ICC204 - Aprendizagem de Máquina e Mineração de Dados

Agrupamento

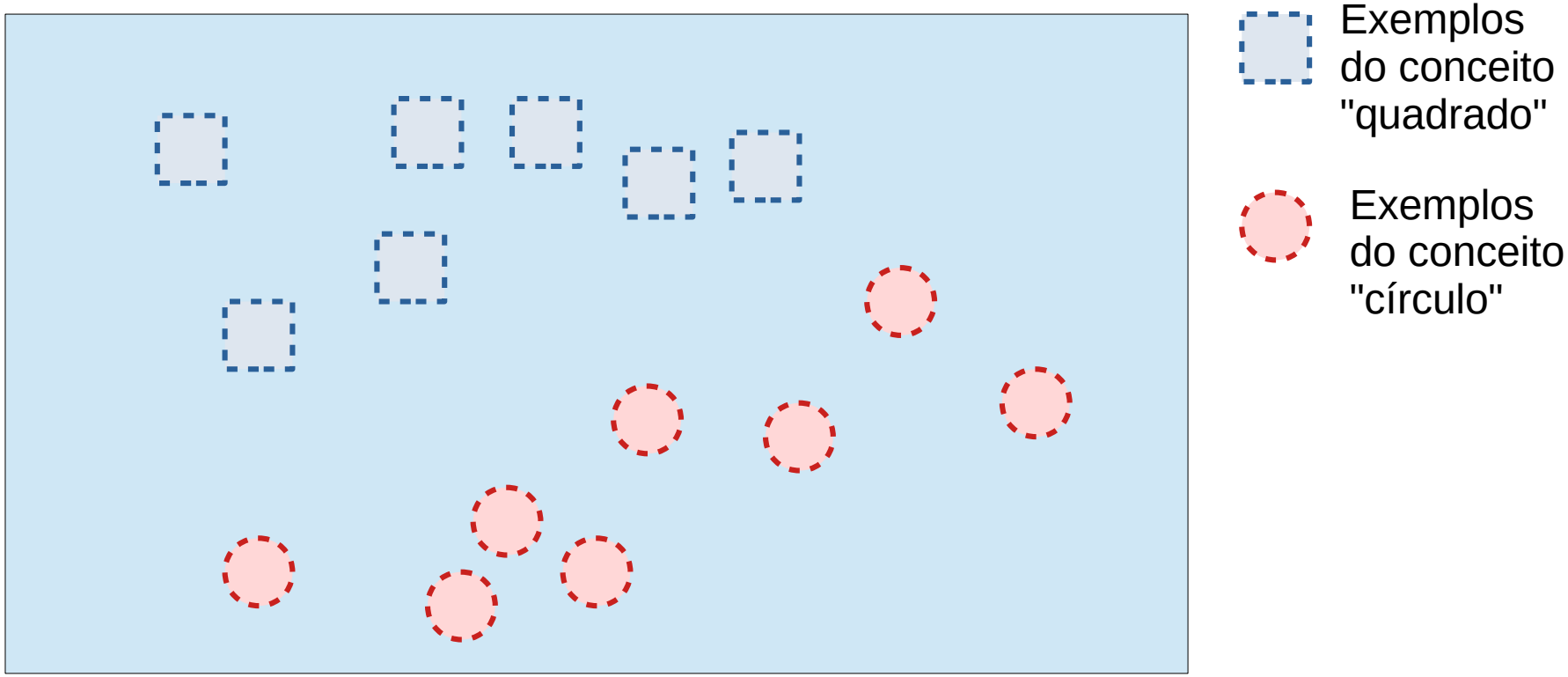
Prof. Rafael Giusti
rgiusti@icomp.ufam.edu.br

Agenda

- Aprendizado não supervisionado
- Grupos e tipos de agrupamento
- Algoritmos de agrupamento
- O k-*means*
- Algoritmos sequenciais
- Agrupamento por densidade – DBSCAN
- Agrupamento hierárquico
- Qualidade de agrupamentos

Aprendizado não supervisionado

- No aprendizado **não supervisionado**, o conceito existe, mas os exemplos **não são rotulados**; o objetivo é encontrar uma estrutura nos dados

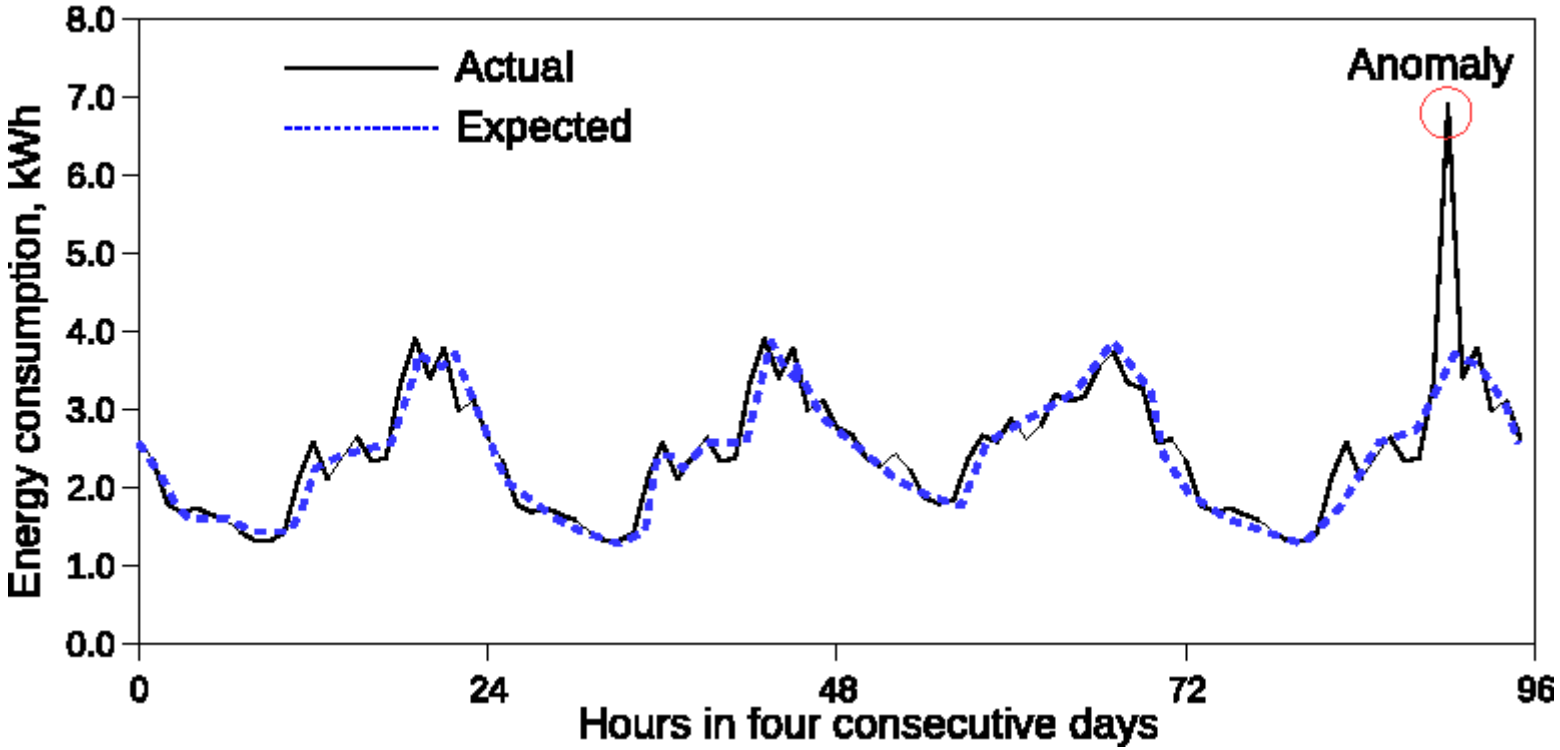


Aprendizado não supervisionado

- Tarefas de aprendizado não supervisionado incluem
  - Agrupamento
  - Detecção de anomalias
  - Redução de dimensionalidade

Aprendizado não supervisionado

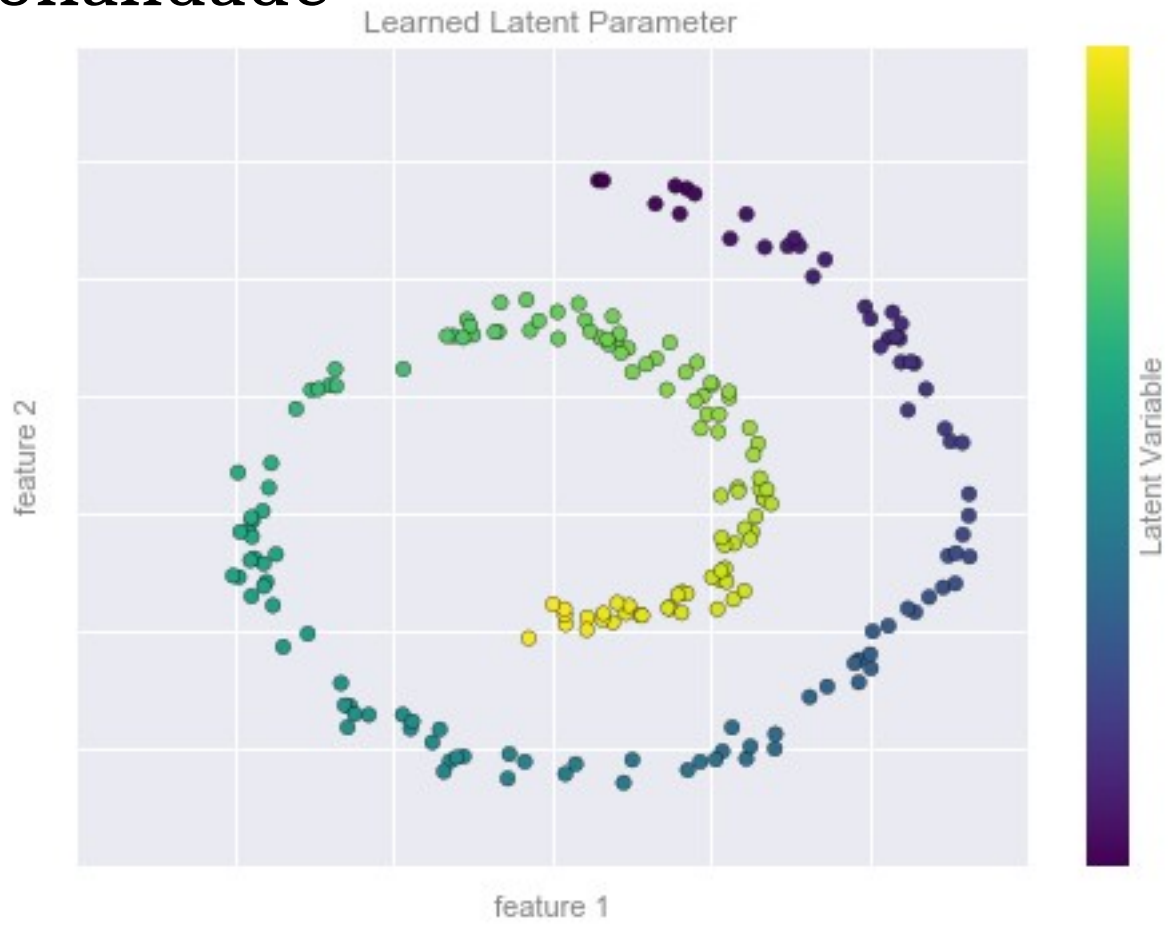
- Na **detecção de anomalias**, os dados seguem um padrão e exemplos que fogem substancialmente do padrão são considerados anômalos



Xiufeng Liu, Per Sieverts Nielsen. Regression-based Online Anomaly Detection for Smart Grid Data. ArXiv (2016)

Aprendizado não supervisionado

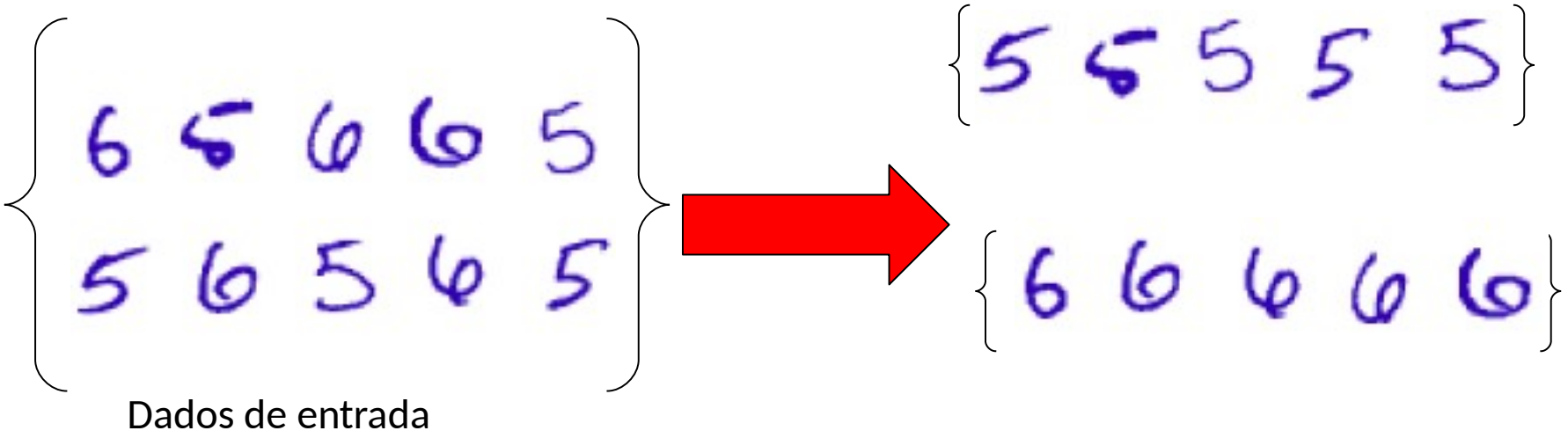
- Na **redução de dimensionalidade**, queremos dados com muitas dimensões em um espaço de menor dimensionalidade



Fonte: Jake VanderPlas. *Python Data Science Handbook.*

Aprendizado não supervisionado

- No **agrupamento**, existe uma estrutura de grupos nos dados que queremos encontrar



Dados de entrada

Aplicações de agrupamento

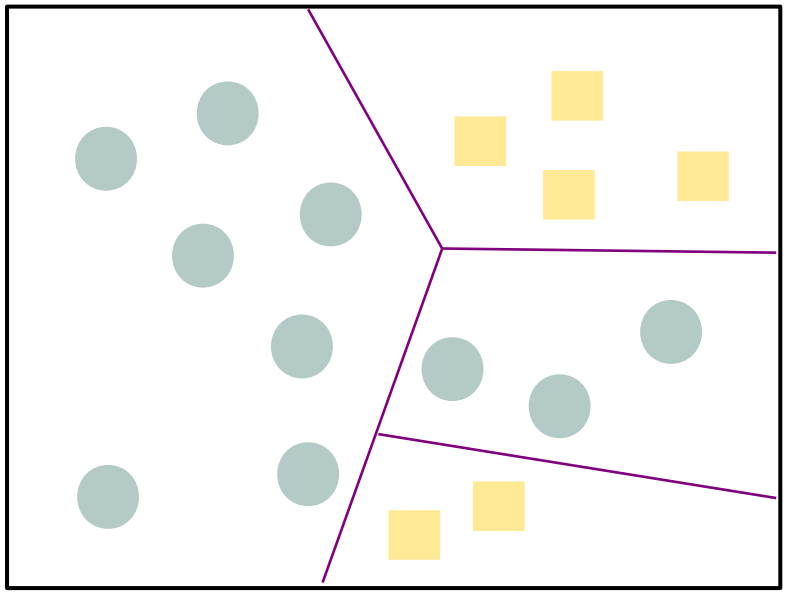
- Reconhecimento de padrões em imagens
- Organizar clientes/usuários de um serviço
- Distribuir os dados em classes "naturais", mas desconhecidas
- Ganhar percepção da natureza / estrutura dos dados
- Rotualação dos dados
  - Aplique agrupamento em uma grande quantidade de dados e só então "use supervisão" para rotular dados de um grupo

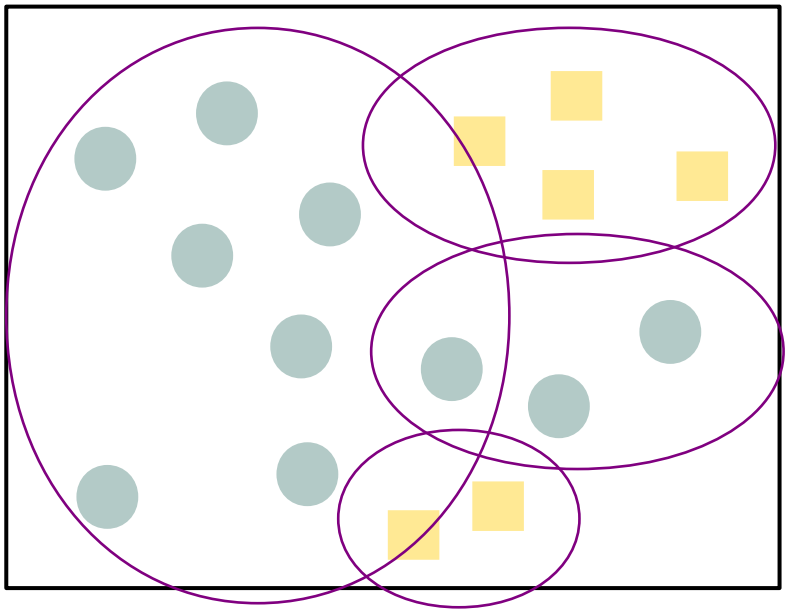
Agenda

- Aprendizado não supervisionado
- Grupos e tipos de agrupamento
- Algoritmos de agrupamento
- O k-*means*
- Algoritmos sequenciais
- Agrupamento por densidade – DBSCAN
- Agrupamento hierárquico
- Qualidade de agrupamentos

Agrupamento rígido ou *fuzzy*

- No agrupamento **rígido**, os grupos são conjuntos de exemplos disjuntos
- No agrupamento ***fuzzy***, pode haver sobreposição entre os grupos



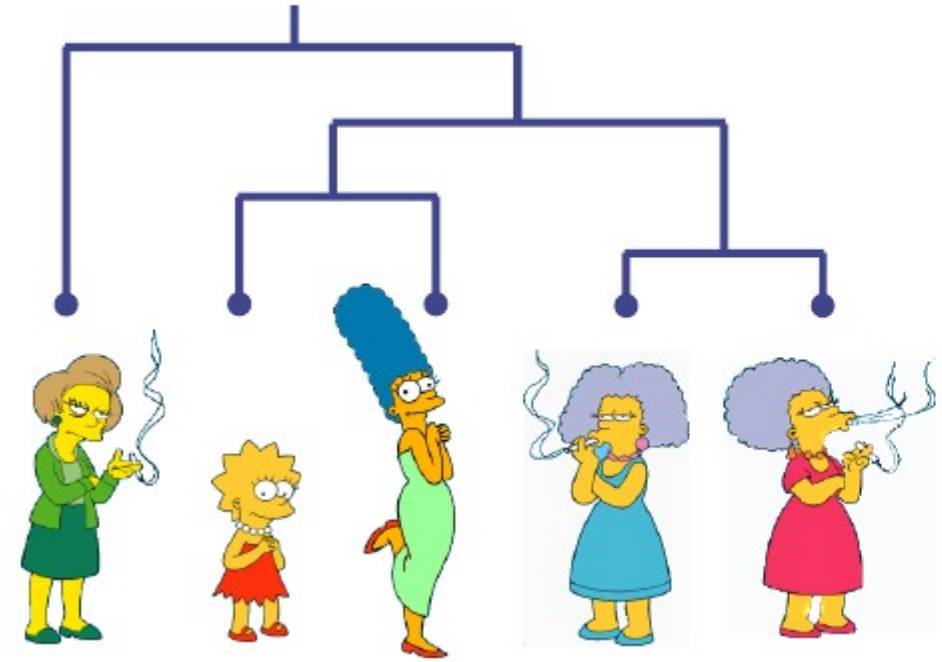


Agrupamento particional ou hierárquico

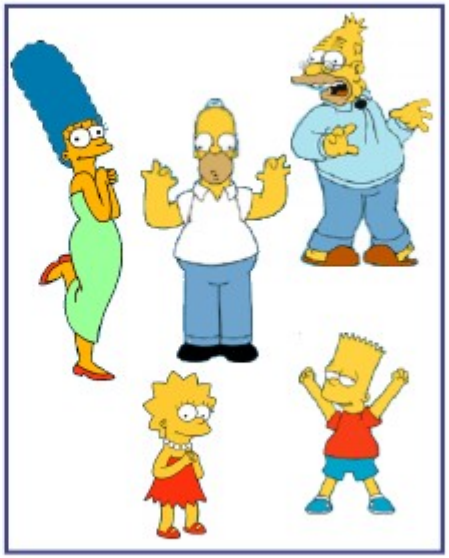
- No agrupamento **particional**, os grupos formam um **particionamento** sobre os dados
  - É um tipo de agrupamento rígido
  - O agrupamento *fuzzy* é semelhante ao particional, mas os grupos não formam um particionamento
- No agrupamento **hierárquico**, os grupos formam uma hierarquia de grupos aninhados

Agrupamento particional ou hierárquico

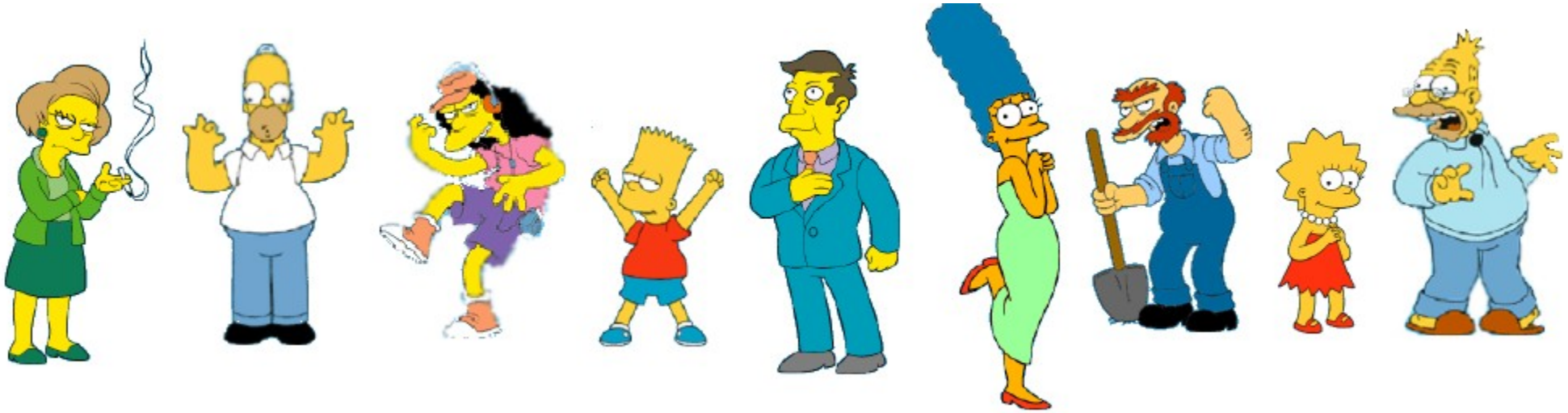
Hierárquicos

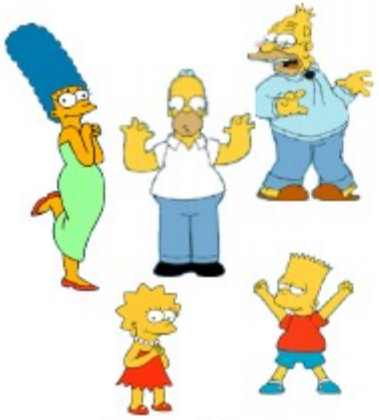


Particionais



Subjetividade do agrupamento





Família

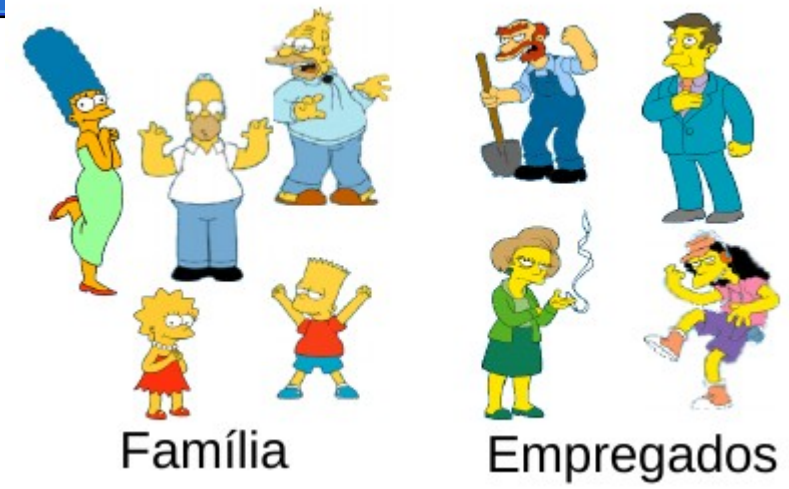
Empregados

Mulheres

Homens

Grupo é um conceito subjetivo

Função de (dis)similaridade



- Essa subjetividade precisa ser "traduzida" em uma função de similaridade ou dissimilaridade
  - Em um bom agrupamento, os exemplos devem ser bastante similares a exemplos do seu próprio grupo (alta **similaridade intra-grupo**)
  - Além disso, devem ser pouco similares a exemplos de outros grupos (baixa **similaridade inter-grupos**)

Agenda

- Aprendizado não supervisionado
- Grupos e tipos de agrupamento
- Algoritmos de agrupamento
- O k-*means*
- Algoritmos sequenciais
- Agrupamento por densidade – DBSCAN
- Modelos de misturas
- Agrupamento hierárquico
- Qualidade de agrupamentos

Algoritmos de agrupamento

- O processo de agrupamento segue, em linhas gerais
  1. Seleção de características
  2. Definição de uma medida de (dis)similaridade
  3. Aplicação do algoritmo de agrupamento
  4. Verificação dos resultados
     - Se os resultados forem pobres, podemos retornar para o passo 1, 2 ou mesmo 3
  5. Interpretação dos resultados

Especificação do problema

- Dado um conjunto de exemplos $X = \{x_1, x_2, ..., x_N\}$, desejamos encontrar uma **estrutura de grupos** sobre X que reflete algum conceito de (dis)similaridade no qual estamos interessados
- Os resultados vão depender
  - Dos atributos / características utilizados para representar os exemplos
  - Do conceito de (dis)similaridade
  - Do algoritmo utilizado

Especificação dos algoritmos

- Algoritmos de agrupamento são, em geral, estocásticos
  - Podem produzir resultados diferentes para um mesmo conjunto de dados sob condições aparentemente iguais
- Tipos de algoritmos de agrupamento
  - Sequenciais
  - Baseados na otimização de funções de custo
  - Hierárquicos
  - Outros: algoritmos genéticos, auto-organizáveis etc.

Agenda

- Aprendizado não supervisionado
- Grupos e tipos de agrupamento
- Algoritmos de agrupamento
- O k-*means*
- Algoritmos sequenciais
- Agrupamento por densidade – DBSCAN
- Modelos de misturas
- Agrupamento hierárquico
- Qualidade de agrupamentos

k-means

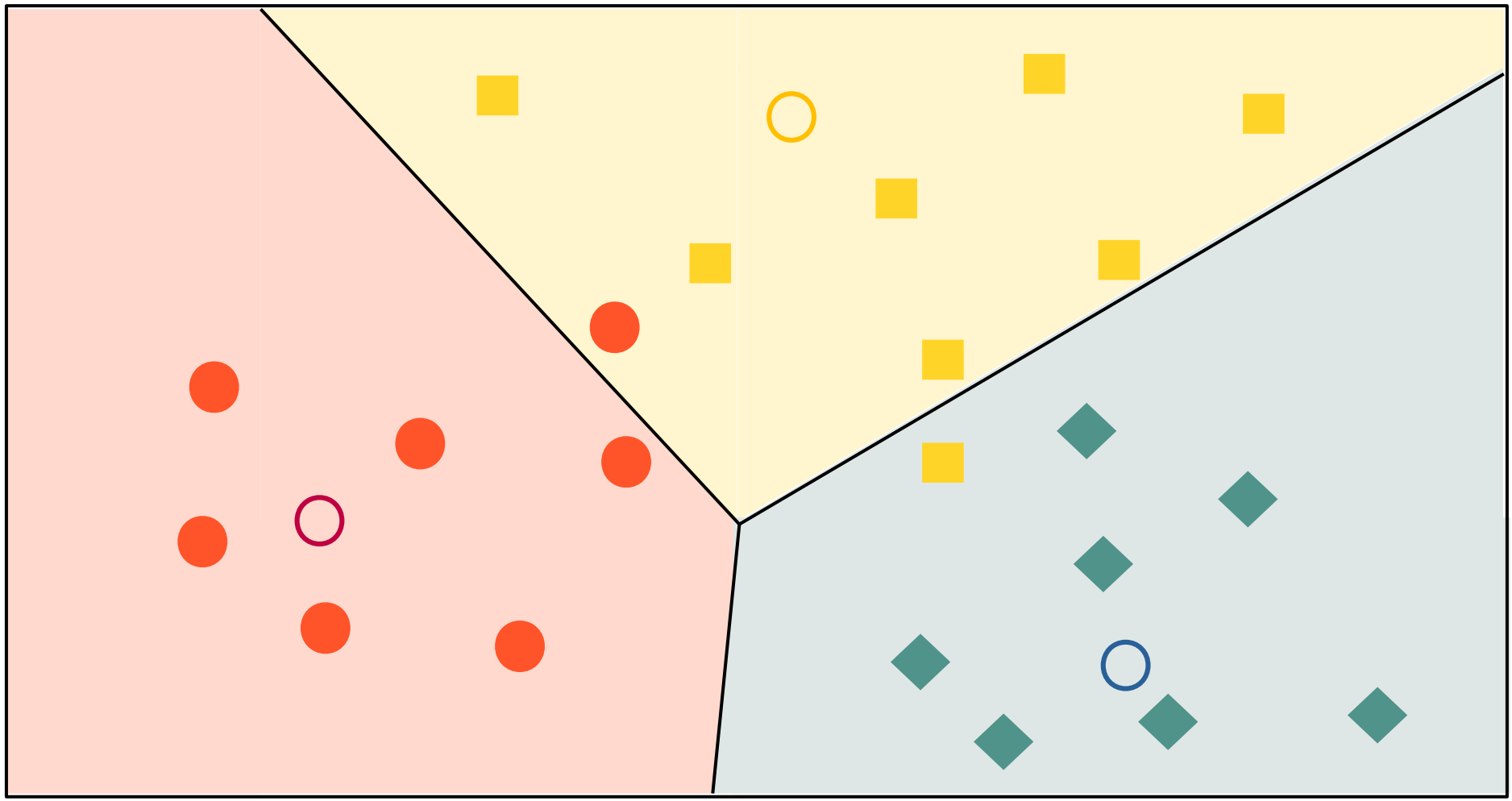
- Algoritmo de agrupamento particional
  - Rígido
  - Dado um conjunto de exemplos $X = (x_1, x_2, ..., x_N)$, o agrupamento rígido é um **particionamento** C sobre X
    - Isto é, $C = \{C_1, C_2, ..., C_M\}$ em que cada C_i é um grupo e
    - $C_i \neq \emptyset$
    - $C_i \cap C_j = \emptyset$ para $i \neq j$
    - $\bigcup C_i = X$

k-means

- O *k-means* particiona os exemplos em *k* grupos disjuntos, determinados por um **centroide**
  - Os centroides não necessariamente são exemplos do conjunto de treinamento
    - Mais que isso, o centroide pode possuir valores que não caracterizam uma instância válida
  - Dado um exemplo *x* e os centroides c_1, c_2, ..., c_k, o exemplo *x* pertencerá ao grupo C_i tal que
    - $i = \text{argmin } d(x, c_j)$

Células de Voronoi

- Os centroides dividem o espaço em **células de Voronoi**



k-means

- **Algoritmo** kMeans(**X**, *k*)

  selecione arbitrariamente *k* centroides c_1, c_2, ..., c_k

  C ← particionamento vazio de k grupos

  repita

    para cada *x* **em** **X**

      identifique o centroide mais próximo c_i de *x*

      atribua *x* ao grupo C_i no particionamento C

    para i **de** 1 **até** *k*

      atualize o centroide c_i para a média dos pontos em C_i

  até convergir

  retorne C

Agenda

- Aprendizado não supervisionado
- Grupos e tipos de agrupamento
- Algoritmos de agrupamento
- O k-*means*
- Algoritmos sequenciais
- Agrupamento por densidade – DBSCAN
- Agrupamento hierárquico
- Qualidade de agrupamentos

BSAS

- BSAS é um modelo geral para algoritmos de agrupamento sequencial
  - *Basic Sequential Algorithmic Scheme*
- Trata-se de um algoritmo que processa os exemplos sequencialmente, verificando se ele pode ser incluído no mesmo grupo de algum exemplo anteriormente processado

BSAS

- O BSAS utiliza como princípio a distância de um exemplo para um grupo
  - **Hiperparâmetros**
    - Θ: distância máxima entre um exemplo e um grupo
    - q: o número máximo de grupos
  - **Princípio**
    - Para cada exemplo x_i, calcule a menor distância entre x_i e algum grupo C_k, se $d(x_i, C_k) > \Theta$, crie um novo grupo; senão, inclua x_i em C_k

BSAS

- **Algoritmo** BSAS(X, d, Θ, q)

  m ← 1, C1 ← { x_1 }

  para i **de** 2 **até** |X|

    C_k ← grupo que minimiza a d(x_i, C_j) **para** 1 <= j <= m

    se d(x_i, C_k) > Θ **e** m < q, **então**

      m ← m + 1

      Cm ← {xi}

    senão

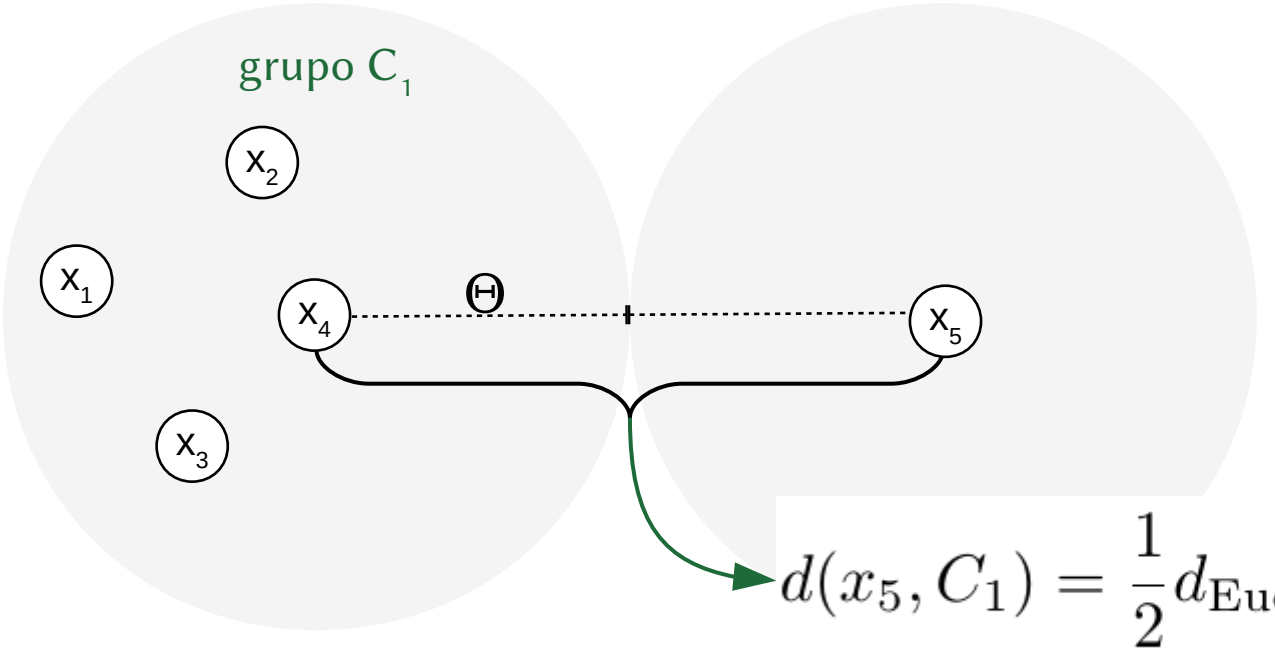
      C_k ← C_k U {x_i}

      se necessário, atualize representantes

BSAS: exemplo

- Vamos definir como distância entre um exemplo e um grupo

$$d(x_i, C_j) = \frac{1}{2} \min_{x_j \in C_j} d_{\text{Euc}}(x_i, x_j)$$



$$d(x_5, C_1) = \frac{1}{2} d_{\text{Euc}}(x_5, x_4) = \theta$$

BSAS: exemplo

BSAS

- Os hiperparâmetros influenciam muito o resultado do algoritmo
  - A ordem em que os exemplos são apresentados
  - O limiar Θ
  - O número máximo de grupos q
- Para definir o limiar Θ ideal, dependemos de validação
- Para definir o número máximo de grupos q, podemos fazer vários testes e escolher o mais frequente

BSAS: escolha do número de grupos

- **Algoritmo** EstimarGrupos(X, Θ_{set}, d, q_{max})

  para cada Θ **em** Θ_{set}, **faça**

    Execute BSAS(X, d, Θ, q_{max}) várias vezes,
      apresentando os exemplos X em uma ordem diferente a cada passo

    Estime q_Θ como o número de grupos mais frequente do BSAS para o limiar Θ

    Acrescente q_Θ a um conjunto q_{set}

  retorne q_{set}

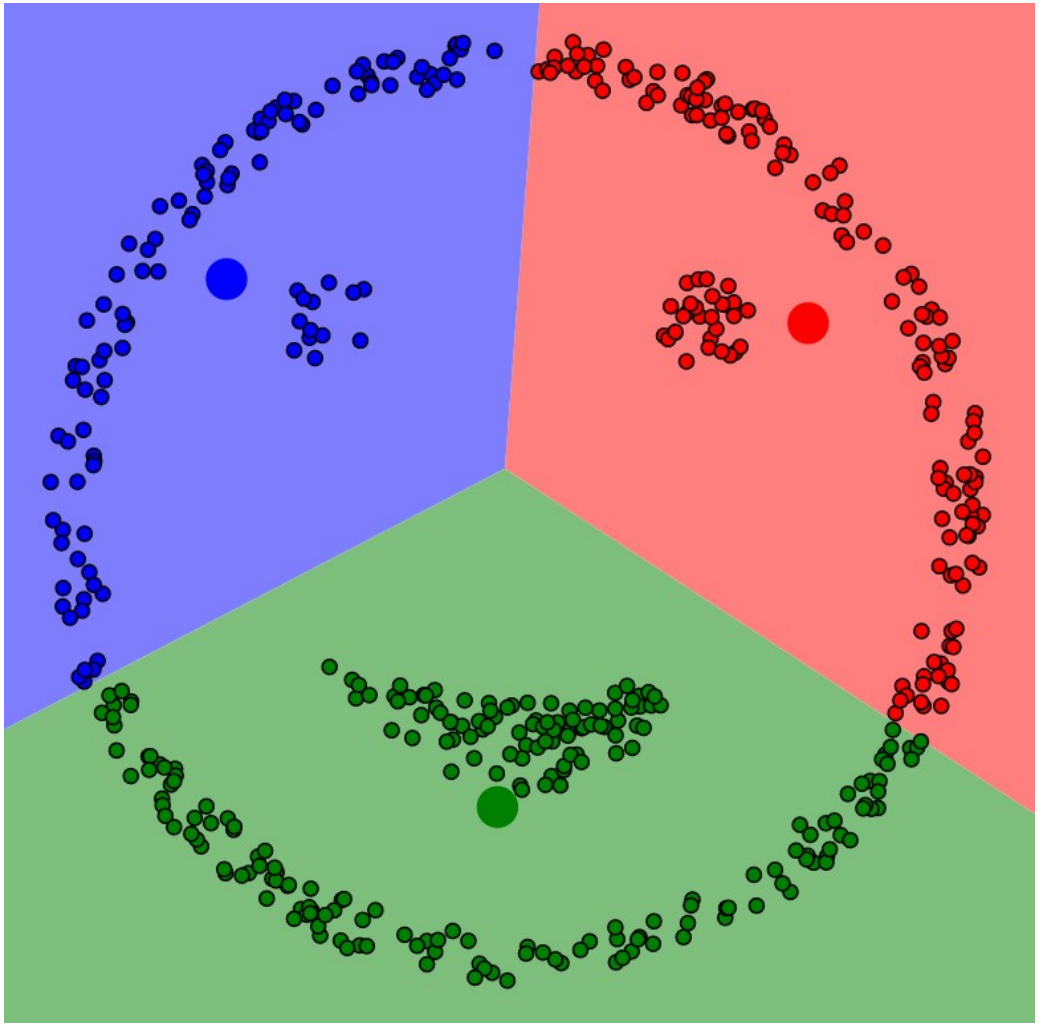
Agenda

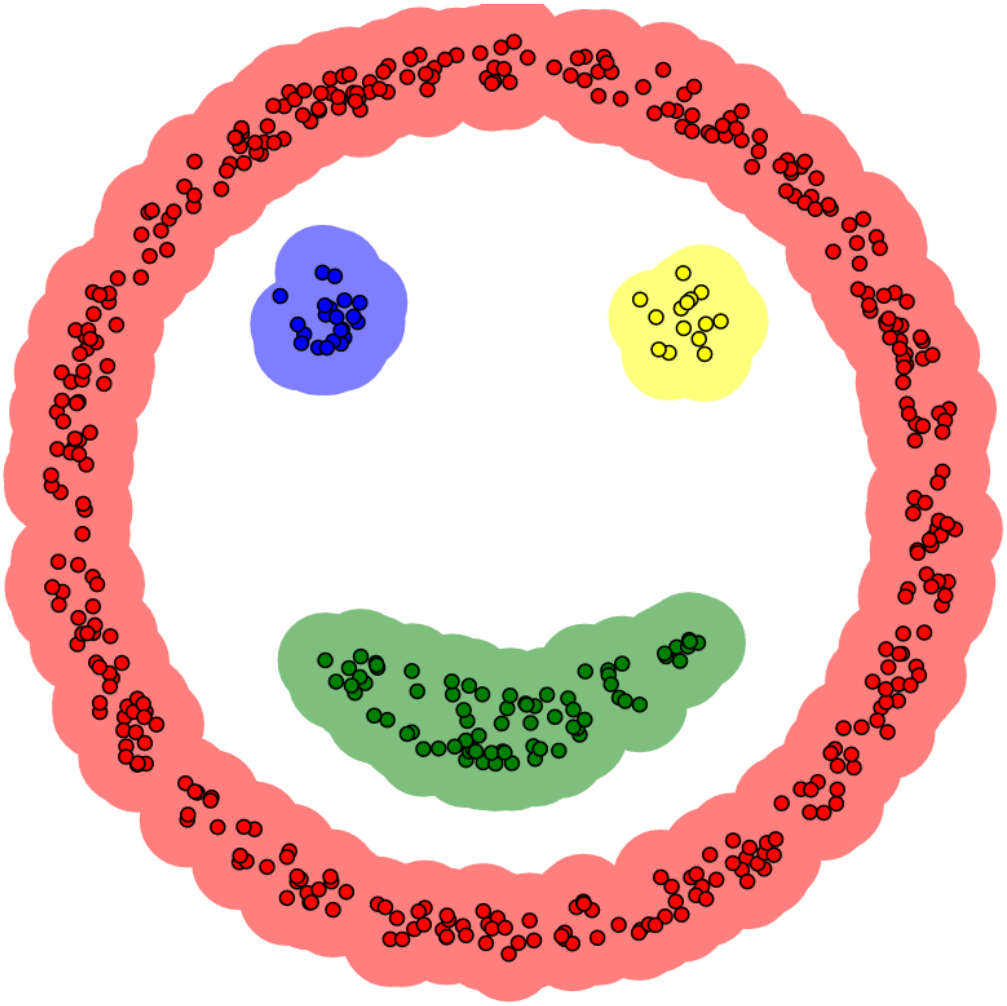
- Aprendizado não supervisionado
- Grupos e tipos de agrupamento
- Algoritmos de agrupamento
- O k-*means*
- Algoritmos sequenciais
- Agrupamento por densidade – DBSCAN
- Agrupamento hierárquico
- Qualidade de agrupamentos

Agrupamento por densidade

- O **DBSCAN** é um dos algoritmos de agrupamento mais utilizados na literatura
  - *Density-based spatial clustering of applications with noise*
- Trata-se de um algoritmo de agrupamento baseado na otimização de uma função de custo
  - Utiliza o conceito de vizinhos mais próximos para estimar regiões densas

Agrupamento por densidade





DBSCAN

- O DBSCAN classifica os exemplos como *core points*, pontos de fronteira ou *outliers*
  - Um exemplo é considerado um ***core point*** se está em uma região de raio ε que possui pelo menos *minPoints* pontos (incluindo ele mesmo)
  - Um exemplo é um **ponto de fronteira** se não é um *core point*, mas tem distância inferior a ε de algum *core point*
  - Os demais exemplos são ***outliers*** (ruído, *noise*)

ALGORITHM 1: Pseudocode of Original Sequential DBSCAN Algorithm

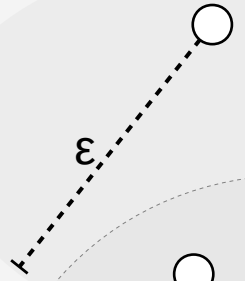
Input: *DB*: Database
Input: ε: Radius
Input: *minPts*: Density threshold
Input: *dist*: Distance function
Data: *label*: Point labels, initially *undefined*

```
1  foreach point p in database DB do                          // Iterate over every point
2      if label(p) ≠ undefined then continue                  // Skip processed points
3      Neighbors N ← RangeQuery(DB, dist, p, ε)               // Find initial neighbors
4      if |N| < minPts then                                   // Non-core points are noise
5          label(p) ← Noise
6          continue
7      c ← next cluster label                                 // Start a new cluster
8      label(p) ← c
9      Seed set S ← N \ {p}                                   // Expand neighborhood
10     foreach q in S do
11         if label(q) = Noise then label(q) ← c
12         if label(q) ≠ undefined then continue
13         Neighbors N ← RangeQuery(DB, dist, q, ε)
14         label(q) ← c
15         if |N| < minPts then continue                      // Core-point check
16         S ← S ∪ N
```

E.SCHUBERT, J.SANDER, M.ESTER, H-P. KRIEGEL e X.XU. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. (2017)

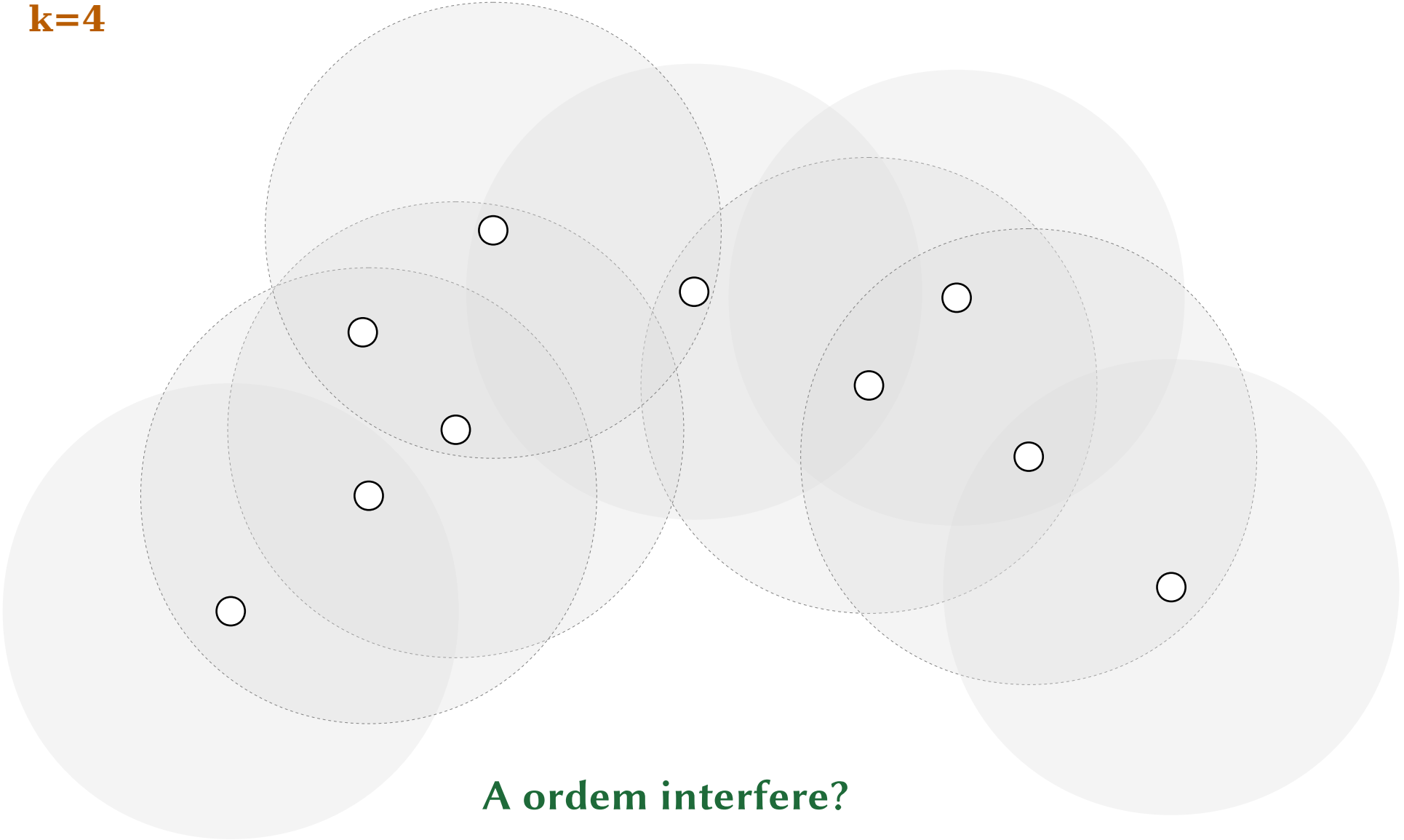
DBSCAN

k=4



DBSCAN

k=4



A ordem interfere?

Agenda

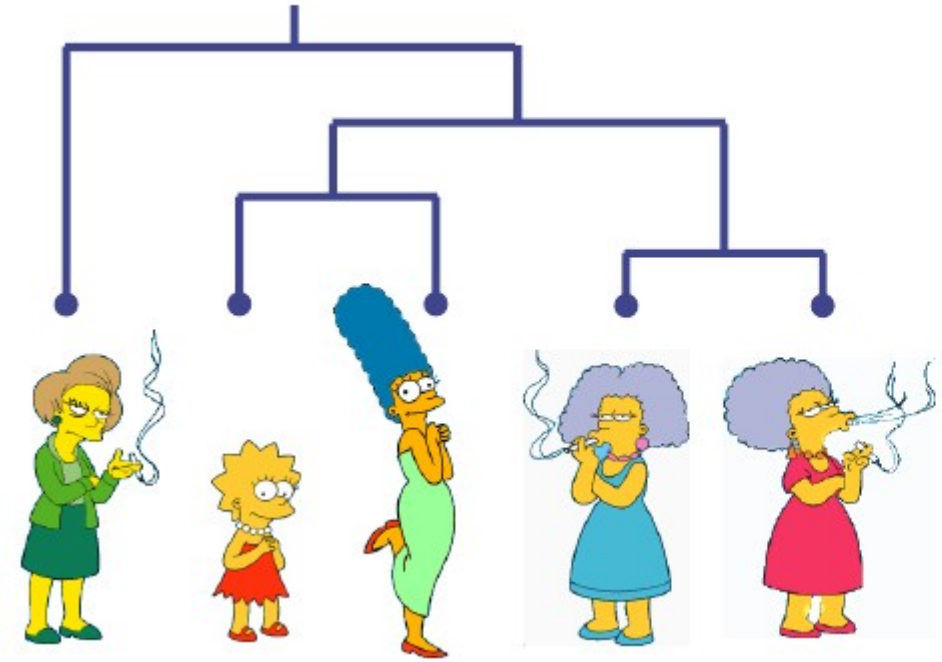
- Aprendizado não supervisionado
- Grupos e tipos de agrupamento
- Algoritmos de agrupamento
- O k-*means*
- Algoritmos sequenciais
- Agrupamento hierárquico
- Qualidade de agrupamentos

Agrupamento hierárquico

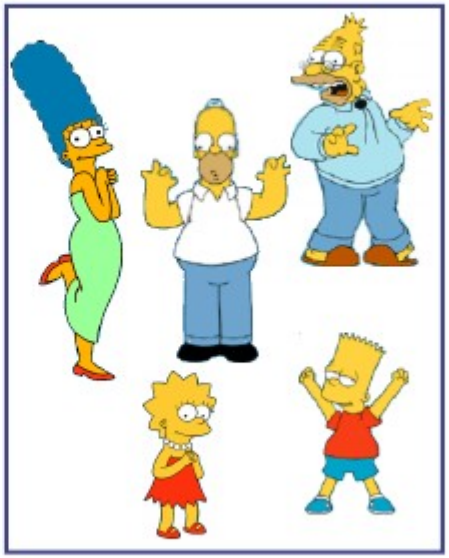
- Distinguem-se dos agrupamentos particionais por definirem um **hierarquia** entre os grupos
  - Define uma relação de ordem parcial entre os grupos
    - Alguns grupos estão **contidos** em outros
    - Alguns pares de grupos estão contidos em um **grupo comum**
  - Agrupamentos hierárquicos são, tipicamente, representados por meio de **dendogramas**

Dendograma vs. particionamento

Hierárquicos



Particionais



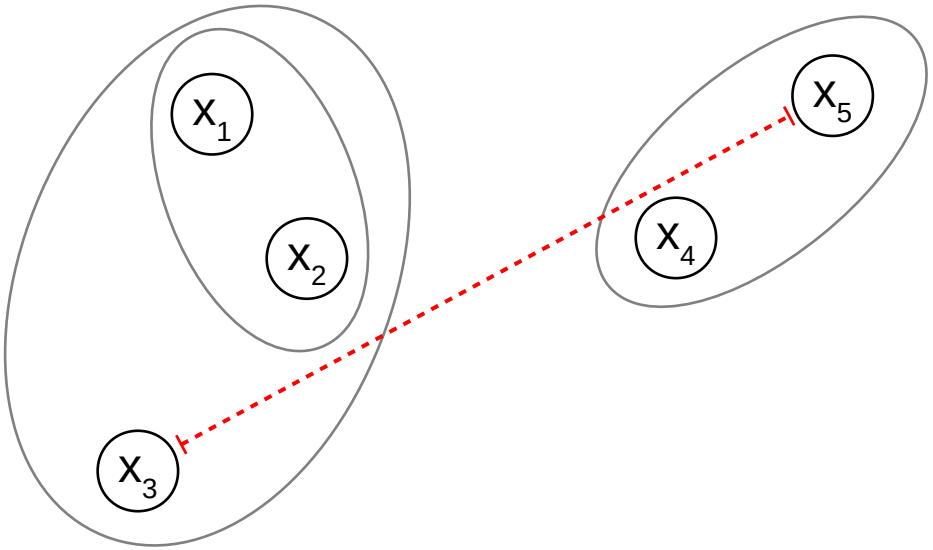
Distância entre grupos

- O agrupamento hierárquico requer uma **função de distância** entre grupos
  - Essa distância é representada como a **altura da junção** dos grupos no dendrograma

Distância entre grupos

- Distância **máxima** ou ***complete-linking***
  - A distância entre dois grupos é a distância entre os exemplos mais distantes

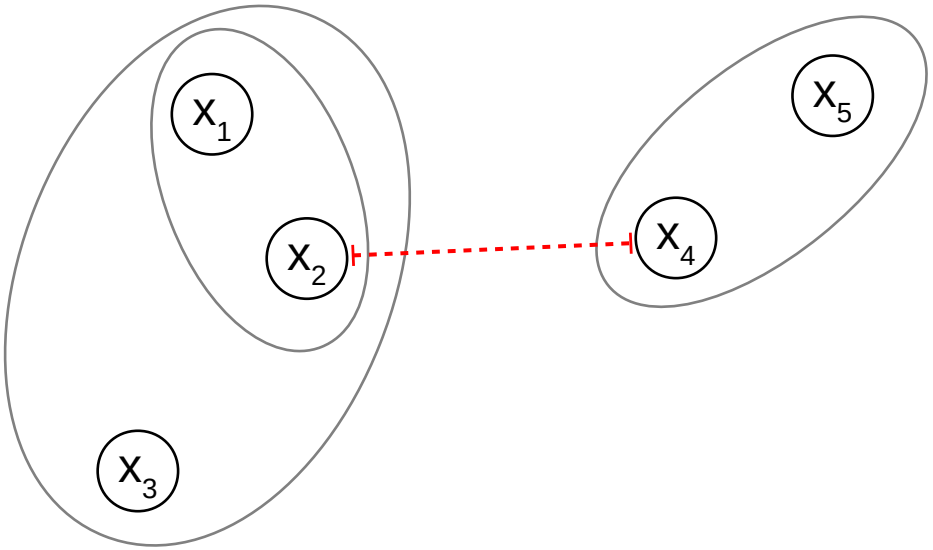
$$d_{\max}(C_i, C_j) = \max\{d(x, y) : x \in C_i, b \in C_j\}$$



Distância entre grupos

- Distância **mínima** ou ***single-linking***
  - A distância entre dois grupos é a distância entre os exemplos mais próximos

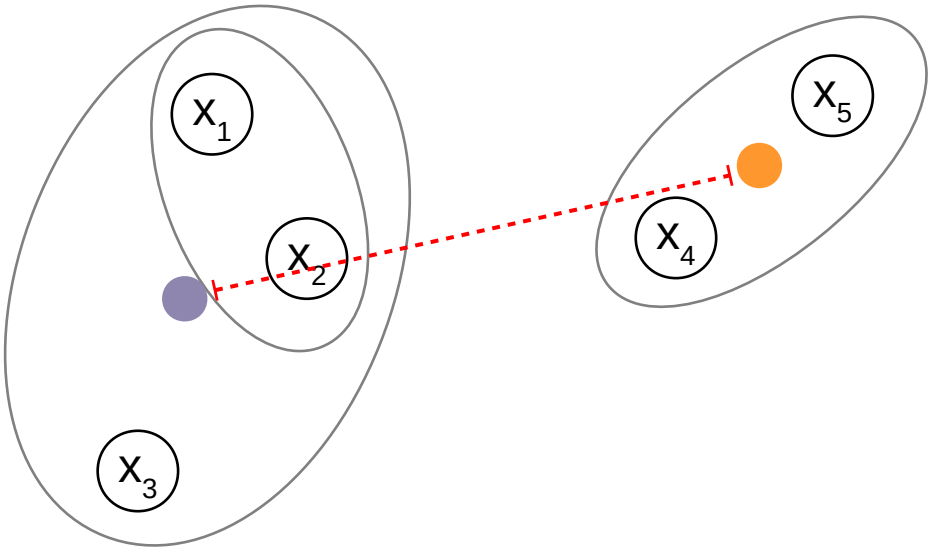
$$d_{\min}(C_i, C_j) = \min\{d(x, y) : x \in C_i, b \in C_j\}$$



Distância entre grupos

- Distância entre centroides
  - A distância entre dois grupos é a distância entre os centroides dos grupos

$$d_{\text{cent}}(C_i, C_j) = d(c_i, c_j)$$



Tipos de algoritmos

- **Algoritmos aglomerativos**
  - Cada exemplo começa em um grupo C_0
  - A cada iteração, dois grupos C_i e C_j são fundidos e ficam aninhados no grupo C_k
- **Algoritmos divisivos**
  - Todos os exemplos começam em um único grupo C_N
  - A cada iteração, um grupo C_k é dividido para formar os grupos aninhados C_i e C_j

Algoritmos aglomerativos

- Algoritmos gulosos podem ser obtidos por uma abordagem gulosa
- AGNES (*Agglomerative Nesting*)
  - Em cada passo do algoritmo, selecione o conjunto de grupos mais próximos
    - O número de grupos é $\mathcal{O}(N^2)$

Algoritmos aglomerativos

- **Algoritmo** AgrupamentoAglomerativo(X)

  d ← matriz de distâncias dos exemplos X

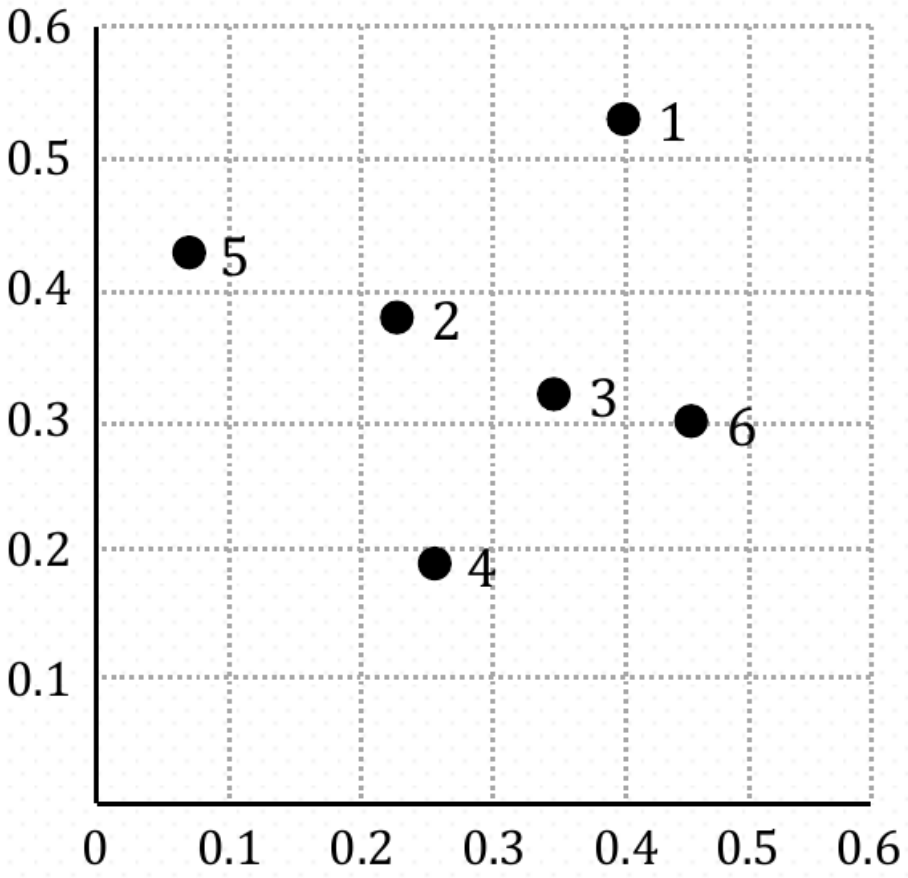
  C ← $\{C_i\}_{i=1..|X|}$ tal que $C_i = \{x_i\}$

  enquanto |C| > 1

    Selecione e mescle os dois grupos mais próximos de C

    Atualize d, se necessário

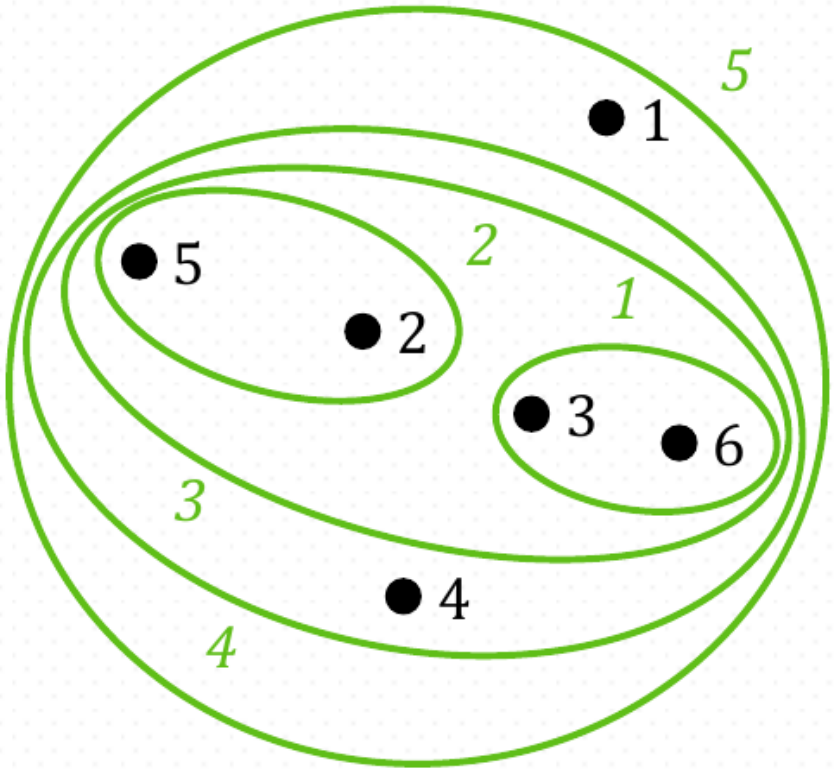
Algoritmos aglomerativos



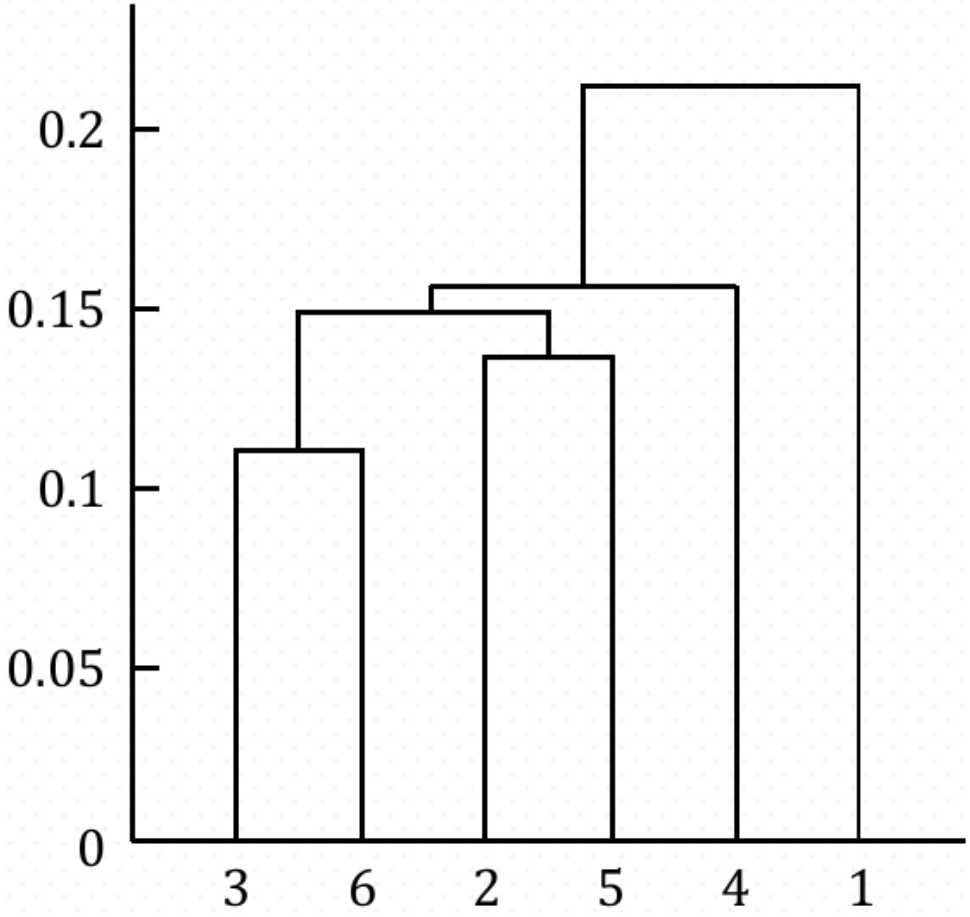
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Algoritmos aglomerativos

Nested Cluster Diagram



Hierarchical Tree Diagram



Algoritmos divisivos

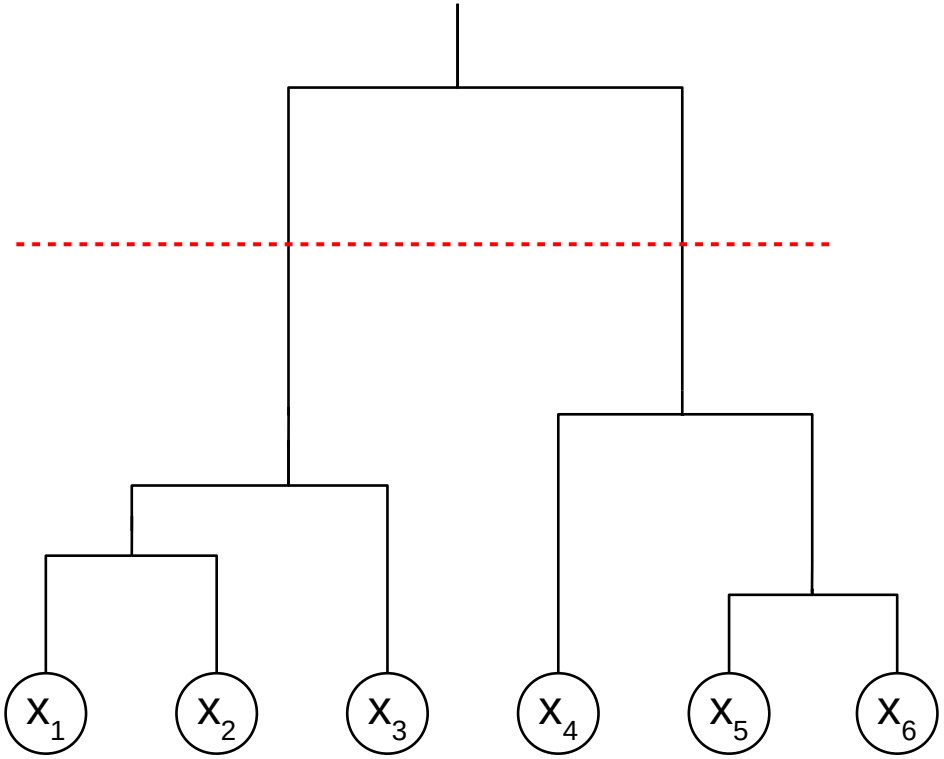
- Algoritmos de agrupamento divisivo não podem adotar uma solução gulosa
  - O número de formas de dividir um grupo que contém N sub-grupos é $\mathcal{O}(2^N)$
- DIANA (*Divisive Analysis*)
  - Em cada iteração, seleciona o grupo mais heterogêneo
  - Aplica heurística para dividir esse grupo
    - Exemplo: k-*means* com k=2 grupos

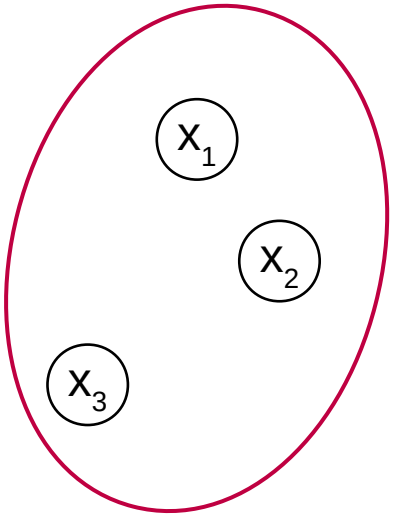
AGNES vs. DIANA

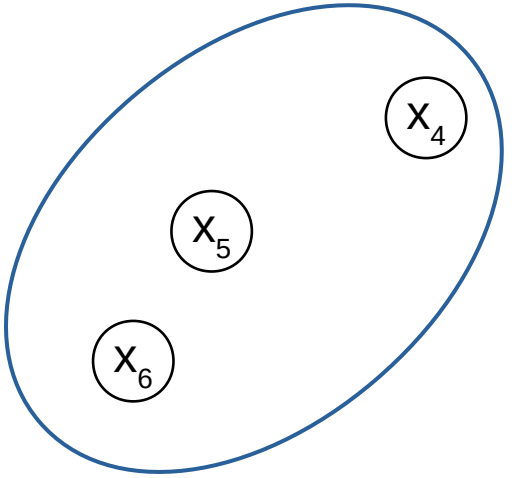
- Alguns autores defendem que algoritmos divisivos podem produz agrupamentos mais eficazes
  - Algoritmos aglomerativos são gulosos
    - Fazem decisões locais
    - Um aninhamento não é desfeito
  - Algoritmos divisivos têm informação global

Particionamento de agrupamentos hierárquicos

- Um agrupamento hierárquico pode ser convertido em um agrupamento particional







Particionamento de agrupamentos hierárquicos

- Podemos fazer esse corte através do tempo de vida
  - O **tempo de vida** de um grupo é a sua distância para o grupo que o "absorveu"
    - A diferença de altura do ponto em que o grupo é criado até o ponto em que ele é "absorvido"
  - Podemos escolher um ponto de corte que aumente o tempo de vida dos grupos

Agenda

- Aprendizado não supervisionado
- Grupos e tipos de agrupamento
- Algoritmos de agrupamento
- O k-*means*
- Algoritmos sequenciais
- Agrupamento hierárquico
- Qualidade de agrupamentos

Imposição de estrutura

- Assim como o aprendizado supervisionado, também existe viés no agrupamento
- O agrupamento impõe uma estrutura sobre os dados
- Essa estrutura pode não existir
  - O k-*means* supõe que os grupos estão distribuídos em torno de centroids
  - O DBSCAN supõe que os grupos são regiões de alta densidade separados por regiões de baixa densidade
  - etc.

Imposição de estrutura

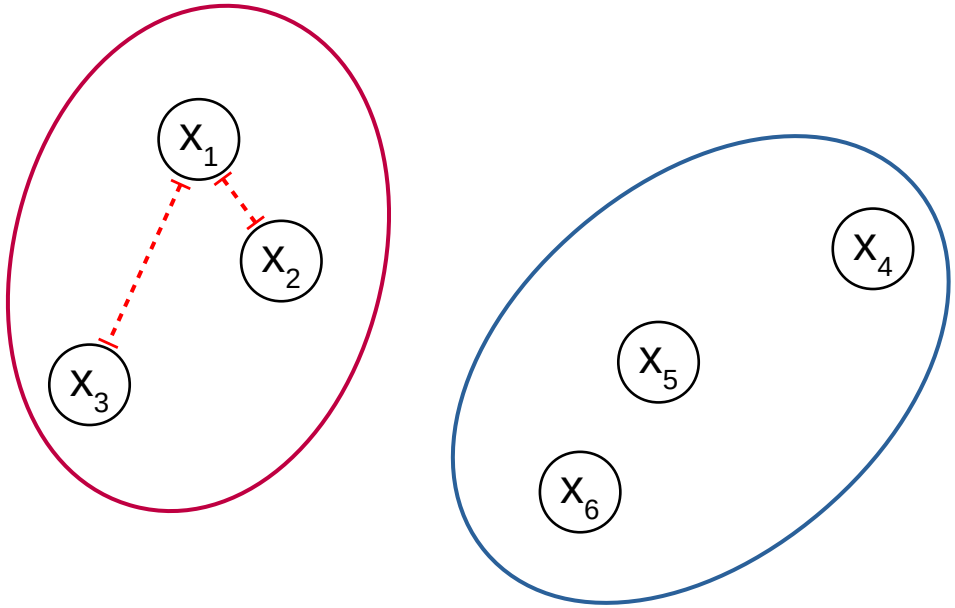
- Alguns algoritmos supõem um número de grupos
- Esse número de grupos pode não condizer com a verdadeira estrutura dos dados
  - Uma possível abordagem é repetir o mesmo algoritmo, variando o número esperado de grupos

Avaliação de agrupamentos

- Como os dados não possuem rótulos, precisamos de métricas alternativas para avaliar os grupos
  - Supomos que a propriedade desejada do agrupamento é máxima similaridade intra-grupo e mínima similaridade inter-grupo
  - Utilizamos índices, tais como silhueta

Silhueta

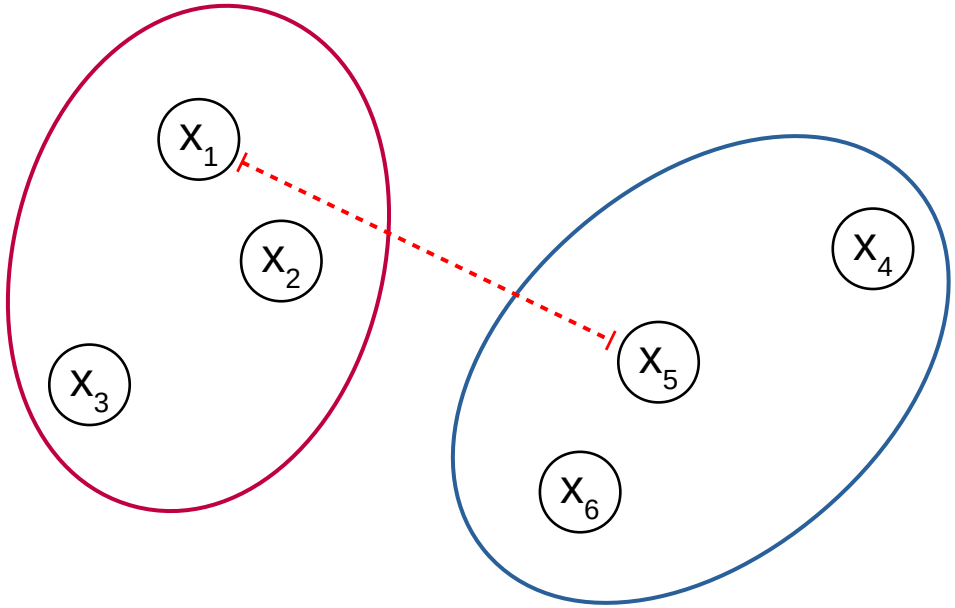
- Dado um agrupamento particional $C = \{C_1, C_2, ..., C_k\}$
- Para cada exemplo $x_i \in C_i$, calcule a **distância intra-grupo média** $\mathbf{a(x_i)}$ e a distância mínima inter-grupo $b(x_i)$



$$a(x_i) = \frac{1}{|C_i| - 1} \sum_{x_j \in C_i} d(x_i, x_j)$$

Silhueta

- Dado um agrupamento particional $C = \{C_1, C_2, ..., C_k\}$
- Para cada exemplo $x_i \in C_i$, calcule a distância intra-grupo média $a(x_i)$ e a **distância mínima inter-grupo** $\mathbf{b(x_i)}$



$$a(x_i) = \frac{1}{|C_i| - 1} \sum_{x_j \in C_i} d(x_i, x_j)$$

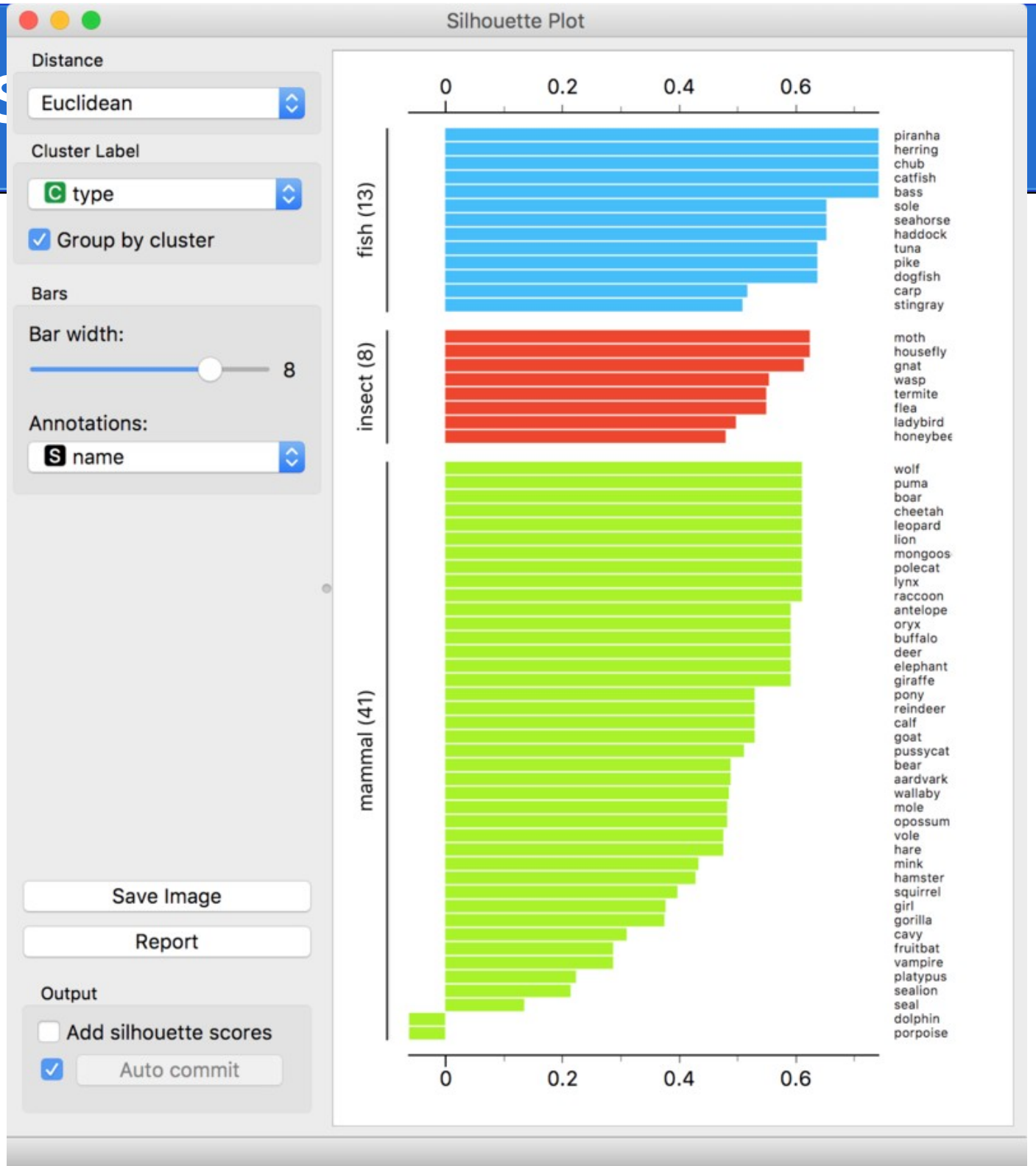
$$b(x_i) = \min_{j \neq i} \left(\frac{1}{|C_j|} \sum_{x_j \in C_j} d(x_i, x_j) \right)$$

Silhueta

- Para cada ponto, sua silhueta será a diferença entre as distâncias intra e inter-grupo, em razão da maior delas
  - Se $|C_i| > 1$, então

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max\{a(x_i), b(x_i)\}}$$

  - Se $|C_i| = 1$, então $s(x_i) = 0$
- A média e a mediana da silhueta dos grupos nos dá informações sobre o quão bem divididos os exemplos estão com respeito aos outros grupos



https://en.wikipedia.org/wiki/Silhouette_(clustering)#/media/File:Silhouette-plot-orange.png