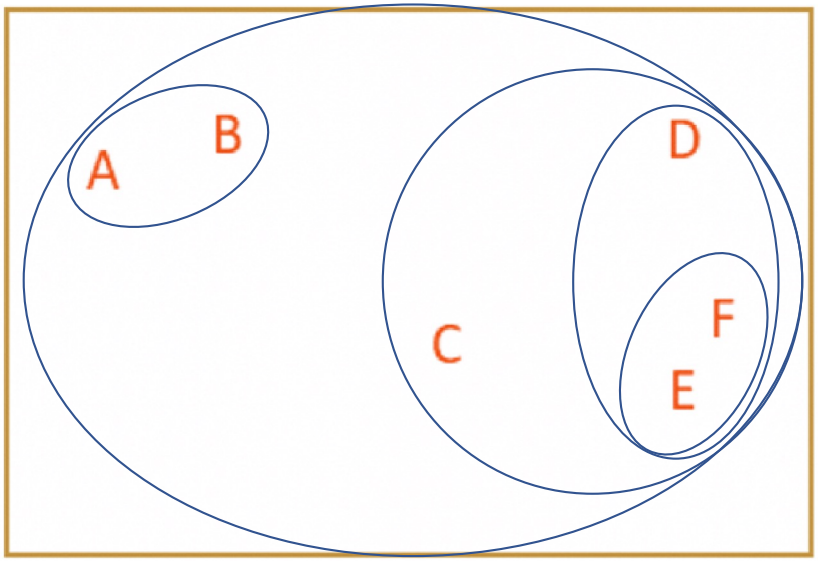
C,D,E,F	37
	A,B



Matriz de Distância Agrupamentos Dendrograma

Hierárquico

A,B,C,D,E,F

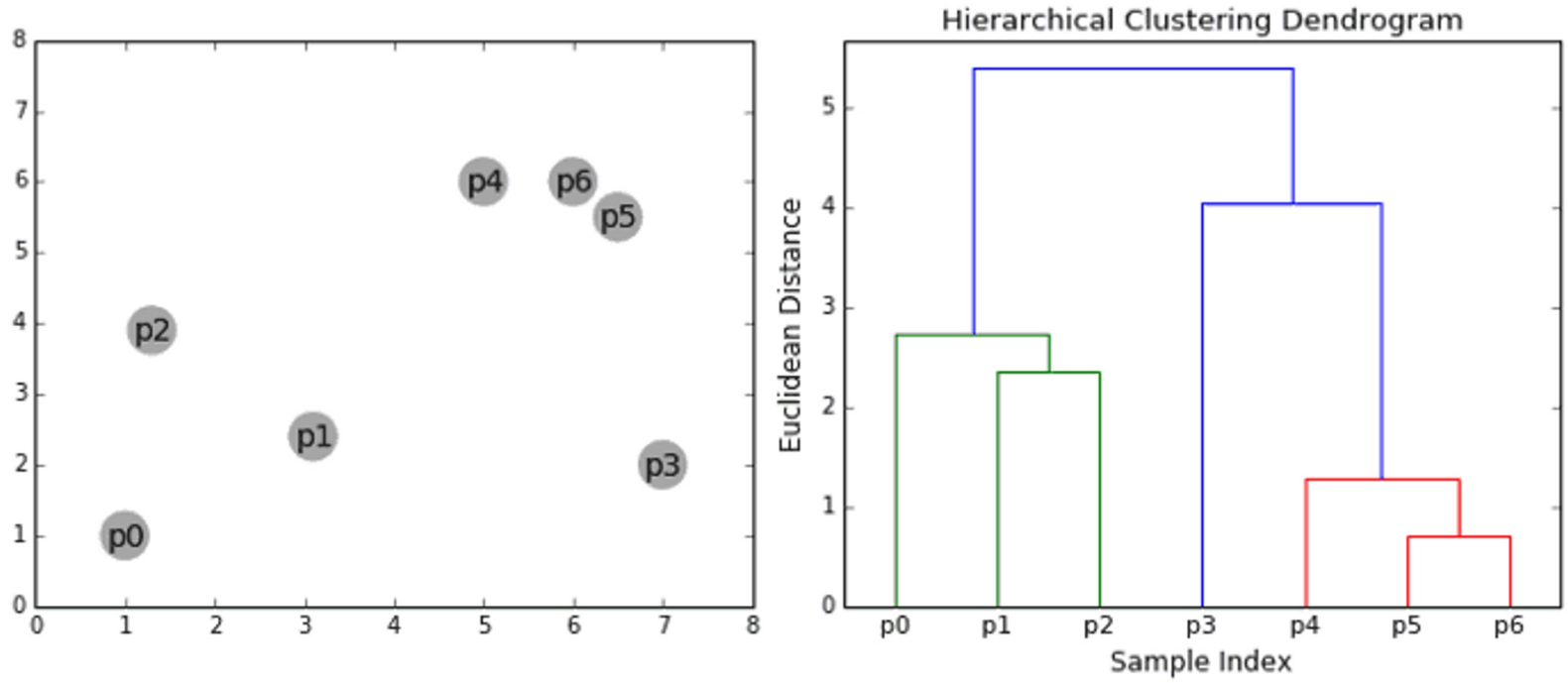


Matriz de Distância

Agrupamentos

Dendrograma

Clusterização Hierárquica

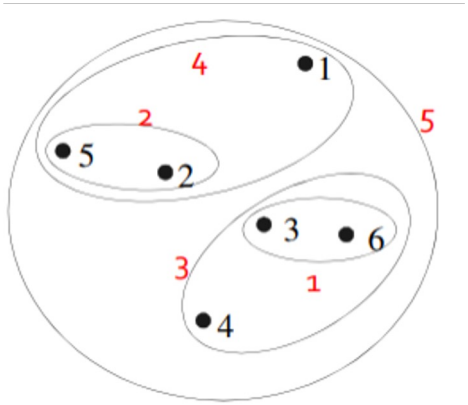


Hierárquico

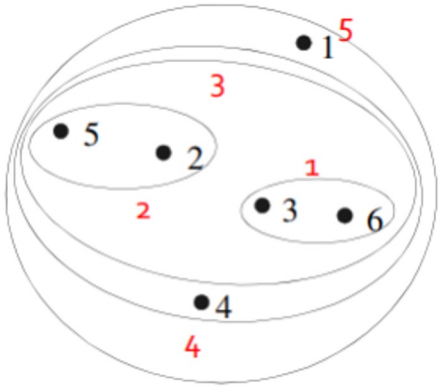
- Modos de junção de grupos
  - Distância Mínima
  - Distância Máxima
  - Distâncias Completas
  - Distância entre Centróide
  - Outras funções

Hierárquico

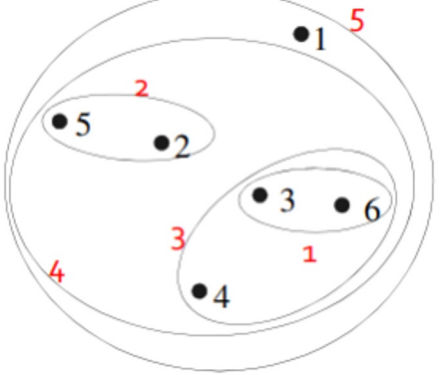
Distância Mínima



Distância Máxima



Distância Média



Hierárquico

- Determinístico
- Fácil interpretação
  - Permite definir a quantidade de grupos após a execução
- Complexidade de tempo O(n * n * n) para implementação ingênua
  - n = número de instâncias
- Pode ser otimizado para alcançar O(n * n * log(n)) com as estruturas adequadas

Hierárquico

- Possíveis problemas:
  - Alta complexidade computacional
  - Implementação Gulosa
  - Sensível a ruído e outliers
  - Formas não globulares (dependendo da forma de junção dos grupos)

Proposta prática

- Notebooks

  KMeans e DBScan

  Hierarquico e GMM