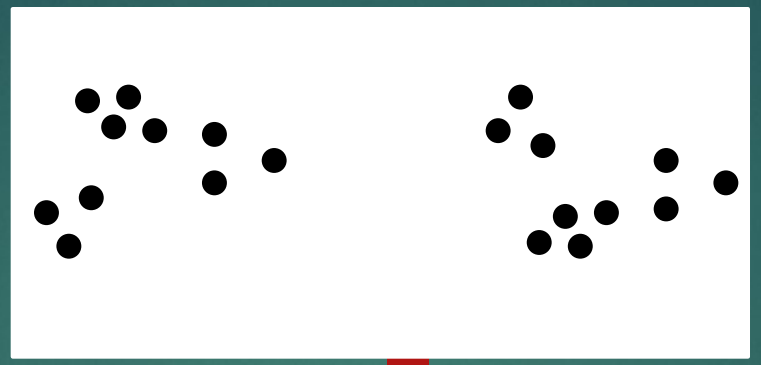
ELTD1 - Introdução À Modelagem e Aprendizado
9: Agrupamentos

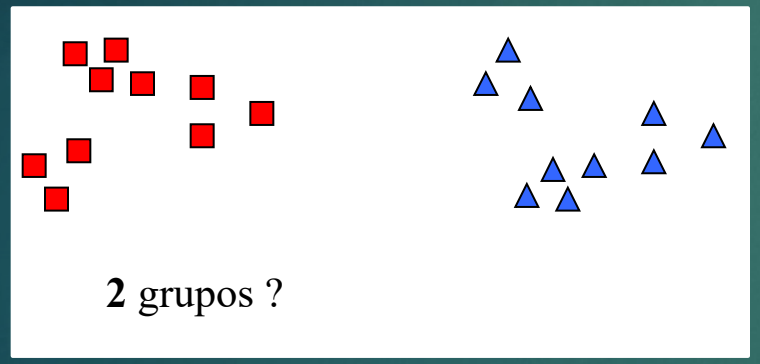
PROF. HEITOR SILVÉRIO LOPES (2024-2)
DEPARTAMENTO DE ELETRÔNICA - DAELN
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - UTFPR

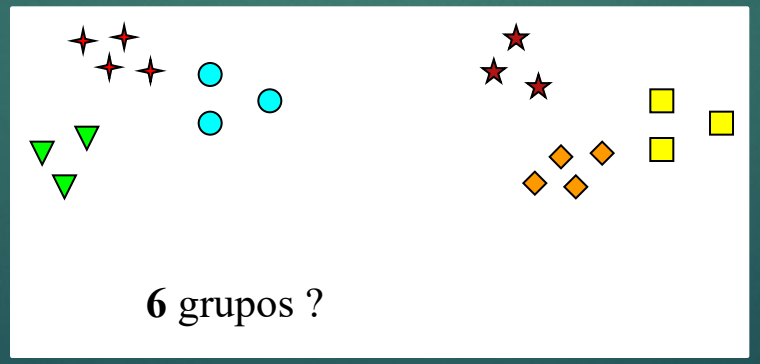
Prof.
H.S.Lopes
UTFPR/2024

Ambiguidade de agrupamentos

Quantos grupos há aqui?







O que é a Análise de agrupamentos

- Definição:
  - Dado um conjunto de dados (objetos) multidimensionais, separar em grupos (*clusters*) que tenham algum significado e/ou utilidade
  - Objetos de um grupo devem ser similares uns aos outros e dissimilares com os objetos de outros grupos, de acordo com algum critério
  - A tarefa de Agrupamento de dados é **NÃO**-supervisionada pois não há informação prévia sobre como dividir os dados em grupos

- Objetivo:
  - Agrupamento para compreensão
  - Agrupamento por utilidade

Agrupamento para Compreensão

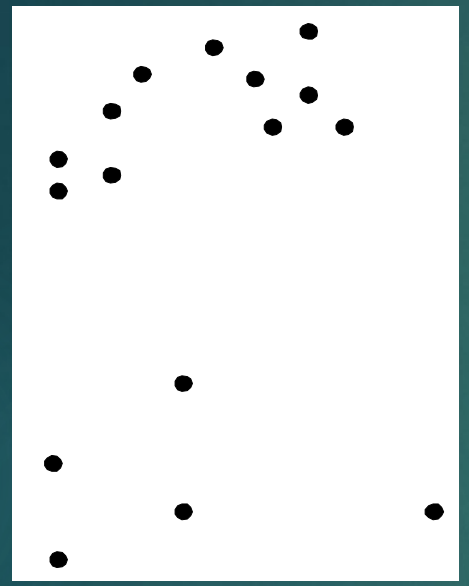
- Grupos de objetos conceitualmente semelhantes compartilham características comuns
- Humanos têm habilidade inata para:
  - Dividir objetos em grupos (agrupamento)
  - Atribuir outros objetos a grupos conhecidos (classificação)
- Os agrupamentos capturam a estrutura natural dos dados, por exemplo:
  - Biologia: taxonomia de seres vivos, busca de genes com funções similares
  - Medicina: identificação de variantes ou tipos de doenças, análise da distribuição temporal ou geográfica de doenças
  - Engenharia: identificação de grupos de falhas em sistemas eletro-eletrônicos, agrupamento de sinais elétricos variantes no tempo

Agrupamento por Utilidade

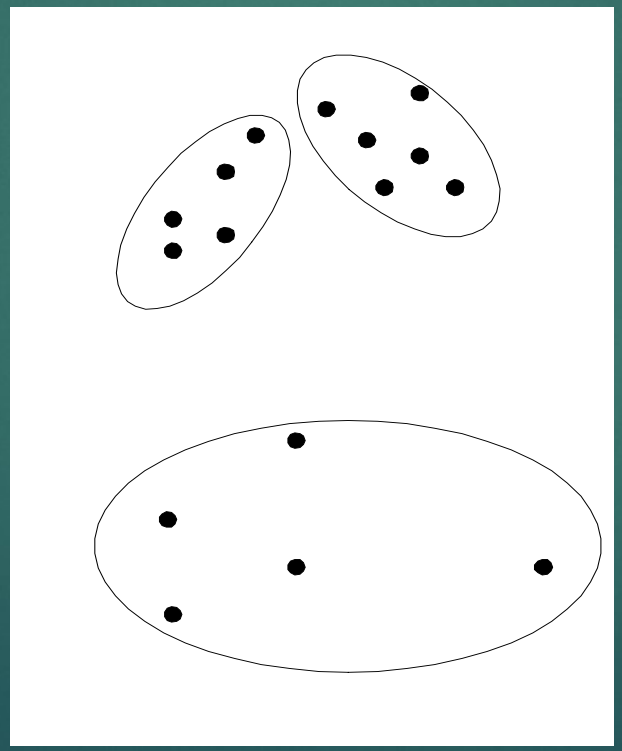
- Cada grupo pode ser caracterizado por um protótipo
- Um protótipo é um elemento representativo do grupo.
- Aplicabilidade:
  - Sumarização: objetiva a redução do tamanho de *datasets* a um conjunto de protótipos pois alguns algoritmos tem complexidade $O(m^2)$
  - Compactação: usado em quantização vetorial de imagens, sons e vídeo
  - Método dos vizinhos mais próximos: (*kNN*) utiliza-se os protótipos em vez dos objetos para reduzir o número de cálculos de distâncias

Formas de agrupamentos

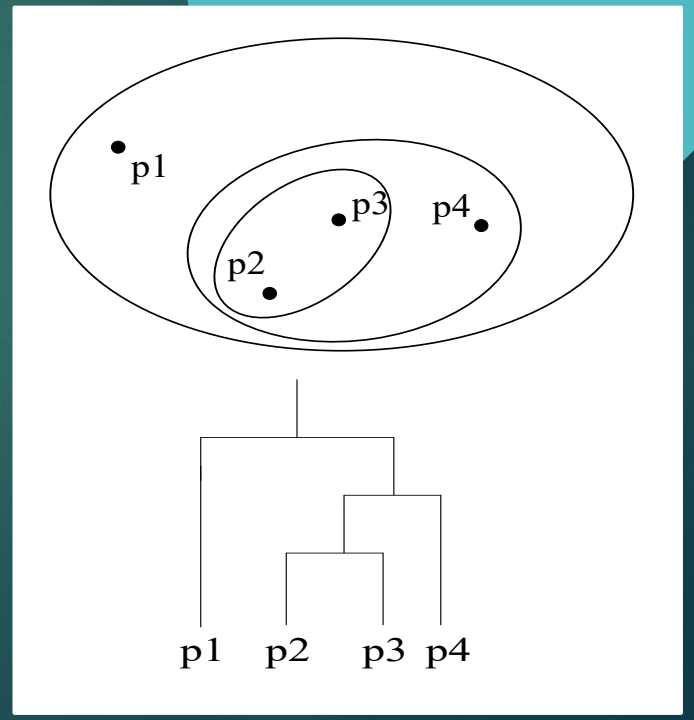
Dados originais:



Agrupamento Particional: é uma divisão simples do conjunto de objetos em subconjuntos sem intersecção



Agrupamento Hierárquico: é um conjunto de grupos organizados como uma árvore hierárquica, onde cada nó (grupo) da árvore é a união de seus nós–filhos (subgrupos)



Tipos de *clusters*

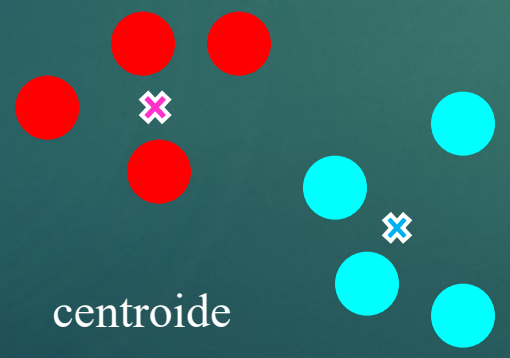
- *Clusters* bem separados
- *Clusters* baseados em protótipos (centros)
- *Cluster* baseados em grafos (contíguos)
- *Clusters* baseados em densidade
- *Clusters* conceituais

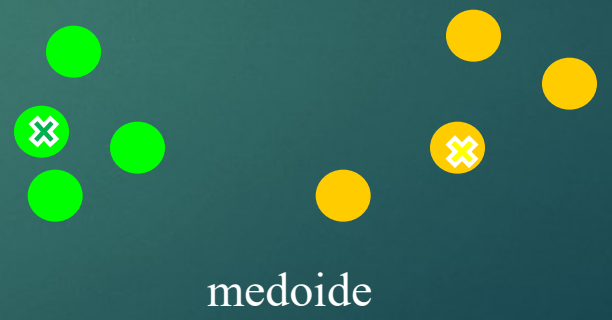
Clusters bem separados

- Um *cluster* é um conjunto de objetos tal que cada objeto está mais próximo (ou é mais similar, segundo alguma métrica) a cada um dos objetos do seu grupo, do que qualquer objeto que não esteja em seu grupo

Clusters baseados em protótipos

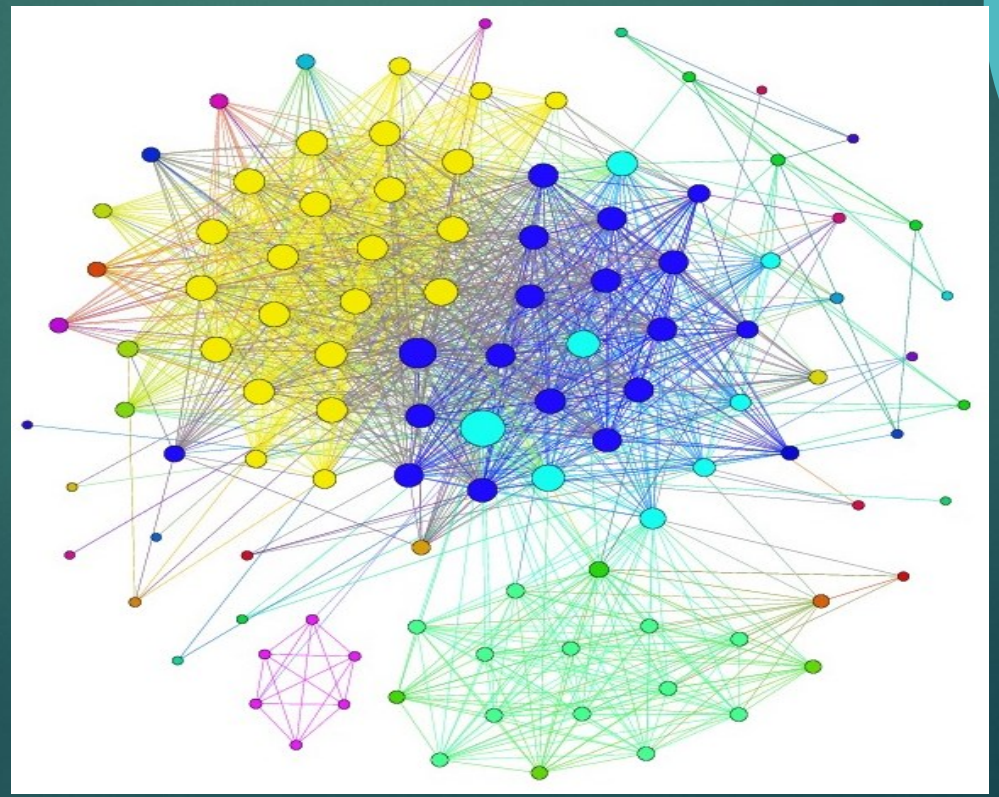
- Um *cluster* é um conjunto de objetos tal que um objeto de um *cluster* está próximo (ou é mais semelhante) ao "**centro**" de seu *cluster* do que do centro de qualquer outro *cluster*
- O centro do *cluster* é designado como:
  - centroide (a média de todos os pontos do *cluster*) ou
  - medoide (o ponto mais representativo do *cluster*)





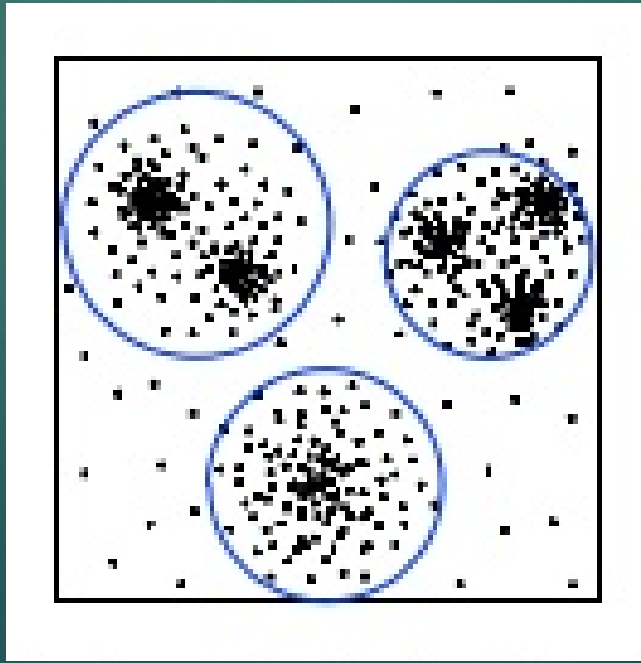
Clusters baseados em grafos

- Também conhecidos como contíguos, vizinho mais próximo ou transitivos
- Um *cluster* é um conjunto de objetos tal que um objeto do *cluster* está próximo (ou é mais similar) a um ou mais objetos do *cluster* do que a qualquer objeto fora do *cluster*



Clusters baseados em densidade

- Um *cluster* é uma região com alta densidade de objetos, que são separadas por regiões de baixa densidade, de outras regiões de alta densidade.
- Utiliza-se esta abordagem quando os *clusters* são irregulares ou entrelaçados, e quando existe ruído ou *outliers* nos dados



Principais algoritmos de agrupamentos

1- Algoritmo K-*means* e suas variantes

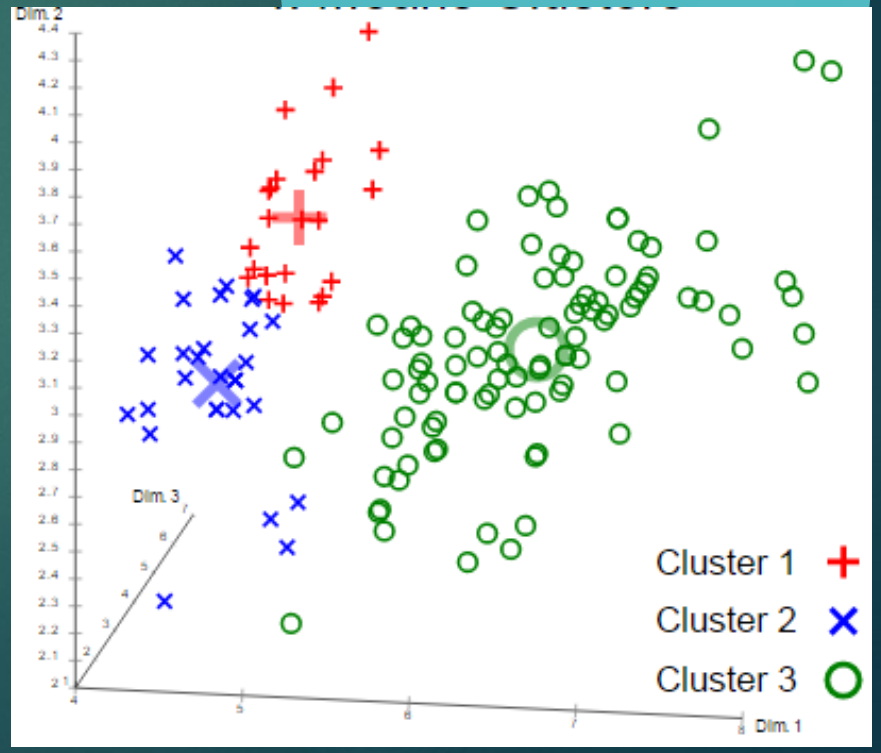
2- Agrupamento hierárquico

3- Agrupamento baseado em densidade (DBSCAN = *Density-based spatial clustering of applications with noise*)

1- Algoritmo K-*means*

$$c_i = \frac{1}{m_i} \sum_{x \in C_i} x$$

- Cada *cluster* é associado a um centroide (média dos pontos que estão no *cluster*)
- O número de *clusters K* deve ser especificado <u>*a priori*</u>
- Em geral, os centroides iniciais são escolhidos <u>aleatoriamente</u>. Assim, *clusters* produzidos podem variar de uma rodada para outra
  - Cada ponto é atribuído ao *cluster* com o centroide mais próximo
  - Normalmente é utilizada a distância **Euclidiana**, mas outras podem ser utilizadas



1- Algoritmo K-*means*

- O algoritmo básico é simples e <u>converge rapidamente</u>
- A condição de parada é a estagnação do reposicionamento dos centroides
- A complexidade de tempo é ***O****(n.K.I.d),* onde:
  - ***n*** = número de pontos, ***K*** = número de *clusters*,
    I = número de iterações, ***d***= número de atributos
- A complexidade de memória é ***O****((d+K)d)*
- O algoritmo é eficiente se *K<<d*

```
Select K points as the initial centroids.
repeat
    Form K clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

Avaliação dos *clusters* do K-*means*

- A medida mais usual para a avaliação dos *clusters* é a Soma dos Erros Quadráticos (SSE)
  - Para cada ponto, o erro é a distância ao *cluster* mais próximo
  - Para obter o SSE, eleva-se ao quadrado e somam-se os erros:

$$SSE = \sum_{i=1}^{K} \sum_{\forall x \in C_i} dist^2(c_i, x)$$

  - Onde x é um ponto de dados alocado ao *cluster* C_i e c_i é o centroide atual do *cluster* C_i
  - Dados dois agrupamentos, pode-se escolher aquele com o menor erro SSE
  - <u>PROBLEMA</u>: pode-se reduzir o SSE incrementando o número de *clusters K*. Porém pode-se obter um agrupamento com poucos *clusters* (*K* baixo) com SSE menor do que um agrupamento muitos *clusters* (alto *K*)

Avaliação dos *clusters* do K-*means*

- Opções para a escolha de:
  - Função de proximidade (distância entre pontos)
  - Tipo de centro: centroide (média) ou medoide (moda)
  - Função objetivo a ser minimizada/maximizada:

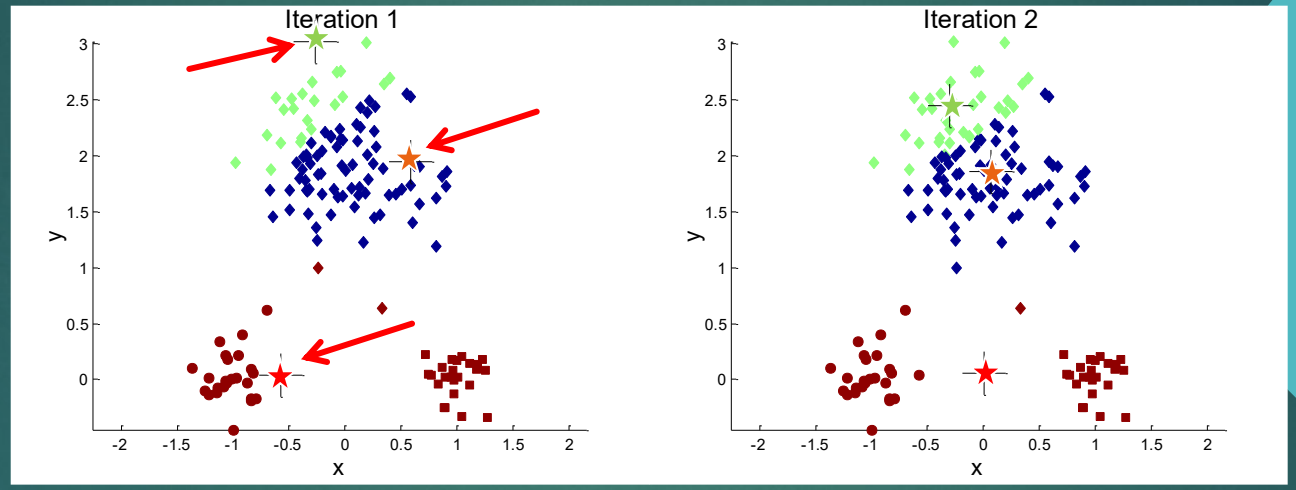
Proximity Function	Centroid	Objective Function
Manhattan (L_1)	median	Minimize sum of the L_1 distance of an object to its cluster centroid
Squared Euclidean (L_2^2)	mean	Minimize sum of the squared L_2 distance of an object to its cluster centroid
cosine	mean	Maximize sum of the cosine similarity of an object to its cluster centroid
Bregman divergence	mean	Minimize sum of the Bregman divergence of an object to its cluster centroid

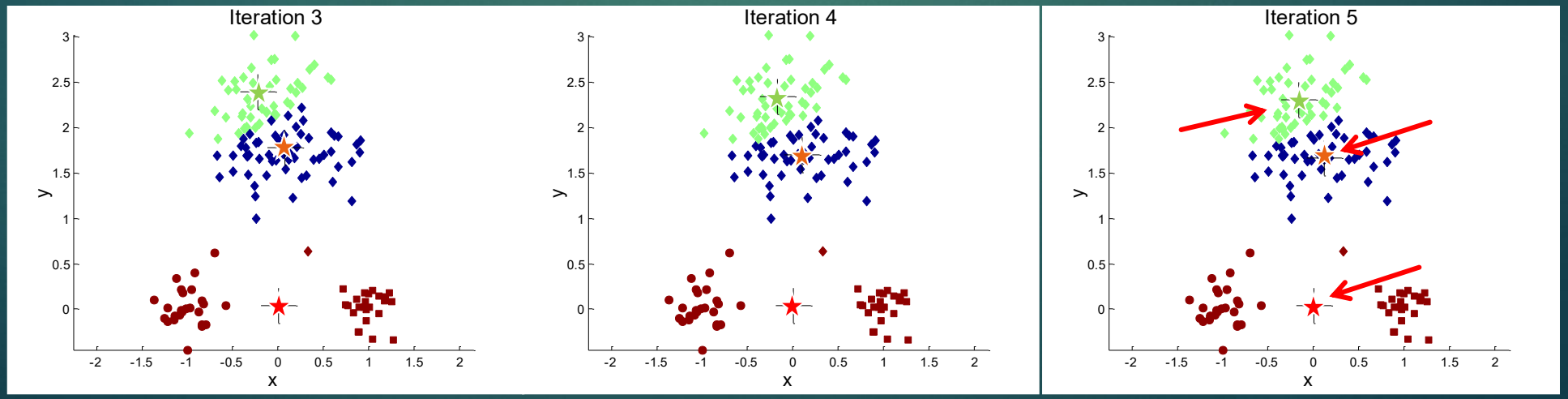
  - Bregman divergence pode ser: distância de Mahalanobis, divergência de Kullback-Leibler, distância Itakura-Saito, etc

Prof.
H.S.Lopes
UTFPR/2024

Escolha dos centroides iniciais

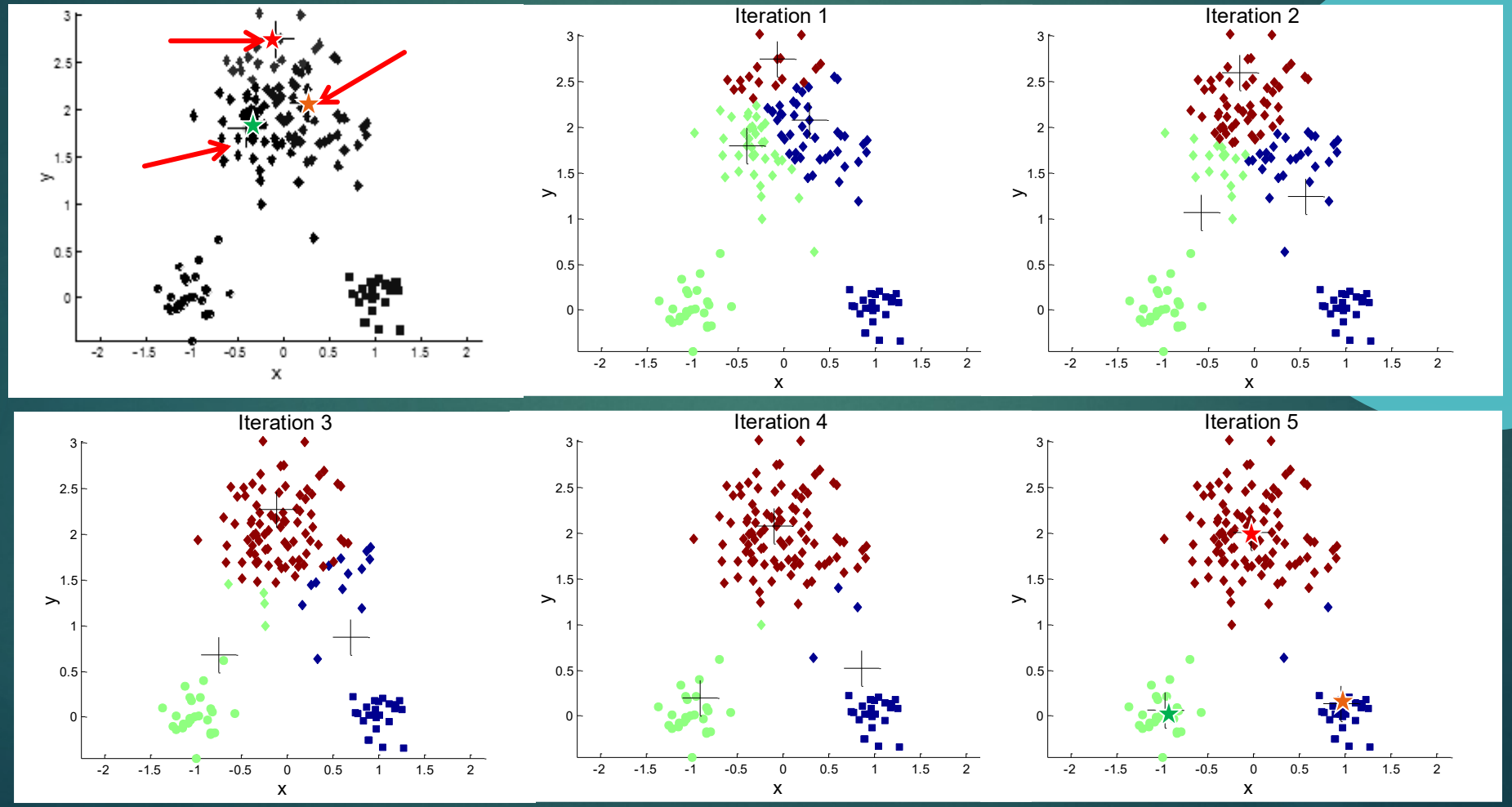
- Escolher centroides iniciais apropriados é uma <u>etapa fundamental</u> para o *K-means*
- <u>Configuração inicial #1:</u> convergência <u>incorreta</u> a partir de uma configuração inicial de centroides:





Escolha dos centroides iniciais

- Configuração inicial #2: convergência correta a partir de uma configuração inicial de centroides:



Escolha dos centroides iniciais

- Se houver K *clusters* (reais), então a possibilidade de selecionar um centroide de cada *cluster* é muito pequena à medida que *K* cresce
- Se os *clusters* forem do mesmo tamanho (*n*), então:

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

- Por exemplo, se K=10, então probabilidade= $10!/10^{10}$ = 0,00036

- Conclusão:

Algumas vezes os centroides iniciais vão se ajustar da maneira certa mas algumas vezes não vão →

Escolha dos centroides iniciais

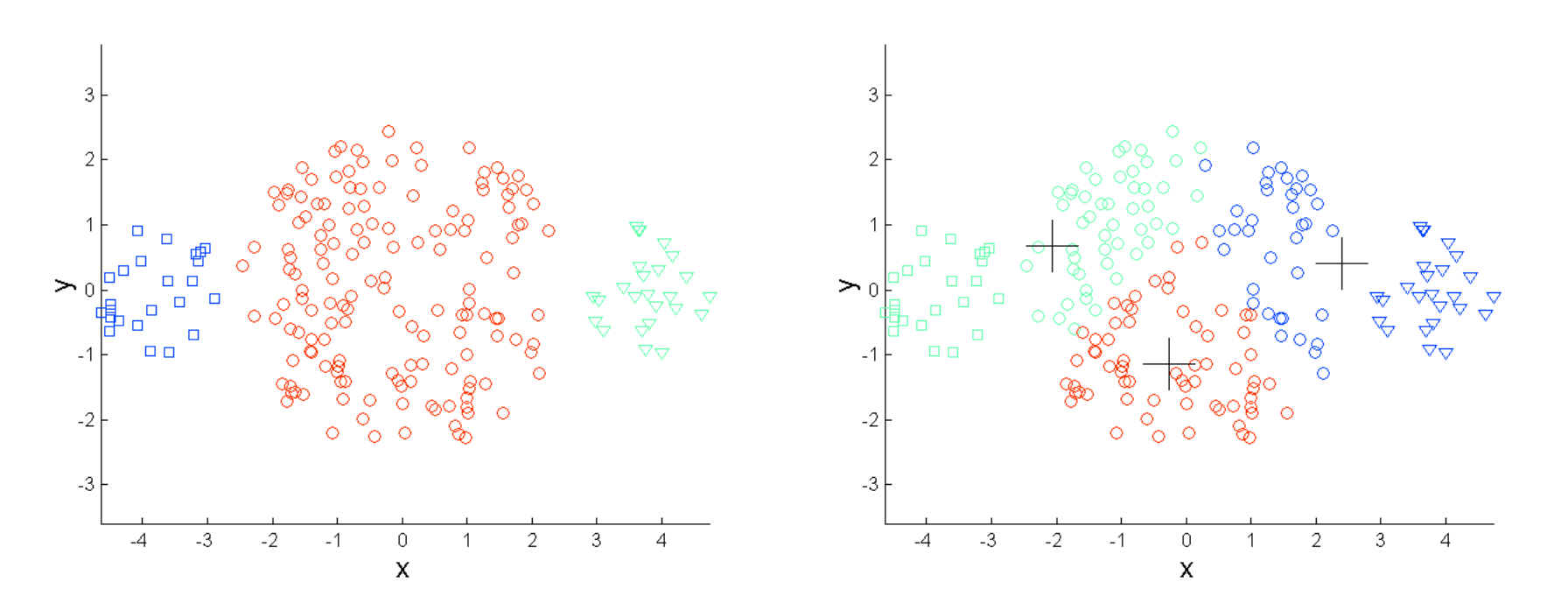
- Realizar múltiplas rodadas para minimizar o efeito da escolha
- Selecionar mais do que *K* centroides iniciais e, então, selecionar dentre estes centroides iniciais
- Selecionar os que forem mais separados entre si

- Pré-processamento:
  - Normalização dos dados
  - Eliminar *outliers*

- Pós-processamento (manual):
  - Eliminar *clusters* pequenos que podem representar *outliers*
  - Particionar *clusters* "frouxos", isto é, *clusters* com SSE relativamente altos
  - Fundir *clusters* que estiverem "perto" e que tenham baixo SSE

Limitações do K-*means*

- *Clusters* com <u>tamanhos</u> muito diferentes

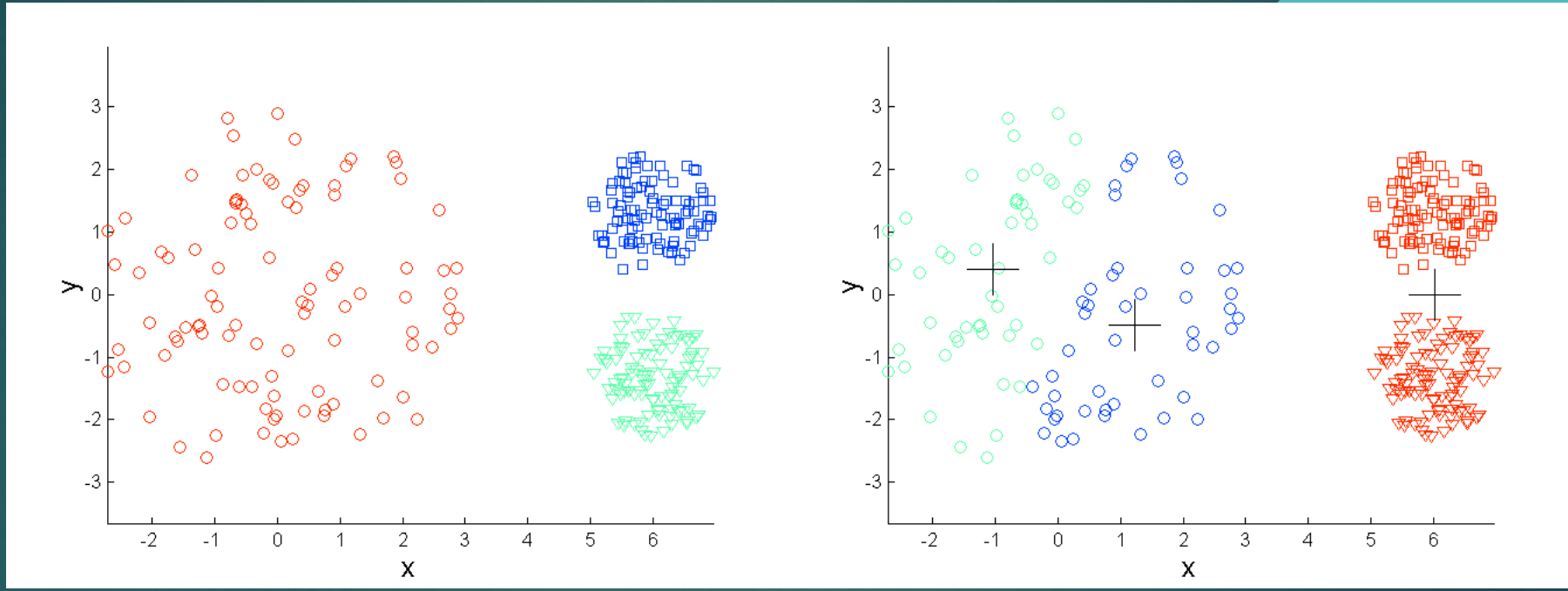


Pontos originais

K-*means* (3 *Clusters*)

Limitações do K-*means*

- *Clusters* com <u>densidades</u> muito diferentes

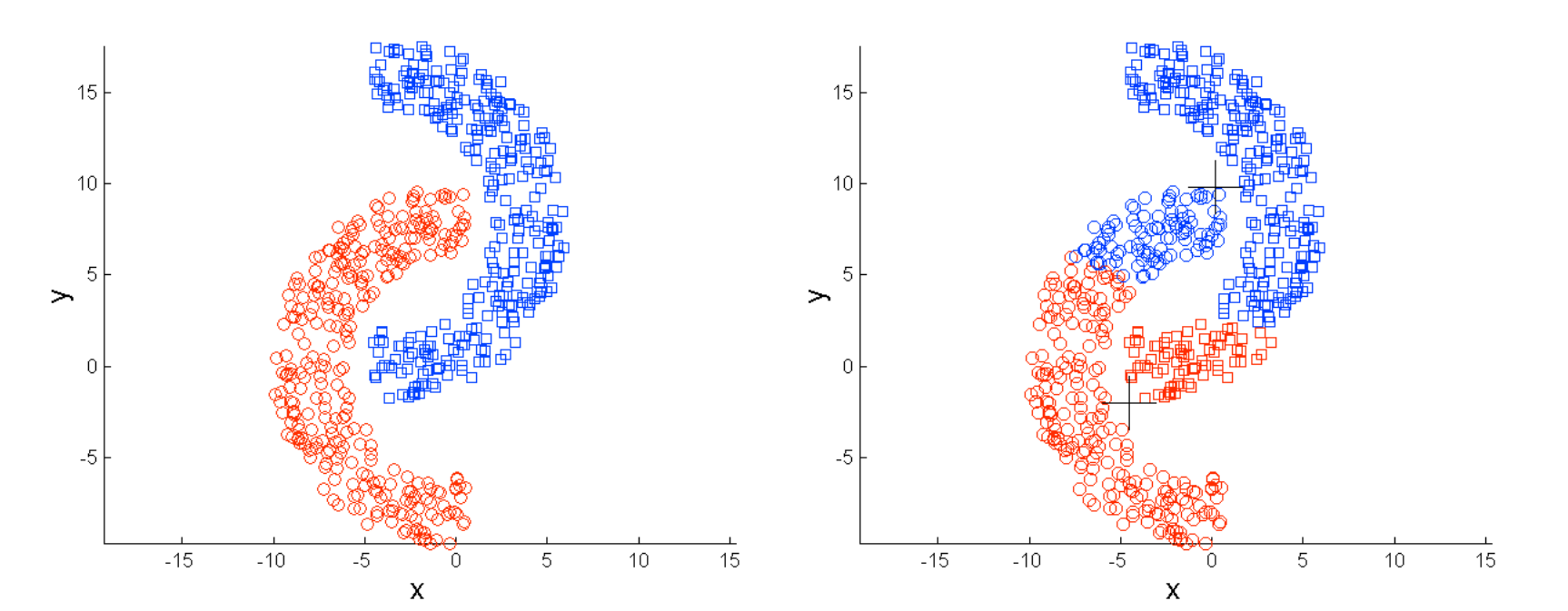


Pontos originais

K-*means* (3 *Clusters*)

Limitações do K-*means*

- *Clusters* com <u>forma</u> não-globular



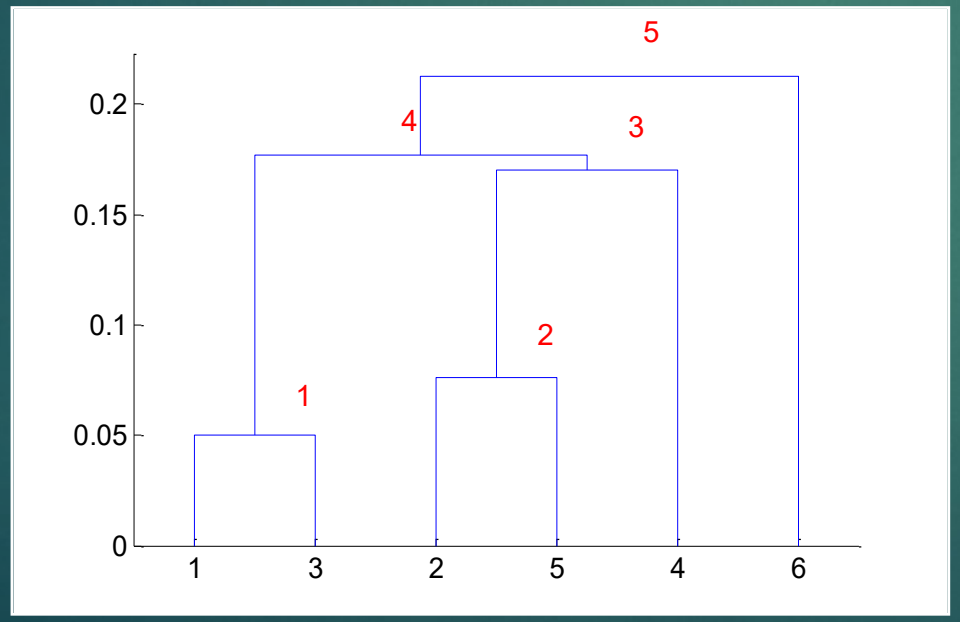
Pontos originais

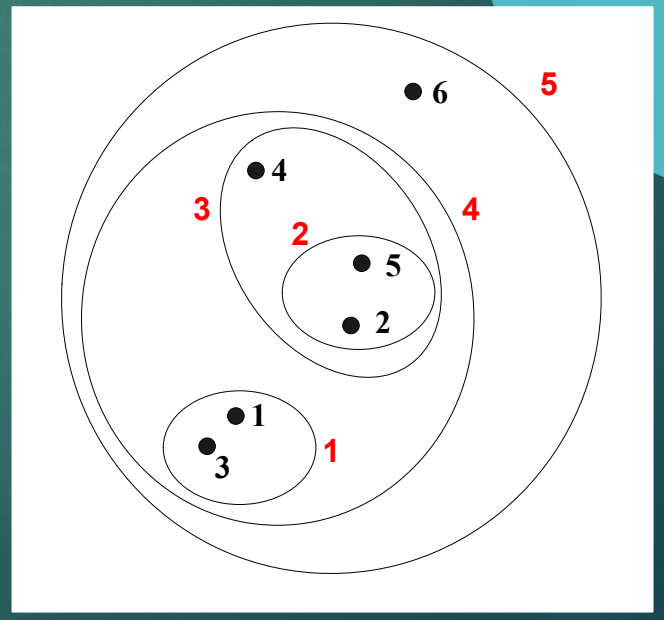
K-*means* (3 *Clusters*)

Prof.
H.S.Lopes
UTFPR/2024

2- Agrupamento hierárquico aglomerativo

- Produz um conjunto de *clusters* aninhados organizados como uma árvore hierárquica
- Pode ser visualizado como um dendograma:
  - É um diagrama semelhante a uma árvore e mostra as sequências de misturas e particionamentos





2- Agrupamento hierárquico aglomerativo

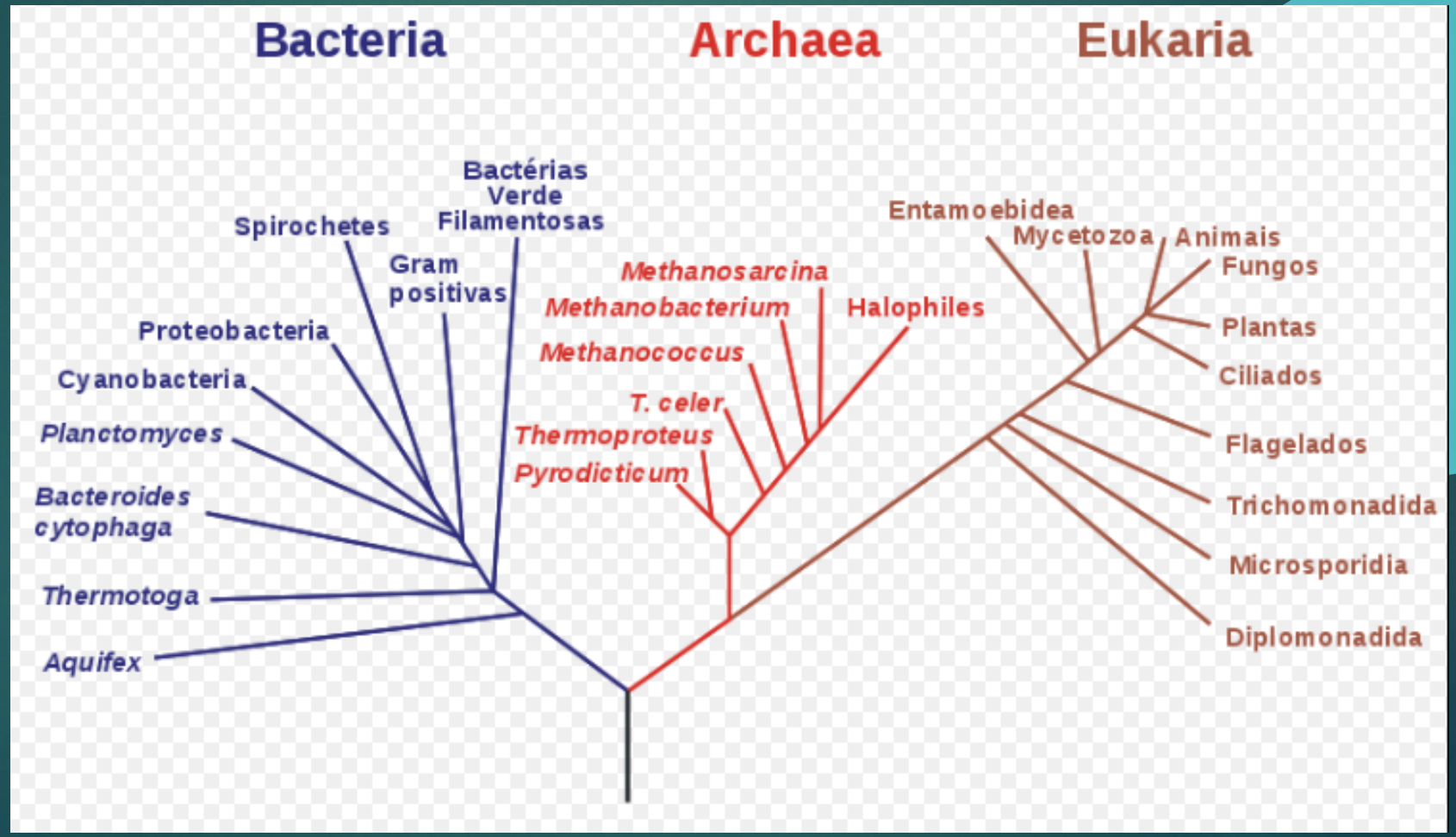
- Pontos fortes:

  - Não precisa assumer qualquer número particular de *clusters* iniciais
    - Qualquer número de *clusters* pode ser obtido por "cortes" do dendograma ao nível que for adequado
  - O agrupamento hierárquico pode corresponder a texonomias com significado próprio:
    - Exemplo da Biologia
    - Exemplo da Computação

Prof.
H.S.Lopes
UTFPR/2024

2- Agrupamento hierárquico aglomerativo

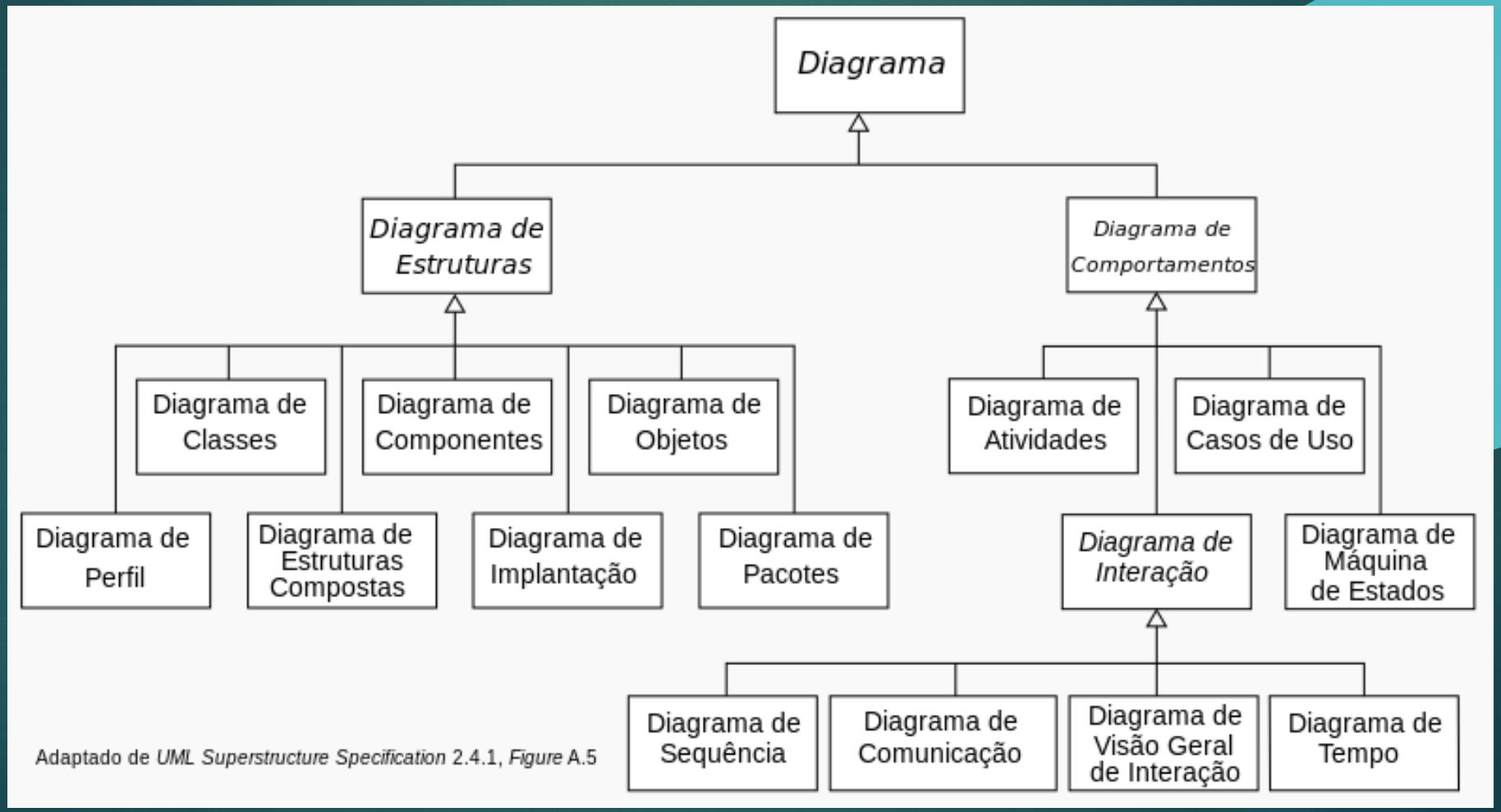
- Árvore filogenética



Prof.
H.S.Lopes
UTFPR/2024

2- Agrupamento hierárquico aglomerativo

- Estrutura dos diagramas UML



Tipos de agrupamento hierárquico

- Aglomerativo:
  - Começa com os pontos como *clusters* individuais
  - A cada passo, misturar o par de *clusters* mais próximo até que que apenas fiquem apenas K *clusters*
- Divisivo:
  - Começa com um *cluster* que inclui todos os pontos
  - A cada passo, divide um *cluster* até que cada *cluster* contenha um único ponto (ou existam K *clusters*)
- Algoritmos tradicionais de agrupamento hierárquico utilizam uma <u>matriz de distâncias</u>
- Mistura ou particiona um *cluster* por vez

Algoritmo de agrupamento hierárquico

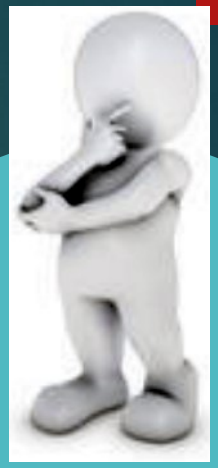
- O algoritmo básico de agrupamento hierárquico inicia considerando cada ponto um *cluster* individual

Algorithm 8.3 Basic agglomerative hierarchical clustering algorithm.

```
Compute the proximity matrix, if necessary.
repeat
    Merge the closest two clusters.
    Update the proximity matrix to reflect the proximity between the new
    cluster and the original clusters.
until Only one cluster remains.
```

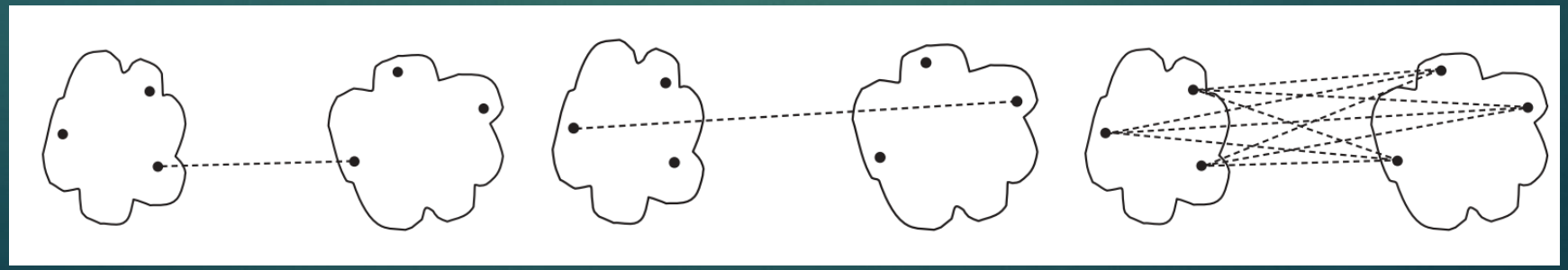
- Complexidade de tempo é $\boldsymbol{O}(n^2.\log n)$ e de memória é $\boldsymbol{O}(n^2)$ onde n é o número de instâncias
- A operação principal é o cálculo da <u>proximidade</u> entre dois *clusters*

Algoritmo de agrupamento hierárquico



Como definir a proximidade inter-*cluster* ?

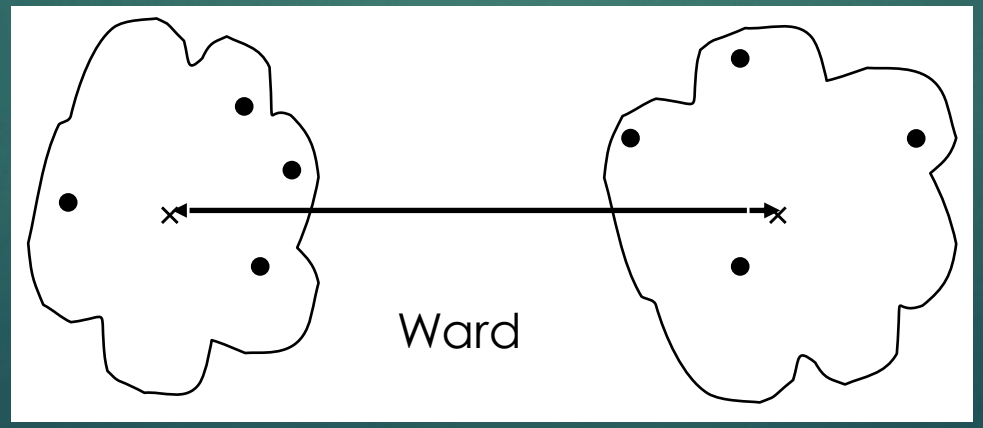
1. **Min:**
   - Também é conhecida como ***single link***.
   - É a distância entre os dois pontos mais próximos de dois *clusters*
2. **Max:**
   - Também é conhecida como ***complete link***.
   - É a distância entre os dois pontos mais distantes de dois *clusters*
3. **Average:**
   - É a distância média entre todos os pares de pontos que estejam em diferentes grupos



Como definir a distância inter- centróides ?

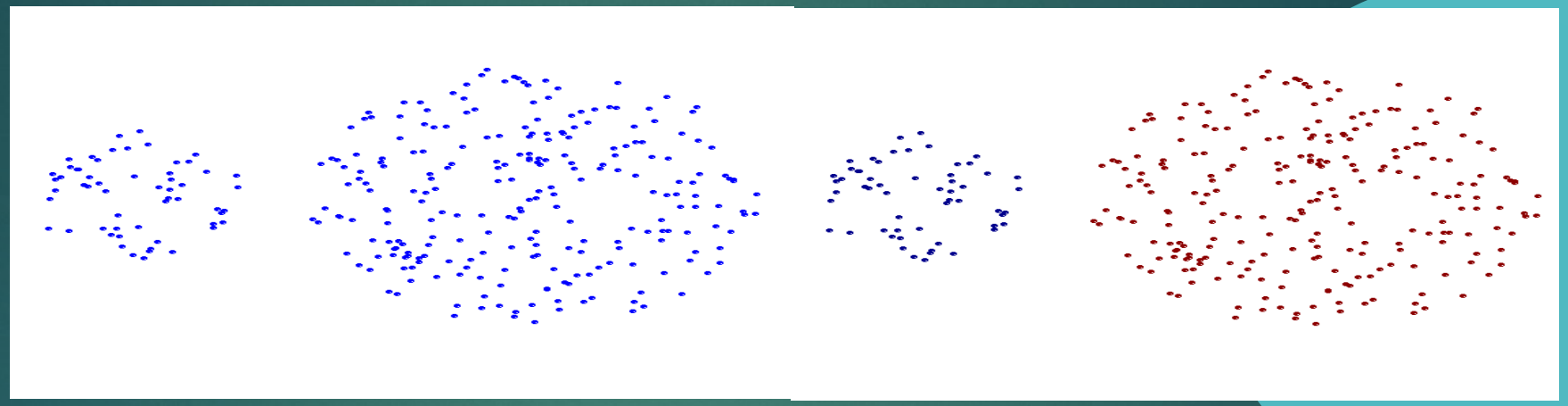
- Distância entre centroides:
  - É a distância entre os centroides de dois grupos

- Método de *Ward*:
  - Também é baseado em centroide
  - Mede a distância entre *clusters* com base no quanto aumenta o SSE ao fundir os dois *clusters* (método de variância mínima)

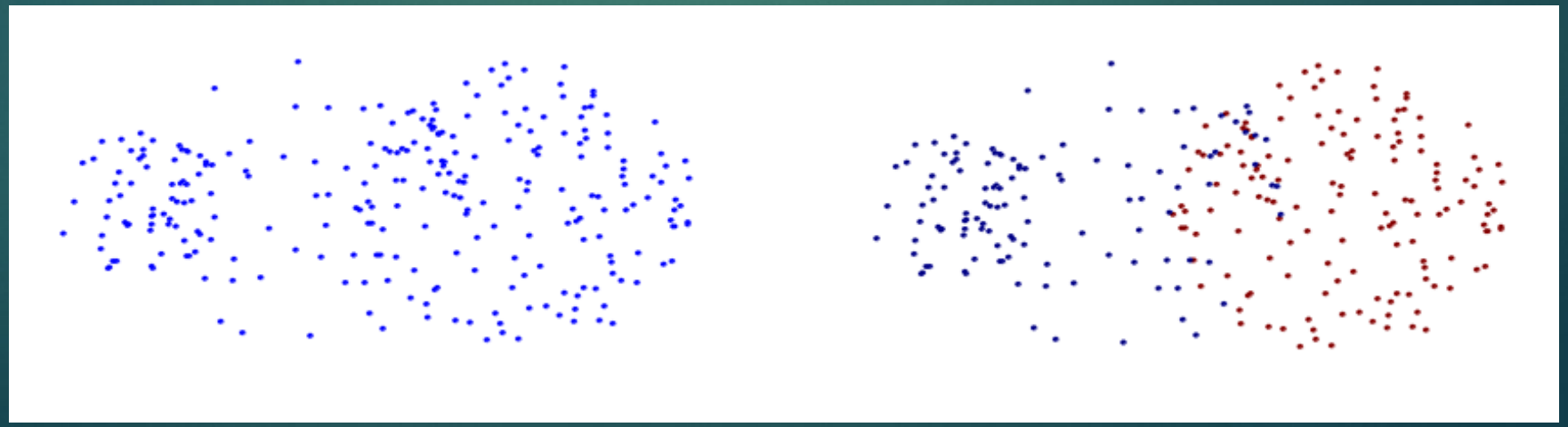


Vantagem/desvantagem do método MIN

- Vantagem: pode lidar com distribuições espaciais não-elípticas

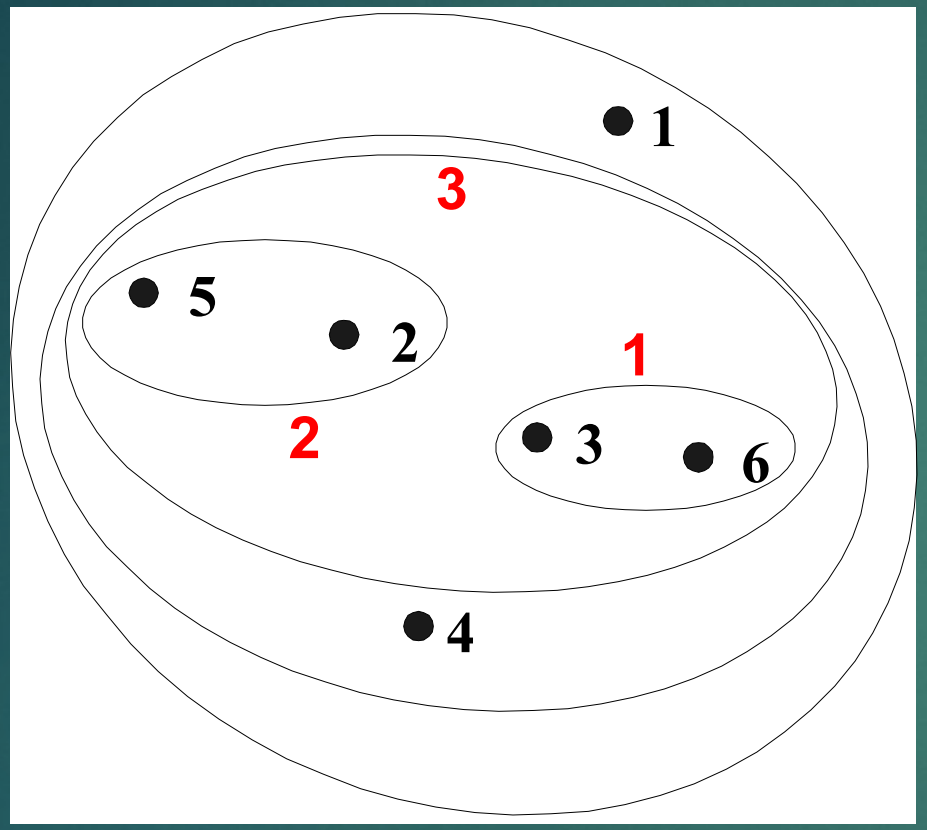


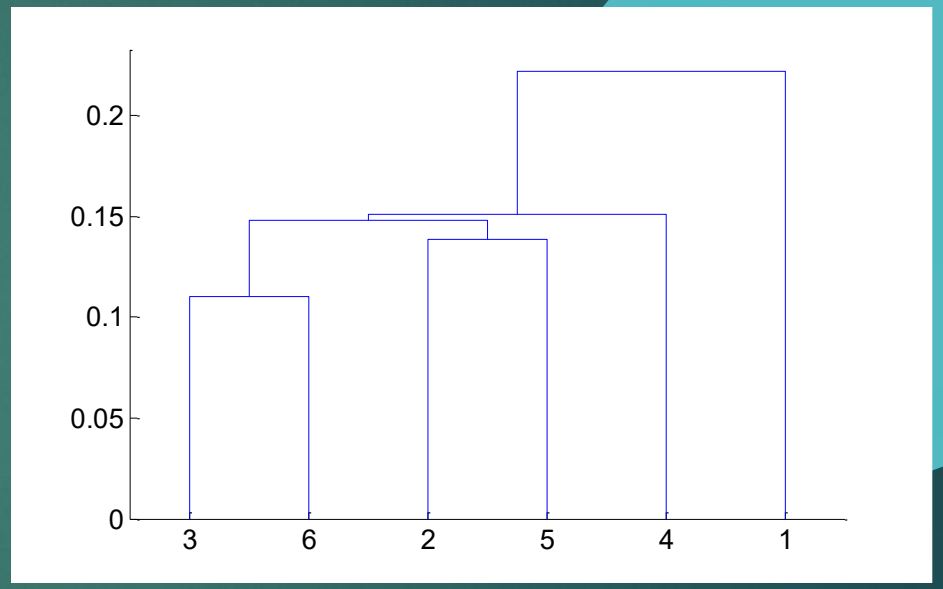
- Desvantagem: sensível a ruído e *outliers*



Prof.
H.S.Lopes
UTFPR/2024

Agrupamento hierárquico com o método **MIN**



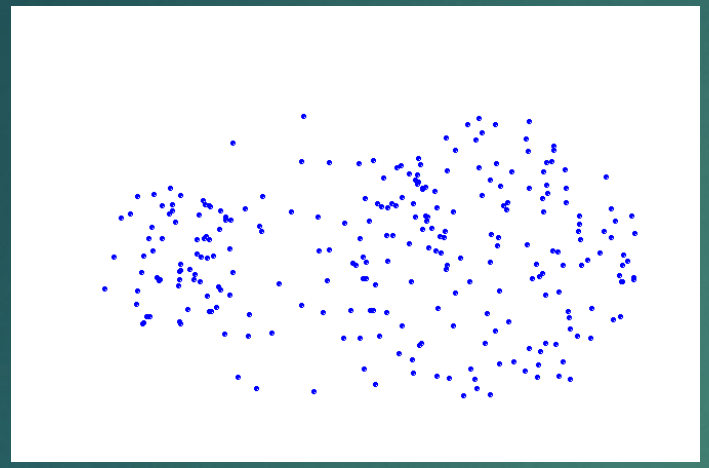


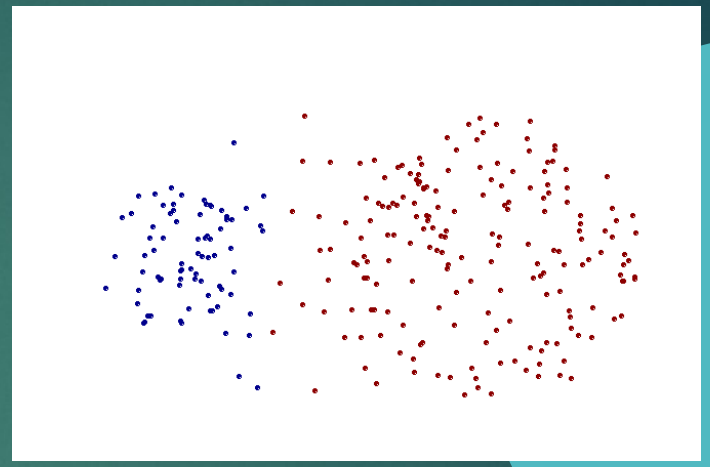
***Clusters* aninhados**

Dendrograma

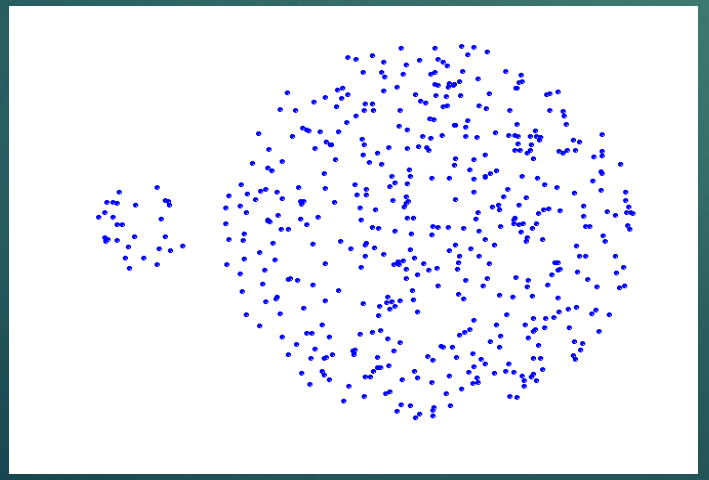
Vantagem/desvantagem do método **MAX**

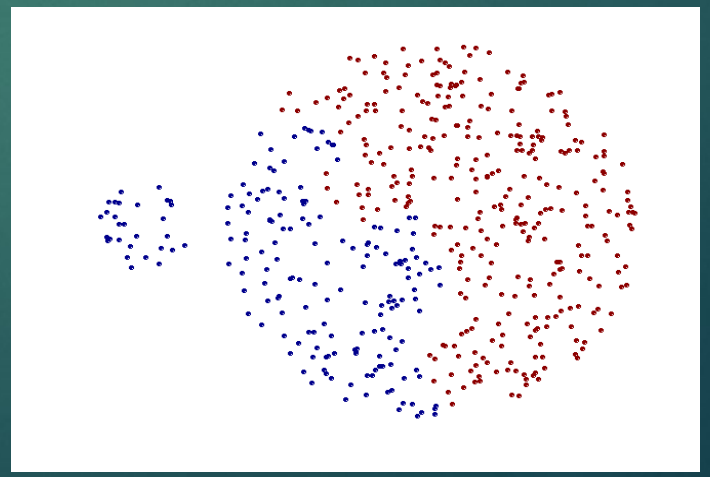
- Vantagem: menos susceptível a ruídos e *outliers*





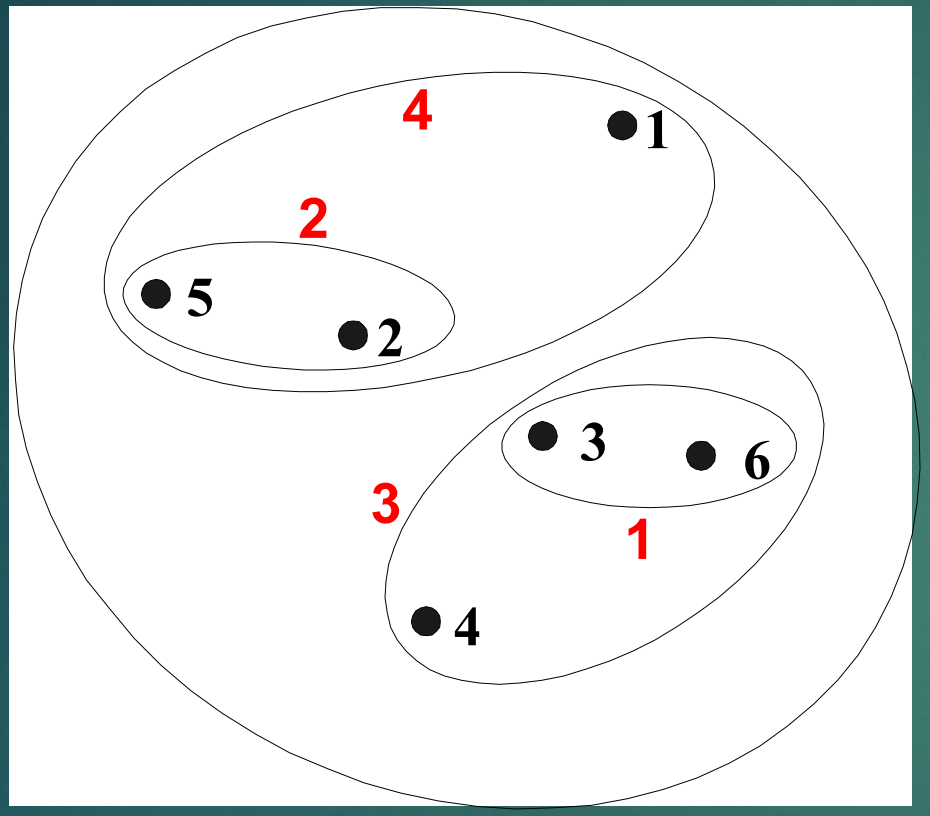
- Desvantagem: tende a quebrar *clusters* grandes e é tendencioso a formar *clusters* globulares



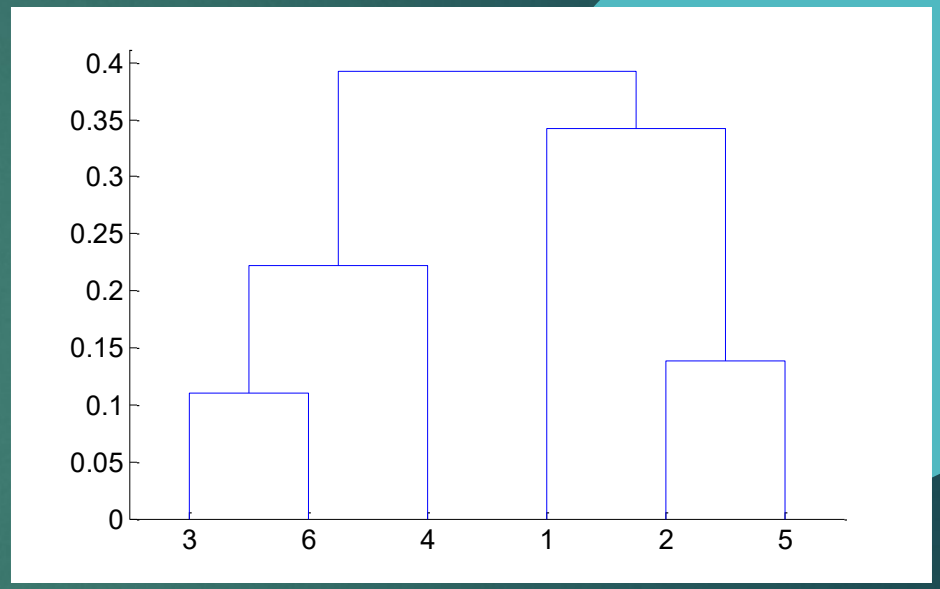


Prof.
H.S.Lopes
UTFPR/2024

Agrupamento hierárquico com o método MAX



***Clusters* aninhados**



Dendrograma

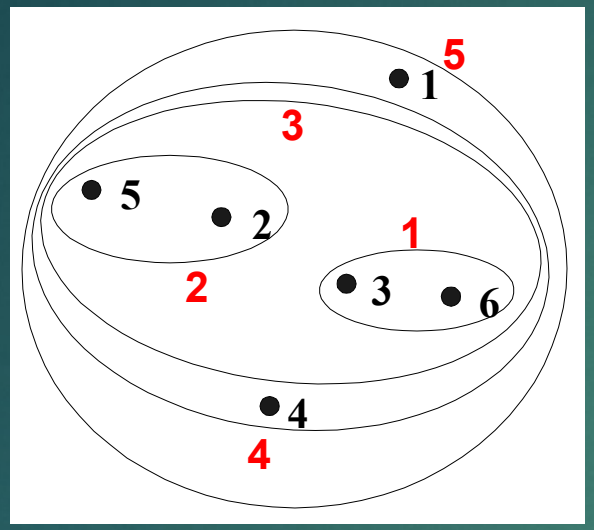
Vantagem/desvantagem dos métodos *Average* e *Ward*

- *Average*:
  - É um compromisso entre Min e Max
  - Vantagem: menos susceptível a ruído e *outliers*
  - Desvantagem: tendencioso a formar *clusters* globulares

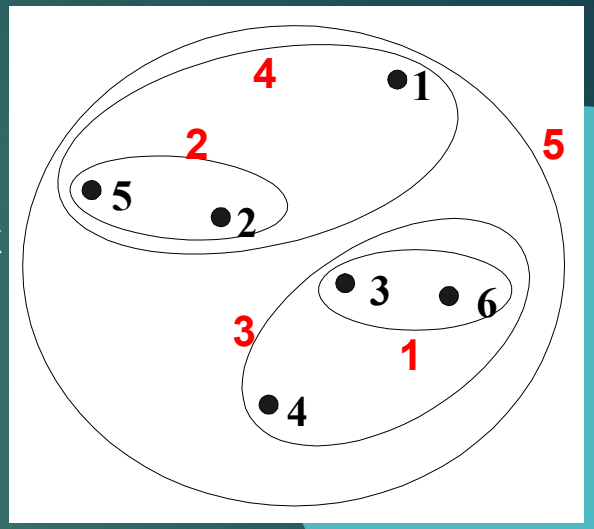
- *Ward*:
  - É similar à média de grupo se a distância entre pontos é medida como distância ao quadrado
  - Tem a mesma vantagem e desvantagem que a *Average*

Prof.
H.S.Lopes
UTFPR/2024

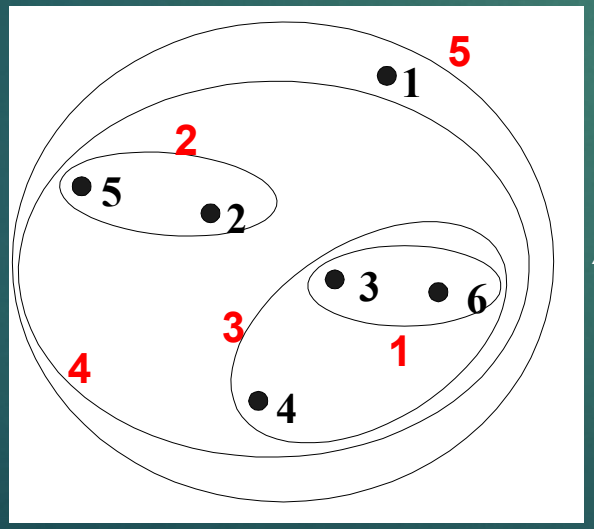
Comparação de métodos de agrupamento hierárquico



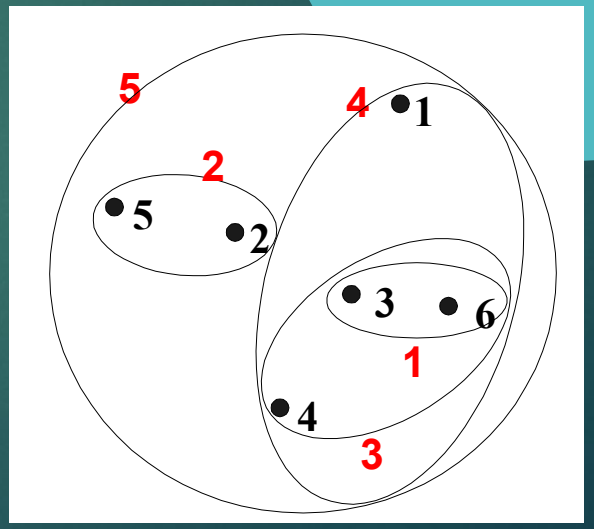
Min



Max



Average

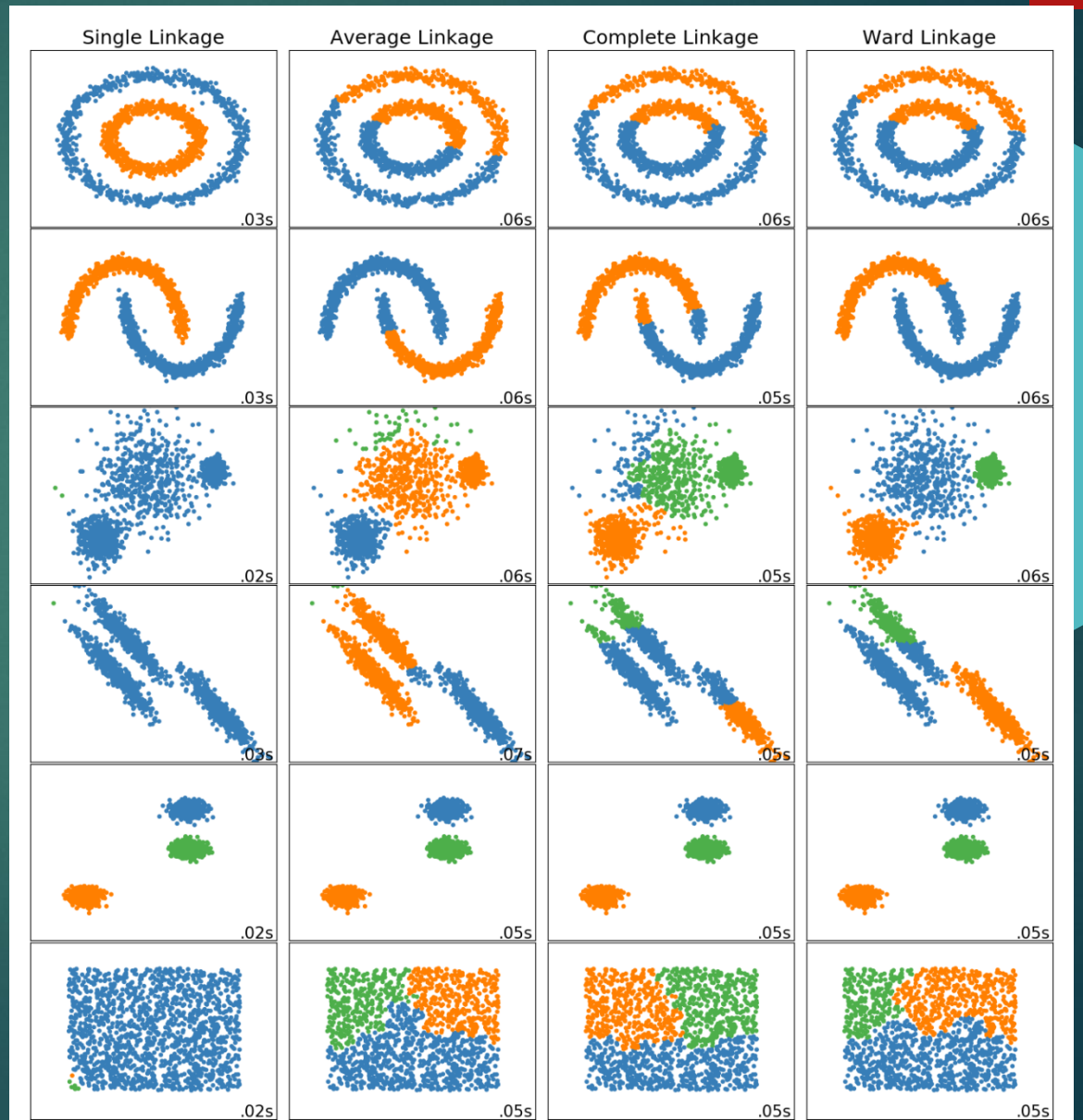


Método de *Ward*

Comparação de métodos de agrupamento hierárquico

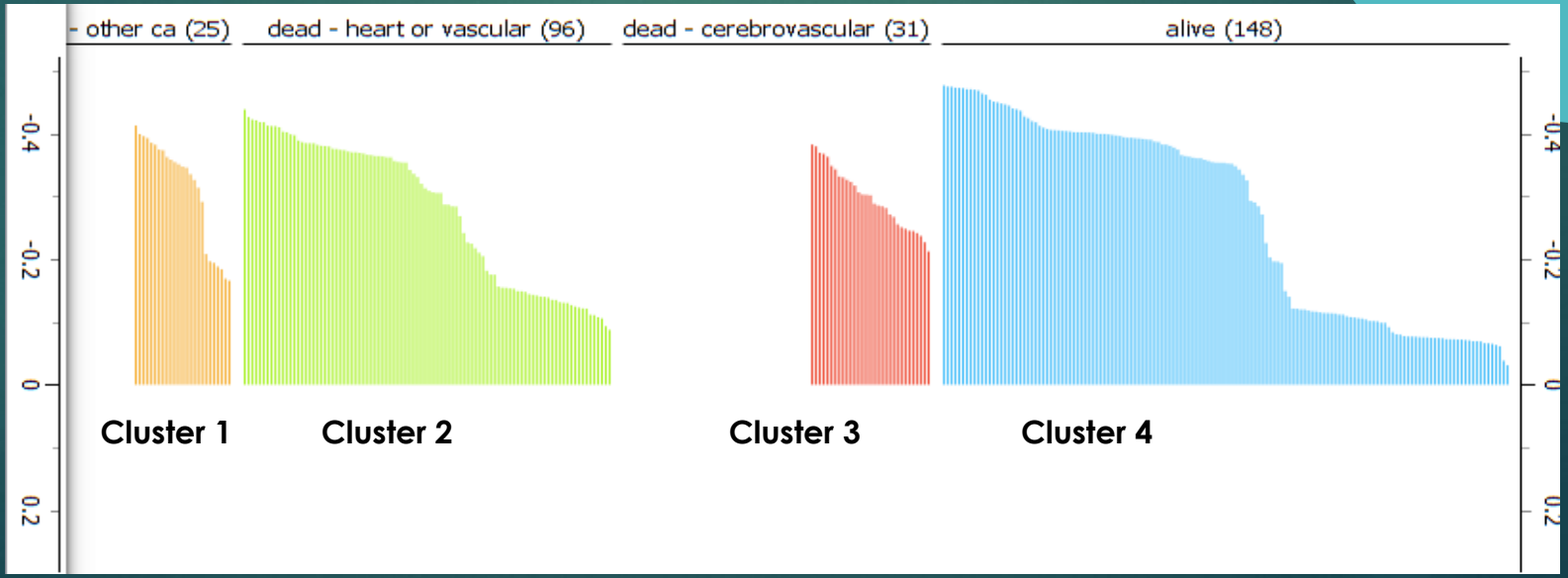
- O Scikit-Learn fornece uma comparação interessante entre métodos:

http://scikit-learn.org/stable/auto_examples/cluster/plot_linkage_comparison.html#sphx-glr-auto-examples-cluster-plot-linkage-comparison-py



Validação gráfica dos agrupamentos

- **Coeficiente de Silhouette** é uma medida que representa o quanto uma instância é semelhante aos seus vizinhos do mesmo *cluster* (coesão) em comparação com as outras instâncias de outros *clusters* (separação)
- O cálculo do coeficiente de Silhouette é baseado em uma métrica de distância, normalmente Euclidiana, cosseno, Mahalanobis, Manhattan, etc.
- O gráfico de Silhouette é bastante informativo para auxiliar a definição do número ideal de *clusters* para uma distribuição desconhecida



Validação gráfica dos agrupamentos

- Para cada instância é calculado o coeficiente de Silhouette entre [-1..+1]* tal que:
  - Um valor positivo indica que a instância está bem adequada ao seu *cluster* e inadequada aos *clusters* vizinhos
  - Um valor próximo a 0 indica que a instância está muito próxima à fronteira de decisão entre dois *clusters* vizinhos
  - Um valor negativo indica que a instância pode ter sido colocada no *cluster* errado
  - Se muitas instâncias têm valor baixo ou negativo, indica que há *clusters* em excesso ou poucos
  - A "largura" do Silhouette indica a quantidade de instâncias em cada *cluster*

* depende da métrica utilizada: distância Euclidiana é [0..1], distância de cosseno é [-1..+1]

Validação gráfica dos agrupamentos

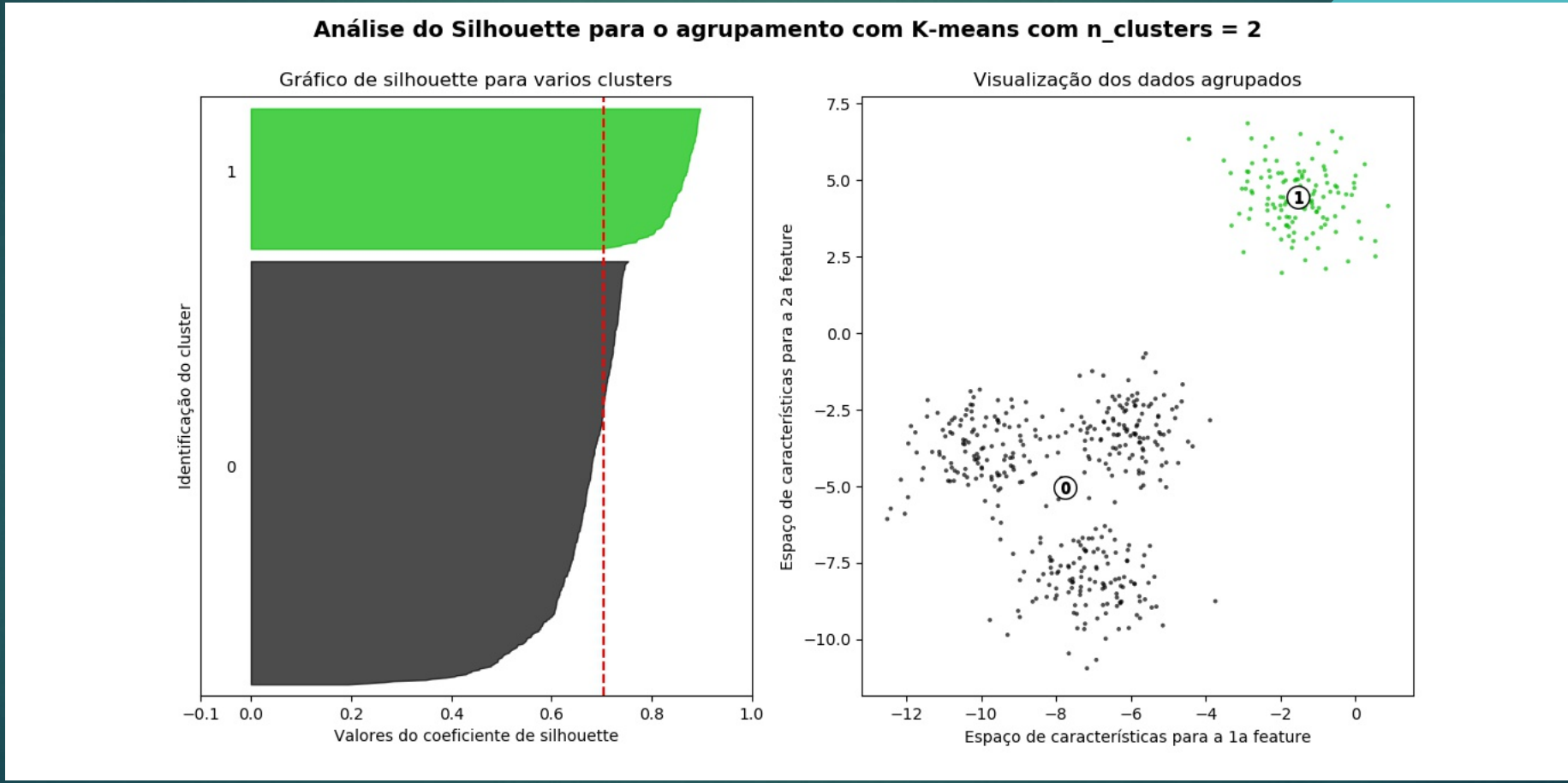
http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

```
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Generating the sample data from make_blobs

X, Y = make_blobs()

no_of_clusters = [2, 3, 4, 5, 6]

for n_clusters in no_of_clusters:

    cluster = KMeans(n_clusters = n_clusters)
    cluster_labels = cluster.fit_predict(X)

    # The silhouette_score gives the
    # average value for all the samples.
    silhouette_avg = silhouette_score(X, cluster_labels)

    print("For no of clusters =", n_clusters,
          " The average silhouette_score is :", silhouet
```

Prof.
H.S.Lopes
UTFPR/2024

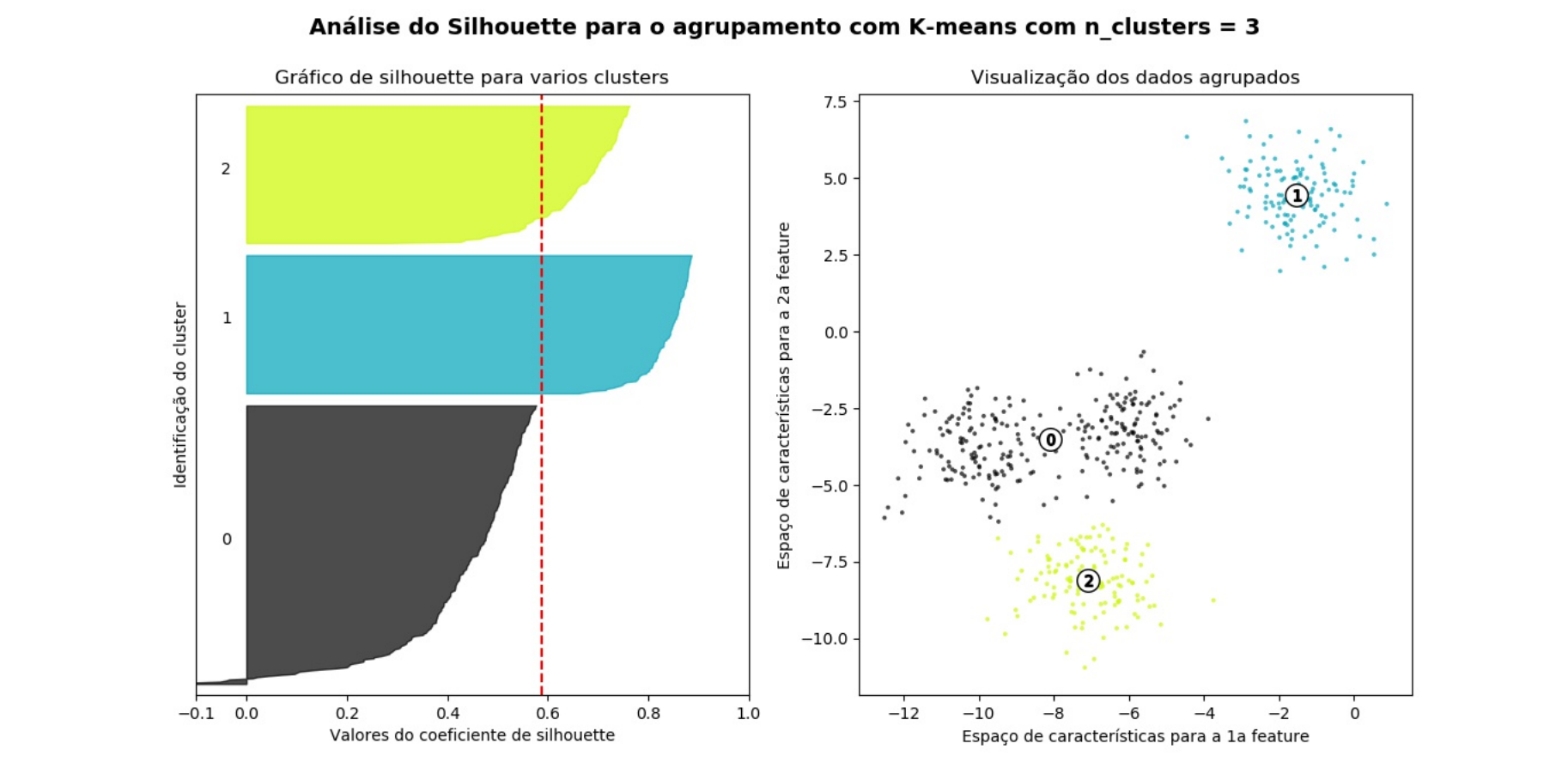
Validação gráfica dos agrupamentos

```
Para n_clusters = 2 o escore silhouette médio é: 0.7049787496083261
Para n_clusters = 3 o escore silhouette médio é: 0.5882004012129721
Para n_clusters = 4 o escore silhouette médio é: 0.6505186632729437
Para n_clusters = 5 o escore silhouette médio é: 0.56376469026194
Para n_clusters = 6 o escore silhouette médio é: 0.4504666294372765
```



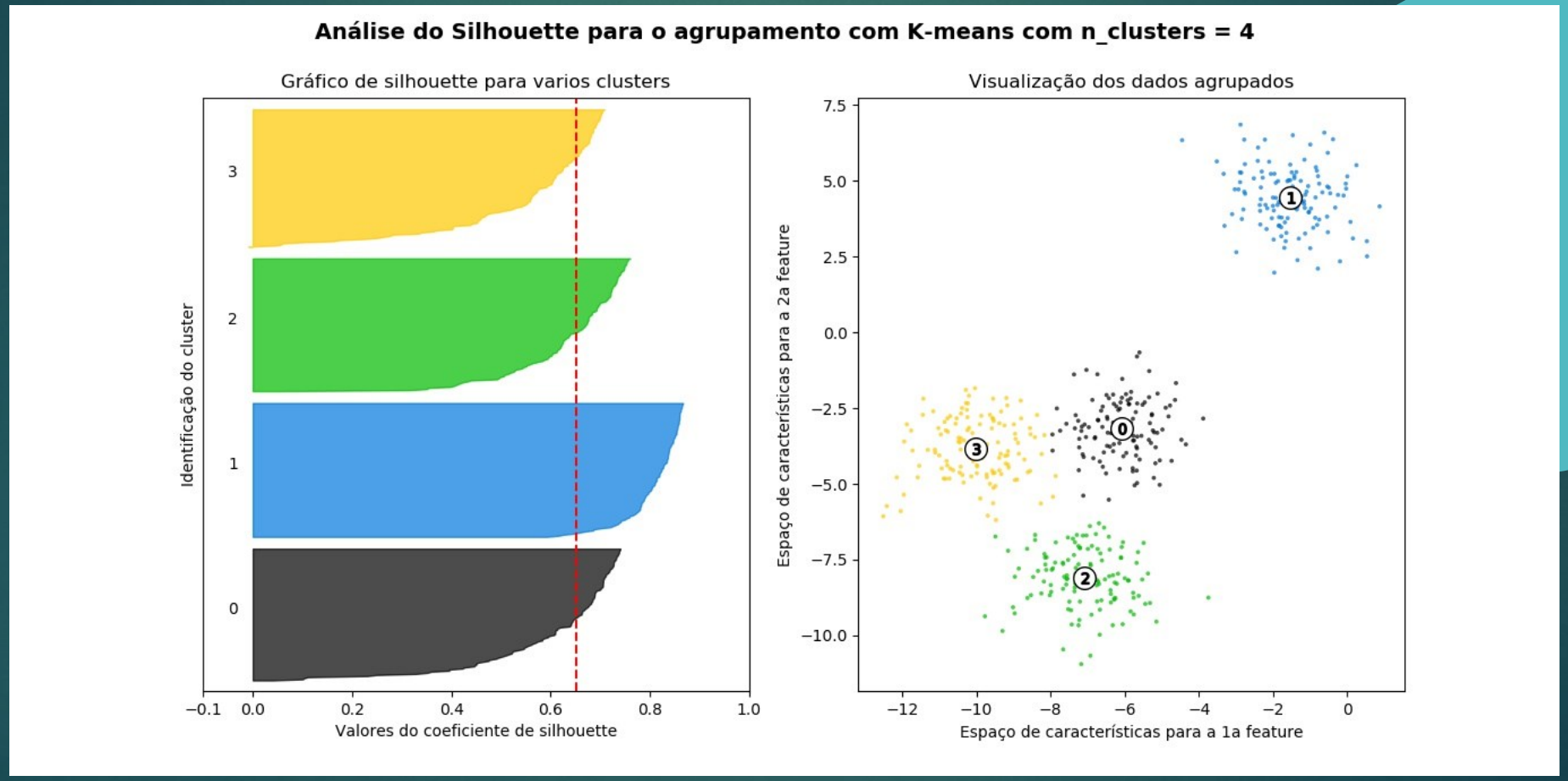
Prof.
H.S.Lopes
UTFPR/2024

Validação gráfica dos agrupamentos



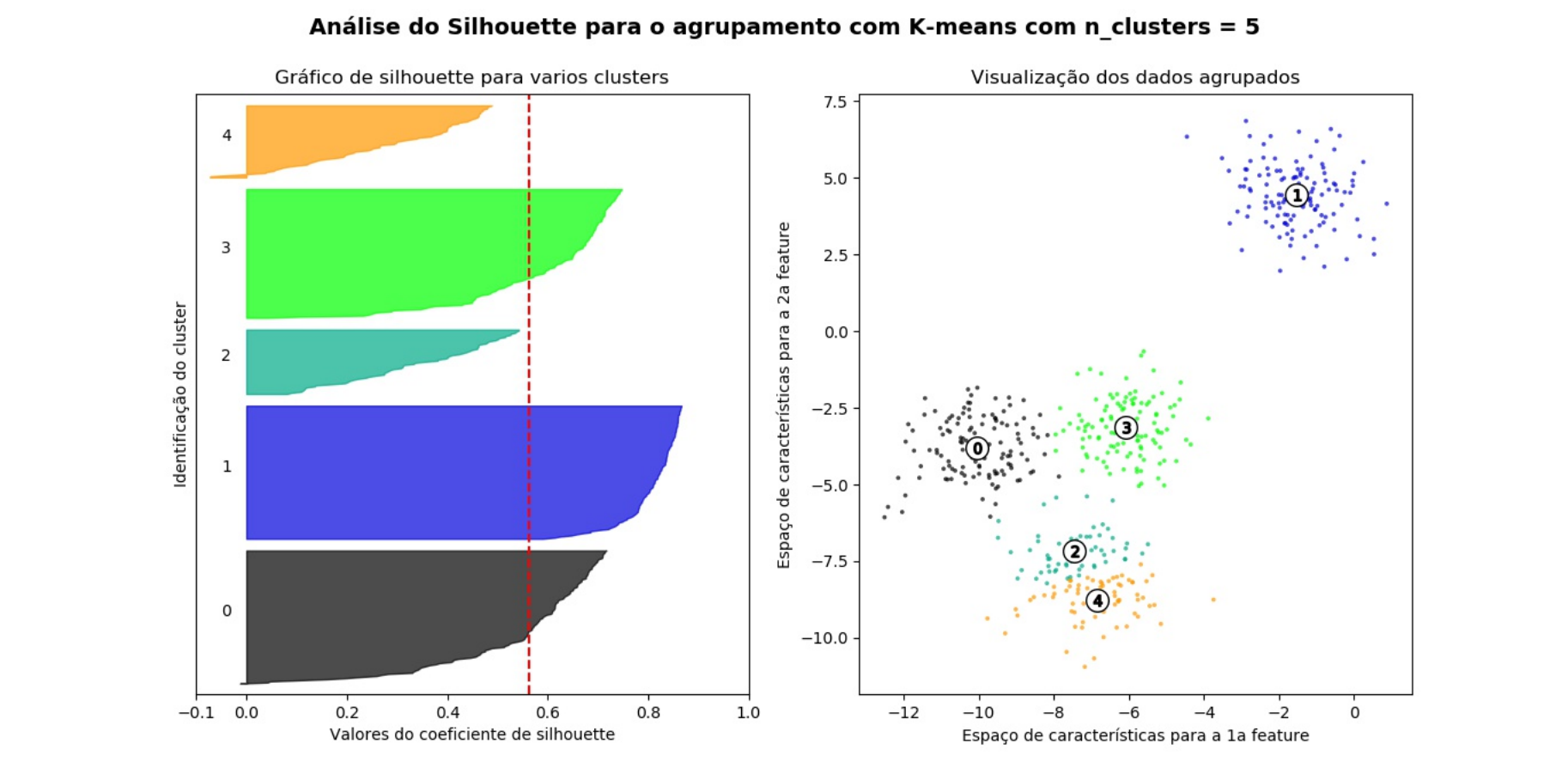
Prof.
H.S.Lopes
UTFPR/2024

Validação gráfica dos agrupamentos



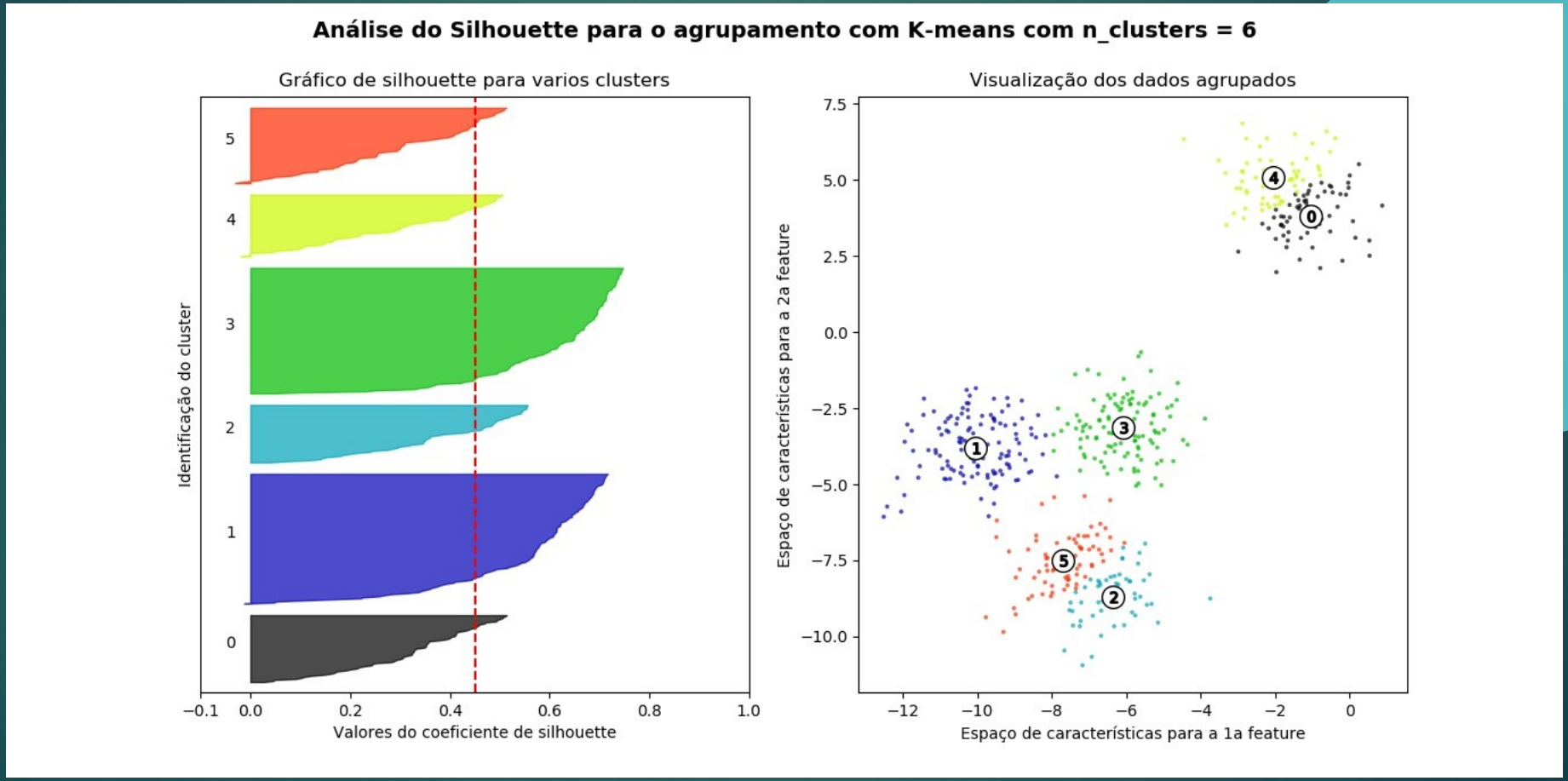
Prof.
H.S.Lopes
UTFPR/2024

Validação gráfica dos agrupamentos



Prof.
H.S.Lopes
UTFPR/2024

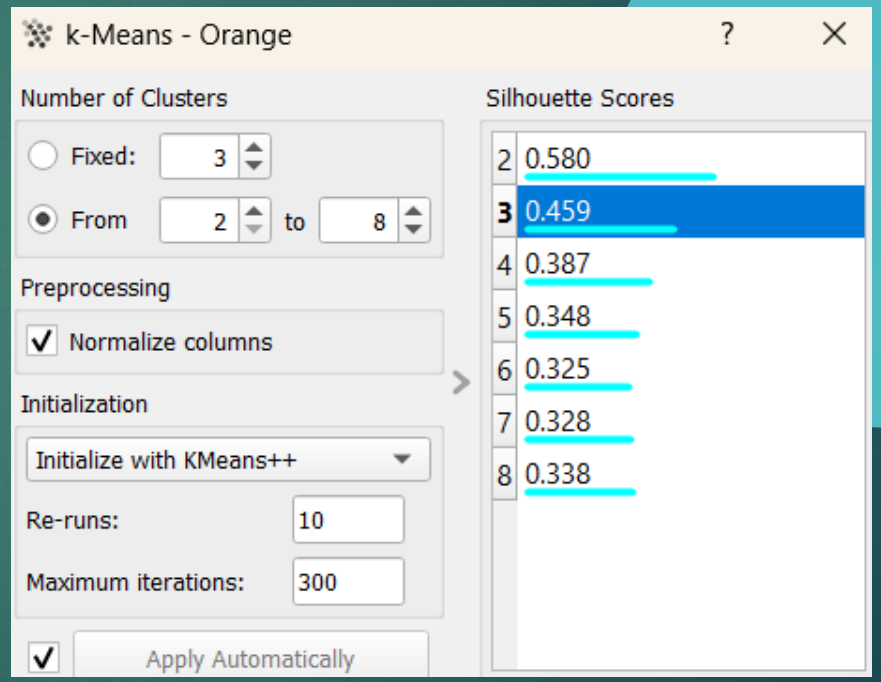
Validação gráfica dos agrupamentos



Prof.
H.S.Lopes
UTFPR/2024

Estudo de caso #1: Iris *dataset*

- Instâncias: 150 (50 por classe)
- Atributos previsores: 4 (*Petal width, Petal length, Sepal width, Sepal length*)
- Atributo meta: Classe {Iris setosa, Iris versicolor, Iris virginica}
- Agrupamento com K-means:

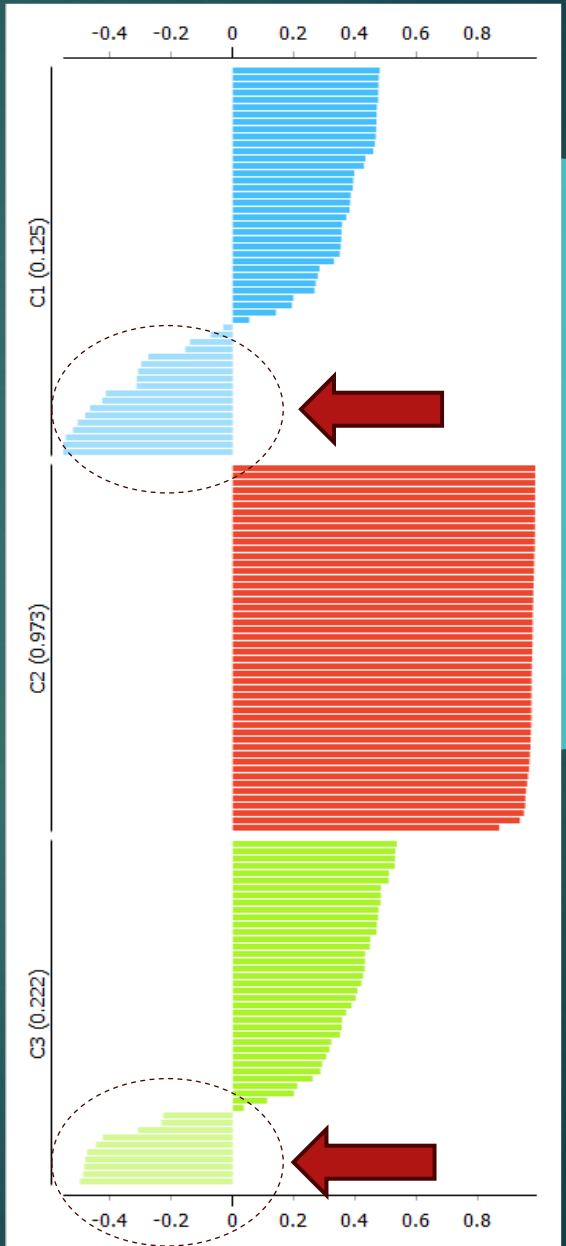


Cluster	Silhouette	sepal length	sepal width	petal length	petal width
C1	0.617263	-0.050	-0.880	0.348	0.282
C2	0.678779	-1.015	0.842	-1.305	-1.255
C3	0.604335	1.136	0.097	0.996	1.017

Estudo de caso #1: Iris *dataset*

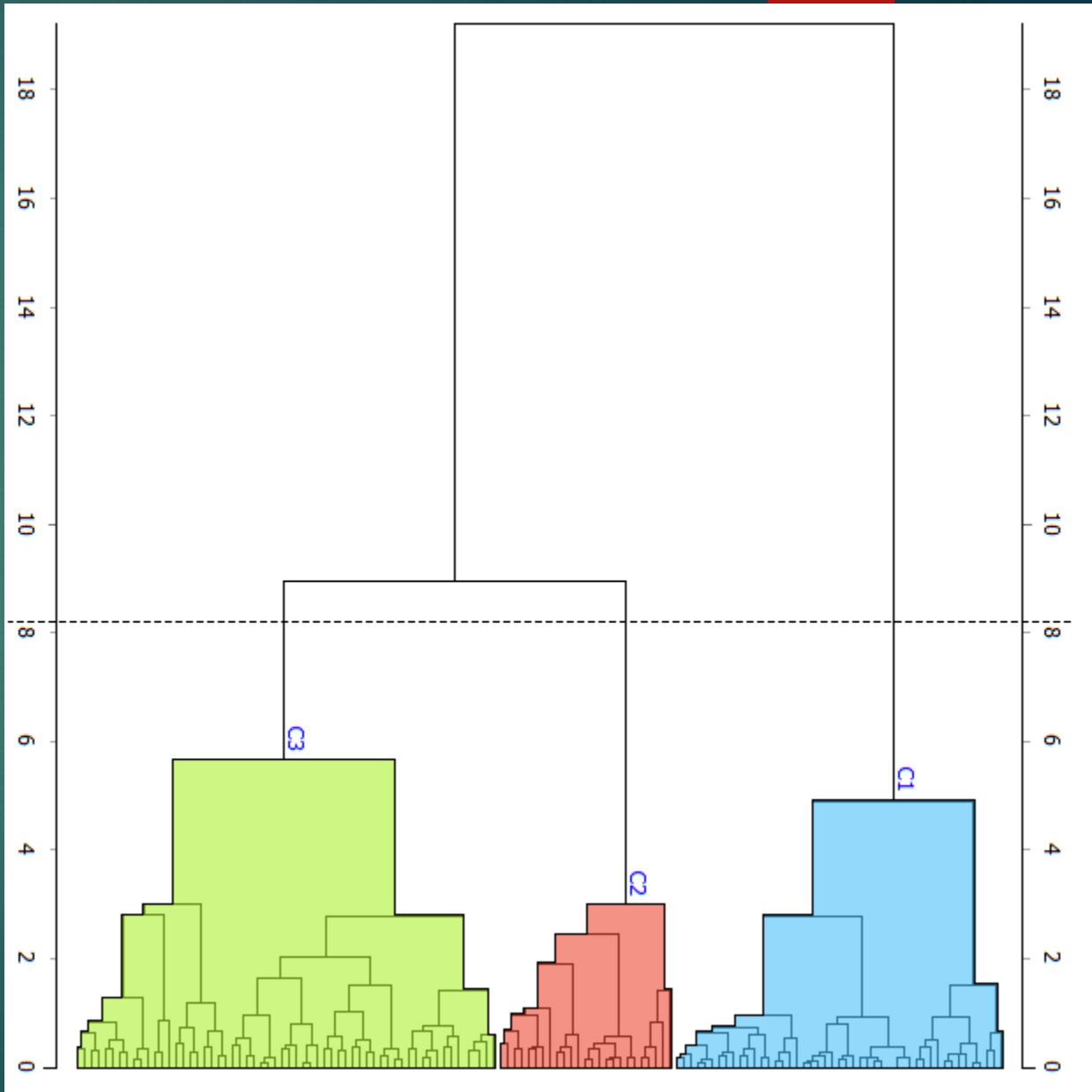
- Pontos não agrupados adequadamente
- Análise pelo coeficiente de silhoueta

	iris	Cluster	Silhouette	Silhouette (Cluster)	sepal length	sepal width	petal length	petal width
1	Iris-versicolor	C3	0.605059	-0.498472	7.0	3.2	4.7	1.4
2	Iris-versicolor	C3	0.553059	-0.44656	6.4	3.2	4.5	1.5
3	Iris-versicolor	C3	0.609772	-0.474834	6.9	3.1	4.9	1.5
4	Iris-versicolor	C3	0.573124	-0.308499	6.3	3.3	4.7	1.6
5	Iris-versicolor	C3	0.559875	-0.483836	6.7	3.1	4.4	1.4
6	Iris-versicolor	C1	0.618671	-0.0309238	5.6	3.0	4.5	1.5
7	Iris-versicolor	C1	0.611986	-0.0716797	6.3	2.5	4.9	1.5
8	Iris-versicolor	C3	0.516608	-0.488066	6.6	3.0	4.4	1.4
9	Iris-versicolor	C3	0.512713	-0.423353	6.8	2.8	4.8	1.4
10	Iris-versicolor	C3	0.605004	-0.226339	6.7	3.0	5.0	1.7
11	Iris-versicolor	C1	0.61851	-0.425572	6.0	2.7	5.1	1.6
12	Iris-versicolor	C1	0.622145	-0.154672	5.4	3.0	4.5	1.5
13	Iris-versicolor	C3	0.535586	-0.232317	6.0	3.4	4.5	1.6
14	Iris-versicolor	C3	0.58706	-0.481129	6.7	3.1	4.7	1.5
15	Iris-virginica	C1	0.605492	-0.55277	5.8	2.7	5.1	1.9
16	Iris-virginica	C1	0.639931	-0.505398	4.9	2.5	4.5	1.7
17	Iris-virginica	C1	0.615736	-0.544429	5.7	2.5	5.0	2.0
18	Iris-virginica	C1	0.518725	-0.481349	5.8	2.8	5.1	2.4
19	Iris-virginica	C1	0.625578	-0.298597	6.0	2.2	5.0	1.5
20	Iris-virginica	C1	0.598039	-0.520664	5.6	2.8	4.9	2.0
21	Iris-virginica	C1	0.556318	-0.313166	6.3	2.7	4.9	1.8
22	Iris-virginica	C1	0.555137	-0.275182	6.2	2.8	4.8	1.8
23	Iris-virginica	C1	0.561494	-0.139557	6.3	2.8	5.1	1.5
24	Iris-virginica	C1	0.605896	-0.313471	6.1	2.6	5.6	1.4
25	Iris-virginica	C1	0.526926	-0.308363	6.0	3.0	4.8	1.8
26	Iris-virginica	C1	0.605492	-0.55277	5.8	2.7	5.1	1.9
27	Iris-virginica	C1	0.572464	-0.413222	6.3	2.5	5.0	1.9
28	Iris-virginica	C1	0.530689	-0.464973	5.9	3.0	5.1	1.8



Estudo de caso #1: Iris *dataset*

- Agrupamento hierárquico:



3- Agrupamento baseado em densidade

- Busca no espaço de dados por regiões de densidades variadas. Isola regiões e atribui pontos de dados a estas regiões dentro do mesmo *cluster*.

- Principais algoritmos baseados em densidade:
  - DBSCAN
  - OPTICS

Algoritmo DBSCAN

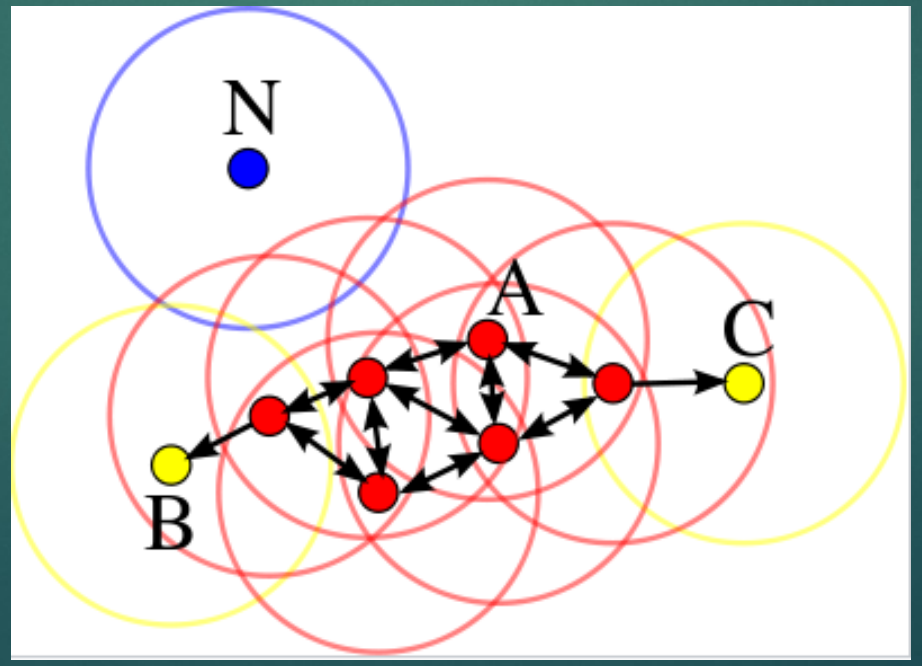
- DBSCAN foi criado por Ester et. al. (1996)
  - Dado um conjunto de pontos no espaço, o algoritmo agrupa os pontos que possam ser "empacotados" juntos, isto é, pontos com muitos vizinhos próximos.
  - Pontos que caem remotamente em regiões de baixa densidade (seus vizinhos estão distantes) são considerados *outliers*.
  - A complexidade temporal do algoritmo é ***O(n.log n)*** e de memória de ***O(n²)***
  - Python: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

Algoritmo DBSCAN

- O algoritmo requer dois parâmetros:
  - *eps* (ε) : raio máximo na vizinhança de um ponto *p*, isto é, os pontos que são alcançáveis por *p*
  - *minPts*: número mínimo de pontos necessários para formar uma região densa.
- Para o agrupamento, os pontos podem ser considerados como núcleos (*core points*), pontos alcançáveis por densidade (*density-reachable points*) ou *outliers*
  - Um ponto é considerado *core point* se pelo menos *minPts* estiverem a uma distância (ε) deste ponto
  - Um ponto *q* é considerado alcançável por *p* se houver um caminho entre os dois com *core points*
  - Pontos não alcançáveis a partir de nenhum ponto são *outliers*

Algoritmo DBSCAN

- Seja *minPts* = 4. Ponto A e os em vermelho são *core points* (estão dentro de um raio (ε) do ponto) pois todos são alcançáveis entre si
- Pontos B e C <u>não</u> são core points, mas são alcançáveis por A, então pertencem a este *cluster*
- Ponto N é um ruído (*outlier*) pois não é alcançável de nenhum *core point*



K-means versus DBSCAN

IMPORTANTE

Prof.
H.S.Lopes
UTFPR/2024

- Diferença:
  - K-*means* agrupa todos os objetos
  - DBSCAN não agrupa aqueles que julga serem *outliers*
- Conceito:
  - K-*means*: protótipo de um grupo
  - DBSCAN: baseado em densidade
- Adequabilidade:
  - K-*means* lida bem com grupos globulares e grupos com tamanhos similares.
  - DBSCAN lida bem com grupos de tamanhos e formatos diferentes e é pouco afetado por ruídos.
- A complexidade de tempo:
  - K-*means* é ***O****(n.K.I.d)*
  - DBSCAN é ***O****(n.log n)*

K-means versus DBSCAN

IMPORTANTE

Prof.
H.S.Lopes
UTFPR/2024

- K-*means* pode ser aplicado quando há dados esparsos de alta dimensionalidade, mas DBSCAN geralmente tem problema para isto pois a métrica Euclidiana não funciona bem para alta dimensionalidade
- K-*means* pode encontrar grupos que não estejam bem separados, mas DBSCAN funde grupos que possuam intersecção
- DBSCAN produz sempre o mesmos grupos em cada rodada, K-*means* não, pois depende da inicialização aleatória dos centroides

Estudo de caso #2: *Seeds dataset*

- Objetivo: Determinar a variedade de sementes de trigo com base nas propriedades geométricas das sementes obtidas por raios-X.
- Características:
  - Instâncias: 210
  - Atributos previsores: 7 (área, perímetro, compactação, comprimento do *kernel*, largura do *kernel*, coeficiente de assimetria, comprimento do *kernel groove*), todos valores reais.
  - Atributo meta: suprimido
  - Distribuição de classes: suprimido

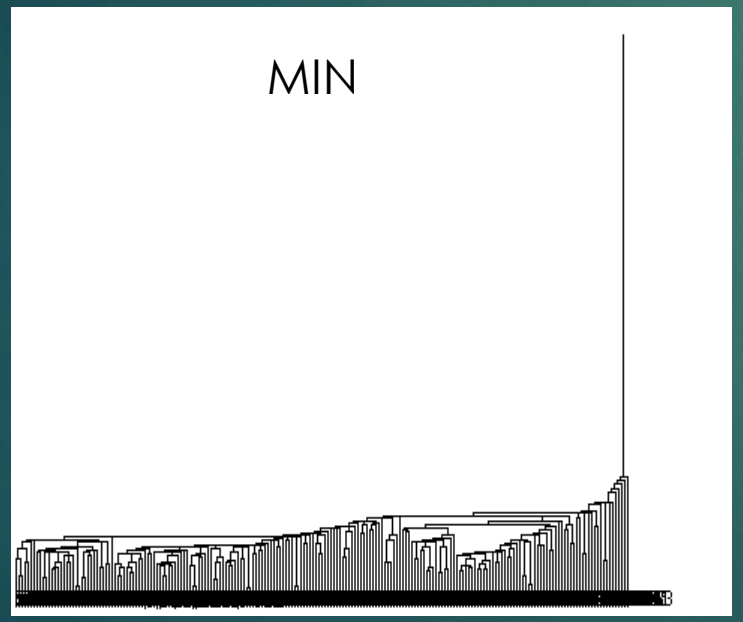
Estudo de caso #2: *Seeds dataset*

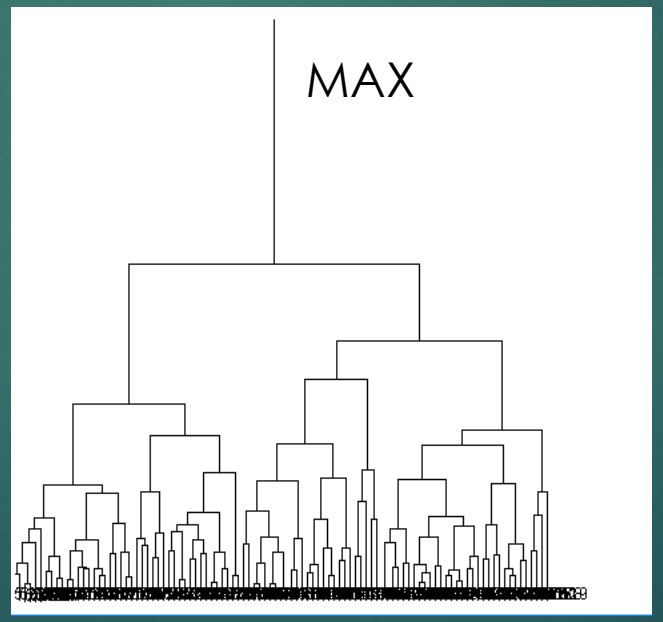
- **K-means** com parâmetros padrão e com inicialização aleatória (mesma semente).
- Cada célula representa porcentagem dos dados em cada *cluster*

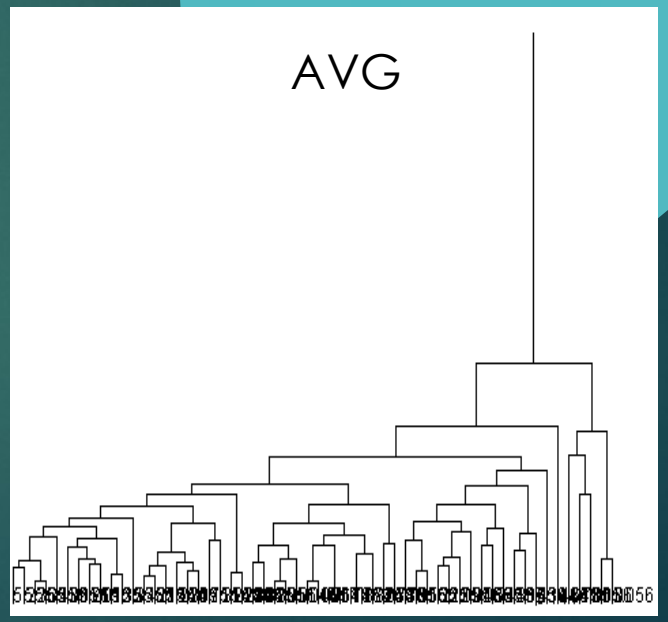
	2	3	4	5	6	7
Cluster #0	36%	33%	14%	11%	5%	5%
Cluster #1	64%	37%	29%	12%	10%	10%
Cluster #2		30%	21%	11%	9%	9%
Cluster #3			36%	36%	36%	23%
Cluster #4				29%	29%	25%
Cluster #5					11%	11%
Cluster #6						18%

Estudo de caso #2: *Seeds dataset*

- Agrupamento hierárquico usando parâmetros padrão e as estratégias: MIN, **MAX** e AVG:



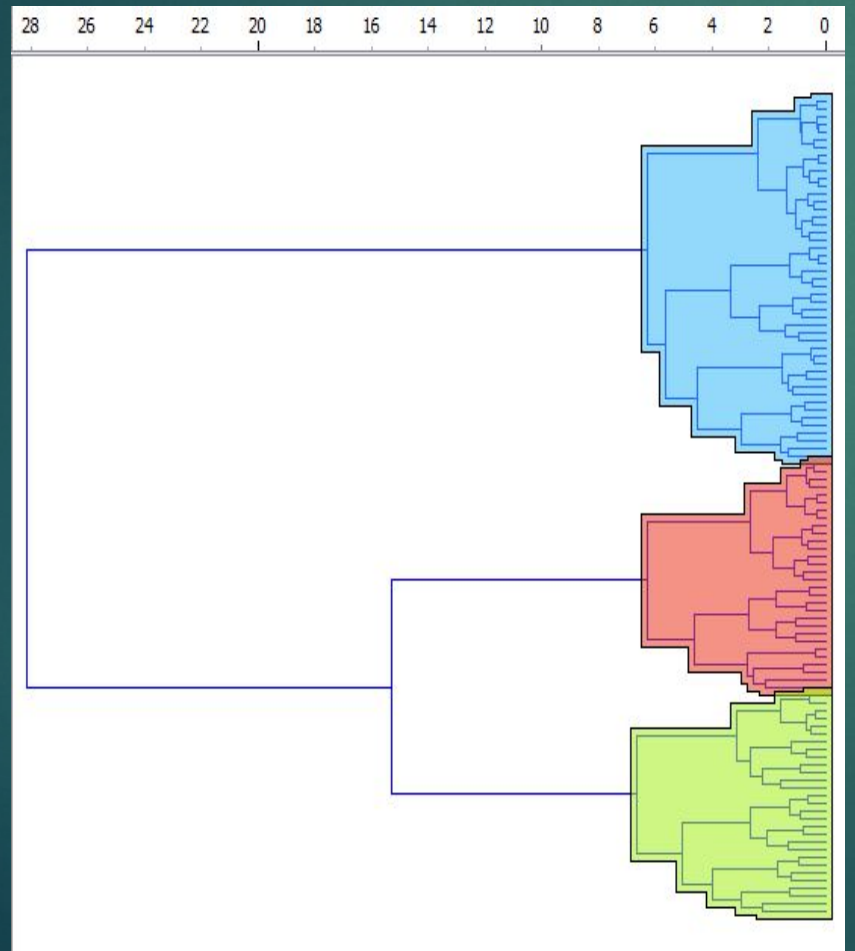


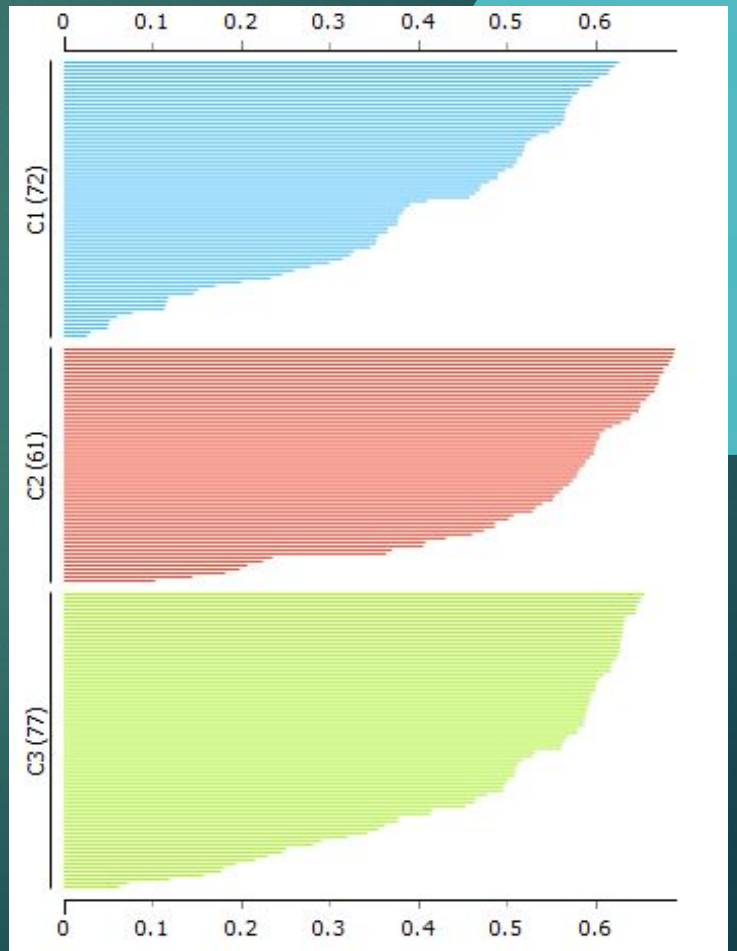


Prof.
H.S.Lopes
UTFPR/2024

Estudo de caso #2: *Seeds dataset*

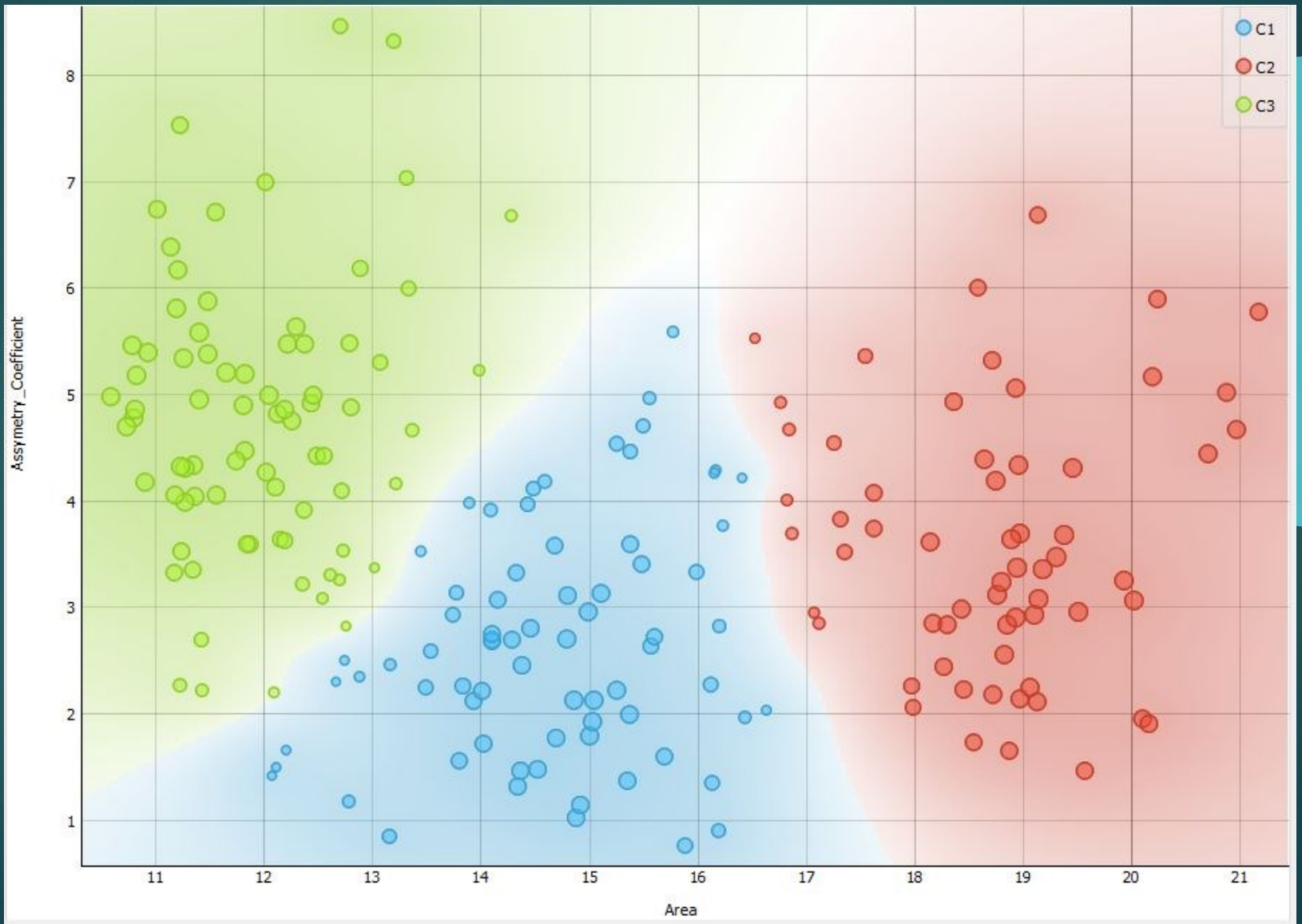
- O gráfico do coeficiente de silhoueta corrobora o número de 3 clusters





Estudo de caso #2: *Seeds dataset*

Prof.
H.S.Lopes
UTFPR/2024

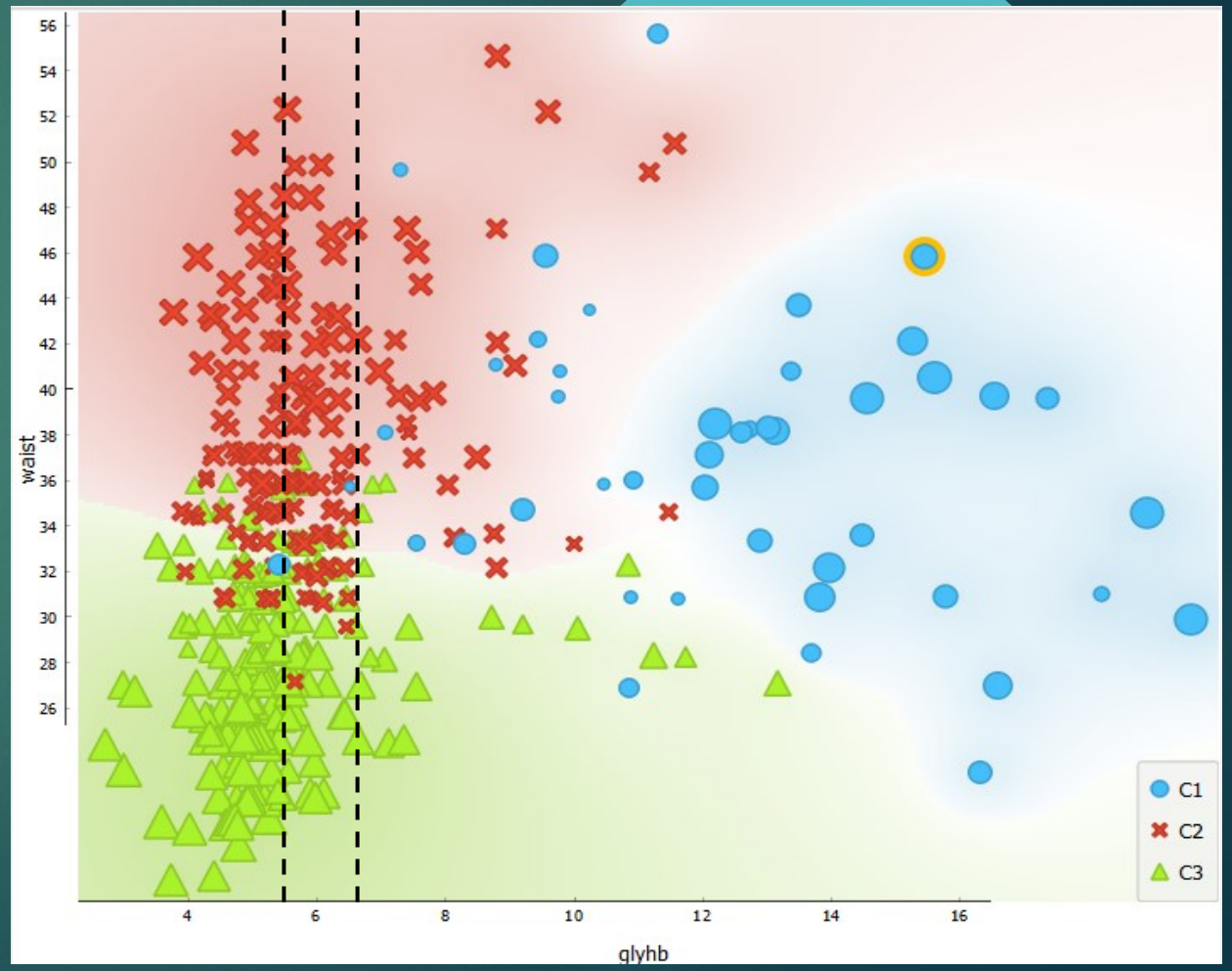


Estudo de caso #3: *Prevalência de diabetes em obesos*

- Fonte: Dr. J. Schorling, Dept. of Medicine, University of Virginia, School of Medicine.
- Objetivo: O Diabetes Mellitus tipo II (DM-II) em adultos é fortemente associado à **obesidade**. A relação entre as medidas da cintura e do quadril pode ser um bom preditor do diabetes e de doenças cardiovasculares. DM II também está relacionado com a **hipertensão**. A dosagem de hemoglobina glicosada > 6,5% geralmente é tomada como um indicador positivo de diabetes.
- Características:
  - Participantes: 403 indivíduos de diferentes localizações
  - Variáveis: colesterol, glicose, HDL, relação cintura/quadril, hemoglobina glicosada, idade, gênero, estatura, pressão arterial, etc

Estudo de caso #3: *Prevalência de diabetes em obesos*

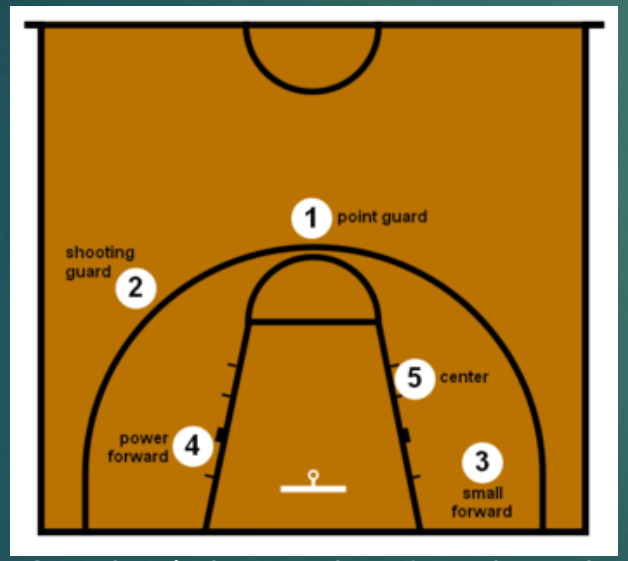
- Os pontos em azul correspondem ao <u>grupo de risco</u>: obesos que têm a hemoglobina glicosada maior do que 6,5%

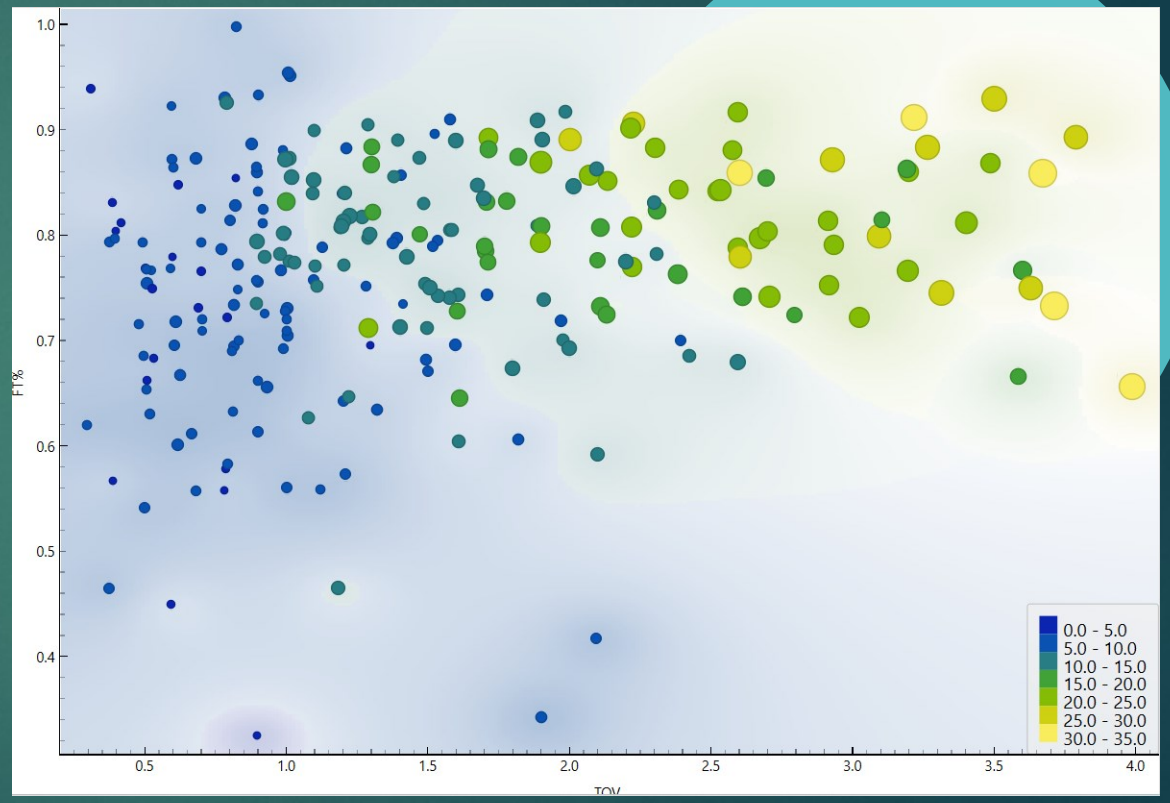


Prof.
H.S.Lopes
UTFPR/2024

Estudo de caso #4: jogadores de basquete da NBA

- Contém 16 variáveis sobre jogadores de basquete: Position, Team, Age, Height, Weight, Hand, 3P%, 2P%, FT%, TRB, AST, STL, BLK, TOV, PF, PTS





- Qual a interpretação dos clusters?

Prof.
H.S.Lopes
UTFPR/2024

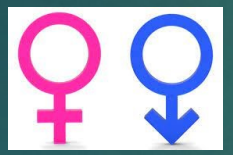
Estudo de caso #5: agrupamento de *imagens de faces*

- Quantos grupos podem ser identificados?
- Quais características devem ser usadas para separar as imagens em grupos ?
  - Parâmetros objetivos e subjetivos
- É possível separar imagens reais de imagens fake ?

Prof.
H.S.Lopes
UTFPR/2024

Estudo de caso #5: agrupamento de *imagens de faces*

- Critério #1: Gênero aparente



Prof.
H.S.Lopes
UTFPR/2024

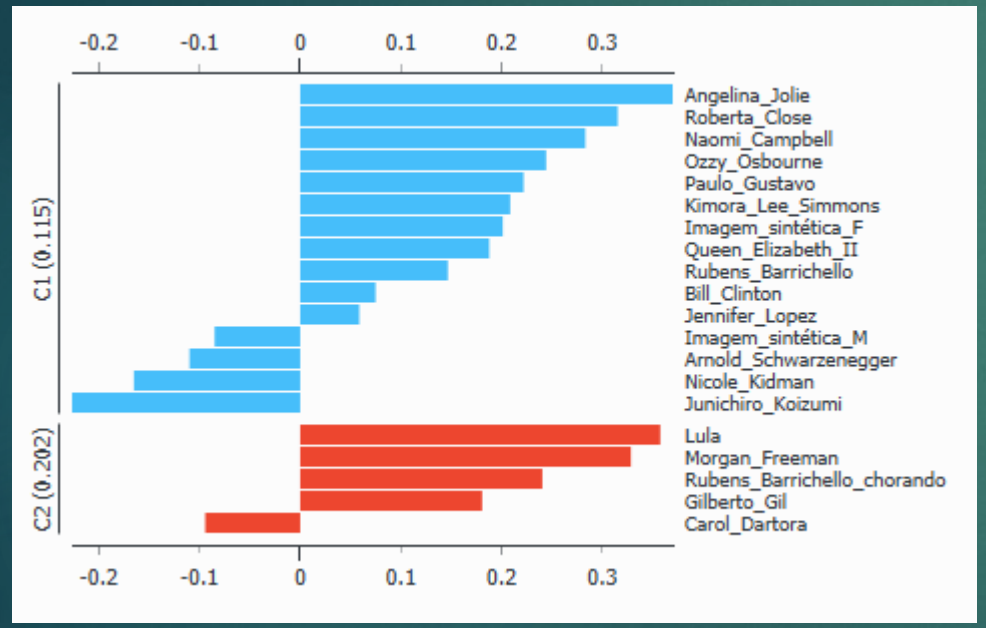
Estudo de caso #5: agrupamento de *imagens de faces*

- Critério 2: artista ou político ?

Prof.
H.S.Lopes
UTFPR/2024

Estudo de caso #5: agrupamento de *imagens de faces*

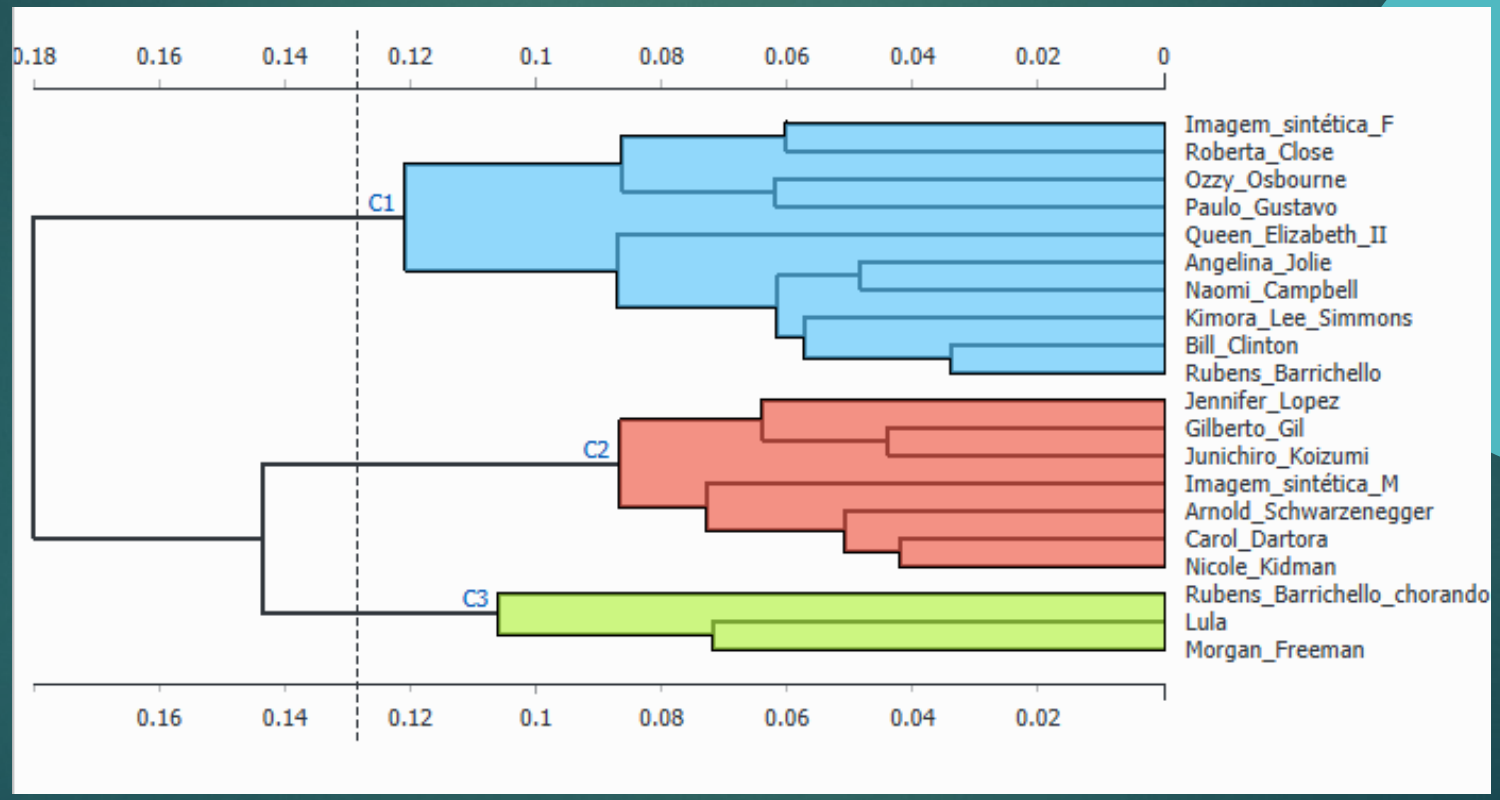
K-means versus Louvain *clustering*



Prof.
H.S.Lopes
UTFPR/2024

Estudo de caso #5: agrupamento de *imagens de faces*

- Agrupamento hierárquico (*linkage=complete*)



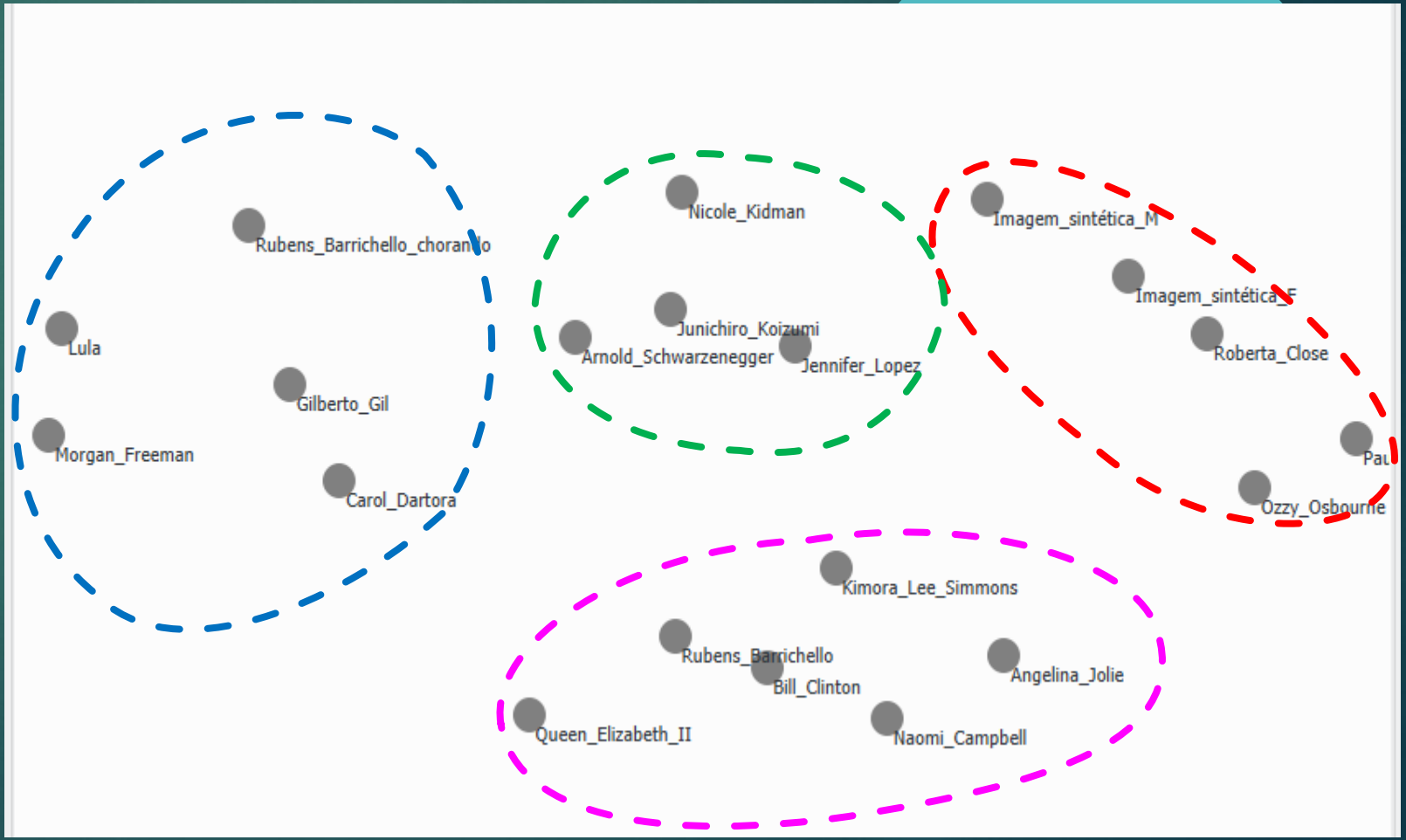
- Qual a interpretação dos clusters?

Estudo de caso #5: agrupamento de *imagens de faces*

Prof.
H.S.Lopes
UTFPR/2024

- Agrupamento com T-SNEE (t-Distributed Stochastic Neighbor Embedding)

- Faz sentido os clusters ?



Estudo de caso #5: agrupamento de *imagens de faces*

Prof.
H.S.Lopes
UTFPR/2024

- Este agrupamento em duas dimensões faz algum sentido ?
- E se fosse em 3 ou mais dimensões ?