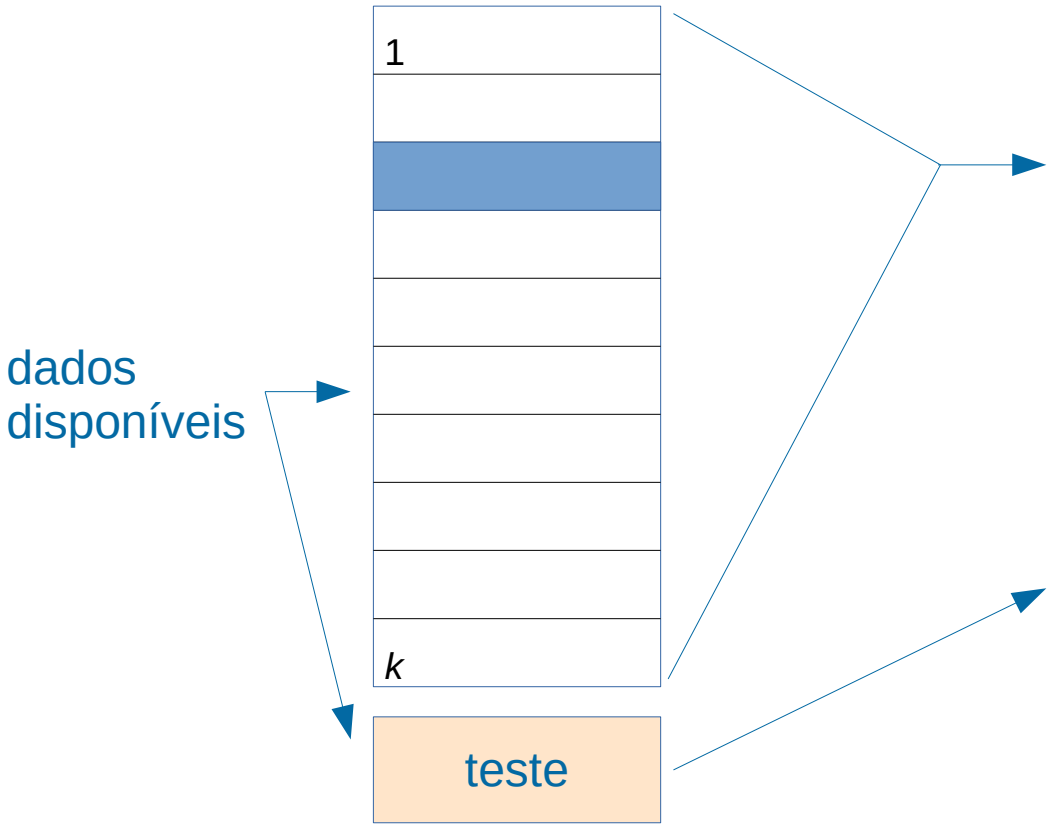
aprendizagem automática baseada em dados – vizinhos mais próximos; aglomeração (*clustering*)

Ciências ULisboa

treino, validação e teste
(consolidação)



treino e validação

k-fold cross validation

treina com *k*-1 e valida com 1

k vezes
(cada 1 dos *k* como validação)

teste

o resultado do modelo (após o treino) é dado com este conjunto

Ciências ULisboa

um modelo não paramétrico

não usa um nº fixo de parâmetros para representar o modelo

⇒ cada avaliação necessita recorrer ao conjunto de treino

+ não tem fase de treino!

k-vizinhos mais próximos

k-nearest neighbours (*NN*)

classificador (regressor) simples

classificação de um novo caso $\mathbf{x}_q \rightarrow NN(k, \mathbf{x}_q)$

com base nos *k* exemplos mais próximos de $\mathbf{x}_q$

em classificação binária

por votação – classe do maior nº dos *k* vizinhos mais próximos

⇒ *k* é ímpar! (para evitar empates)

distâncias

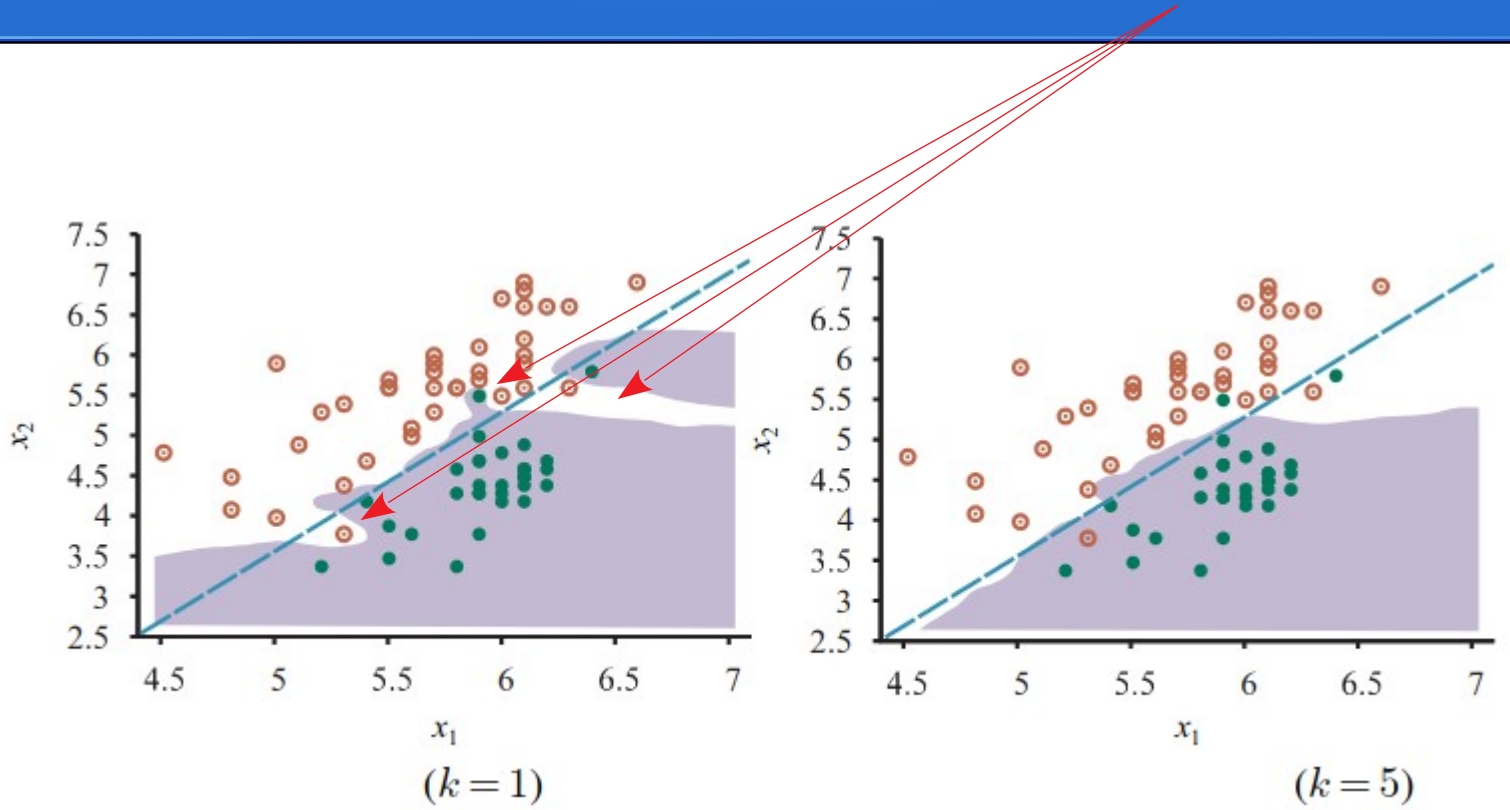
"proximidade" ⇒ métrica de **distância**

Euclideana: $D(\mathbf{x}_j, \mathbf{x}_q) = \sqrt{\sum_i (x_{ji} - x_{qi})^2}$ atributos nas mesmas unidades

Manhattan: $D(\mathbf{x}_j, \mathbf{x}_q) = \sum_i |x_{ji} - x_{qi}|$ atributos em unidades diversas

Hamming: *nº de atributos diferentes* atributos booleanos

vizinhos mais próximos - overfit



o *NN* não precisa fazer isto!

é só para visualizarmos como classifica qualquer ponto no plano

normalização

medidas de distância como Euclideana e Manhattan dão mais importância a atributos de valores elevados

para evitar isso → **normalização**

para cada atributo x_i calcula-se a média μ_i e o desvio padrão σ_i

o atributo normalizado é $x'_i = \frac{x_i - \mu_i}{\sigma_i}$

implementação do *k*-NN

complexidade de procurar os *k* vizinhos mais próximos em *N* exemplos

estrutura de dados linear (lista,…): $O(N)$

árvore binária: $O(\log N)$

tabela de dispersão (*hash table*): $O(1)$

maldição da dimensionalidade

num espaço *n* dimensional

se *N* exemplos cabem num hiper-cubo de volume 1

e *k* exemplos ocupam um hiper-cubo de lado *l* ⇒ volume l^n

⇒ em média $l^n = k/N \Rightarrow l = (k/N)^{1/n}$

supondo $k = 10$ e $N = 1.000.000$

- ➔ com $n = 2 \Rightarrow l = 0{,}003$ (0,3% do lado do espaço de *N*, quadrado de lado 1)
- ➔ com $n = 3 \Rightarrow l = 2\%$
- ➔ com $n = 17 \Rightarrow$ ***l* = 94%**

as distâncias entre quaisquer dois pontos são semelhantes (próximas da média)

aprendizagem não supervisionada

os exemplos apenas têm caraterísticas (atributos)

não têm resultado correto (classe, ou valor)

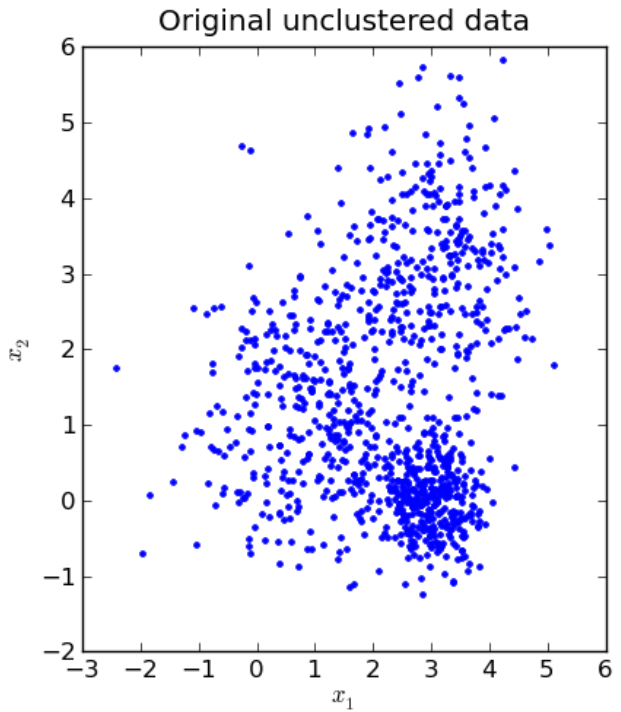
---

em algoritmos de aglomeração / agrupamento (*clustering*):

o conjunto de treino é usado para identificar grupos (aglomerados) com caraterísticas similares

alguns destes algoritmos definem um protótipo de cada *cluster*

algoritmos de aglomeração (*clustering*)



fonte:https://i.stack.imgur.com/cIDB3.png

Ciências ULisboa

clusters

caraterísticas desejáveis

elementos de um *cluster* devem ser semelhantes

- **similaridade *intra-cluster* elevada**

elementos de *clusters* diferentes devem ser bem distintos

- **similaridade *inter-cluster* baixa**

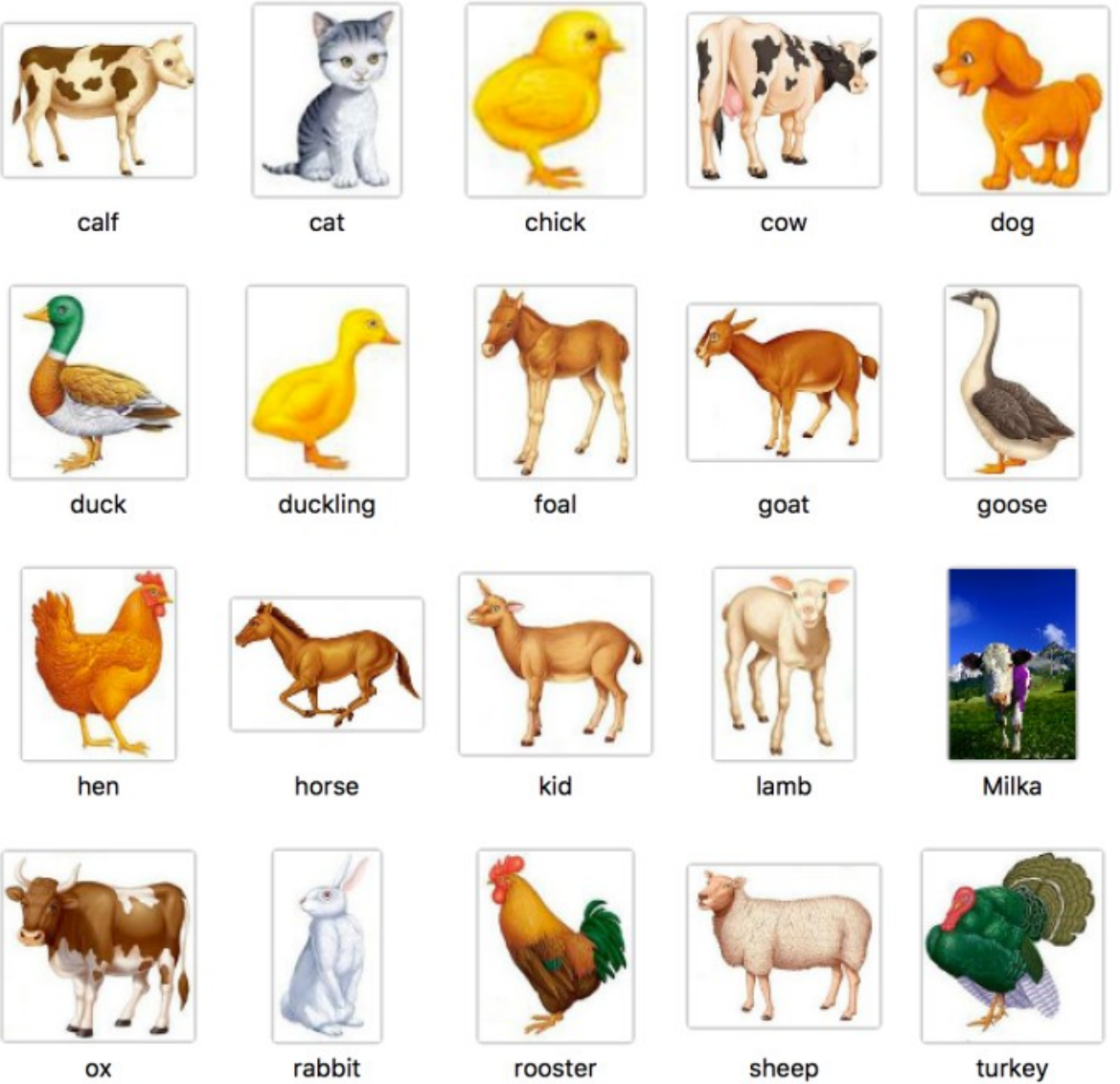
---

os grupos são disjuntos

- cada exemplo só pertence a um grupo

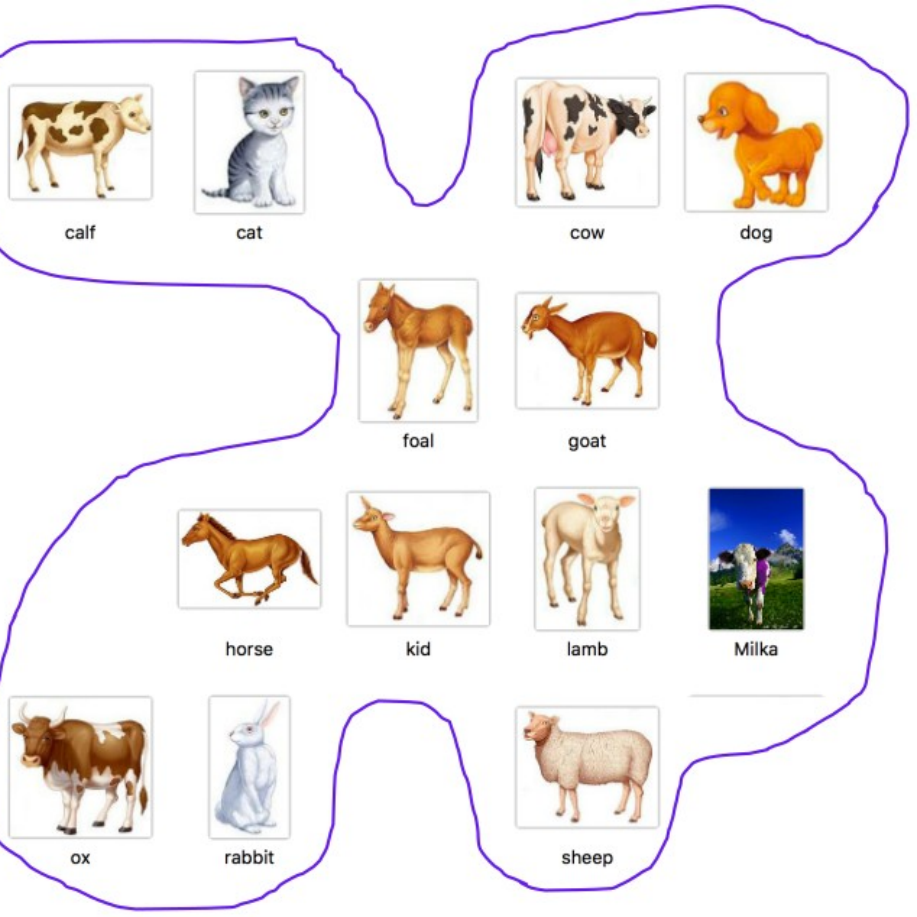
animais domésticos

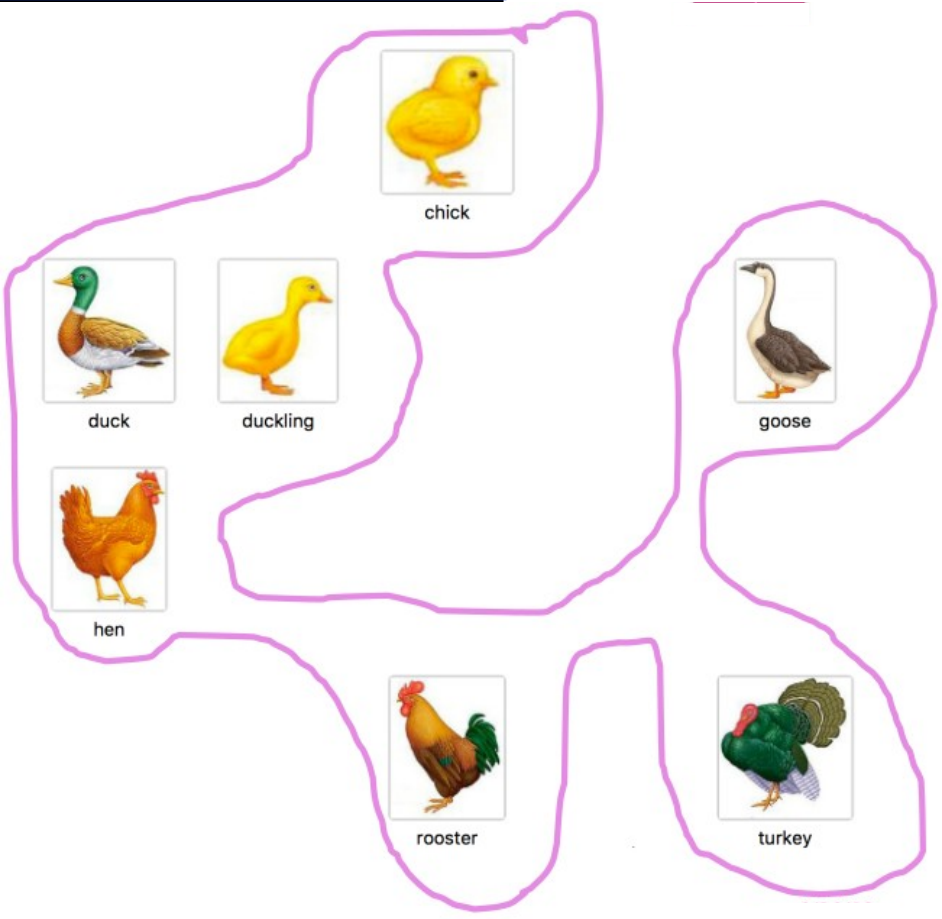
quantos grupos?



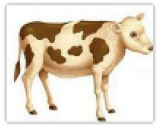
fonte:
https://blog.biolab.si/2017/04/03/image-analytics-clustering/

2 grupos


calf
cat
cow
dog
foal
goat
horse
kid
lamb
Milka
ox
rabbit
sheep


chick
duck
duckling
goose
hen
rooster
turkey

3 grupos


calf


cow


foal


goat


horse


kid


lamb


Milka


ox


rabbit


sheep


cat


dog


chick


duck


duckling

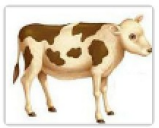

hen


goose


rooster


turkey

3 grupos – alternativa


calf


cow


foal


goat

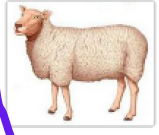

horse


kid


lamb


Milka


ox


sheep


cat


dog


rabbit


chick


duck


duckling


goose


hen


rooster


turkey

aspetos a ter em conta

nº de grupos dado *a priori*, ou automaticamente determinado

formar grupos é uma forma de classificação!

como se determina a semelhança entre exemplos

como se avaliam os resultados

como se interpretam os resultados

k-means (*k*-médias)

um modelo baseado em centroides

- centroide $\boldsymbol{C}_j$ = protótipo de cada grupo
- representa o grupo
- é o **valor médio** (*means*) dos exemplos do grupo

distância Euclideana para medir a dissemelhança entre dois pontos

- em particular entre um exemplo $\mathbf{x}_i$ e o centróide $\boldsymbol{C}_j$

k-means – passos

1. seleciona *k* exemplos do conjunto de treino *D*

   cada um deles figura como um centroide

2. para cada um dos exemplos de *D*, afeta-o ao grupo do centroide mais próximo (menor distância Euclideana)
3. para cada grupo calcula a média dos exemplos afetados ao grupo, que passa a ser o respetivo centroide, $\boldsymbol{C}_i$
4. repete desde 2 até uma iteração em que não há alterações nos grupos

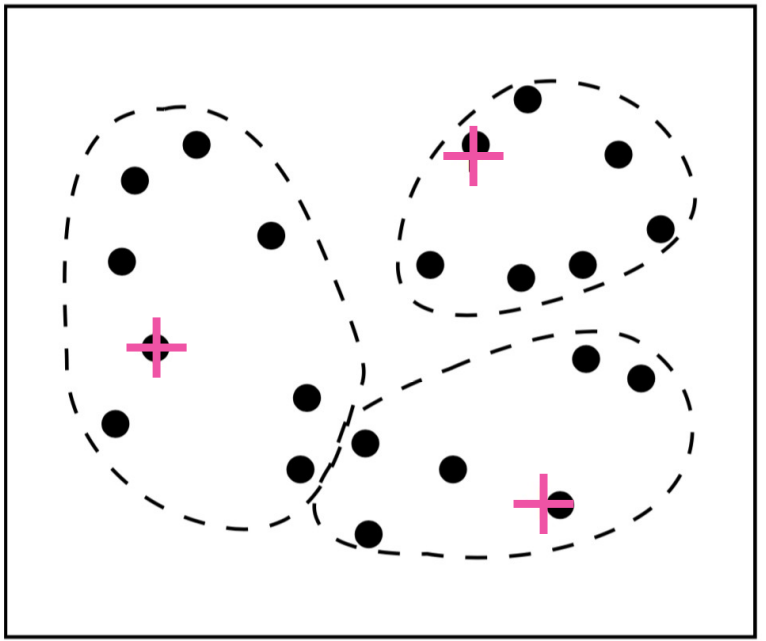
exemplo – a)

problema c/ 2 dimensões

2 atributos numéricos

+ centroides iniciais – pontos do conj. treino (aleatoria/)

exemplos atribuídos ao grupo do centroide mais próximo



exemplo – b)

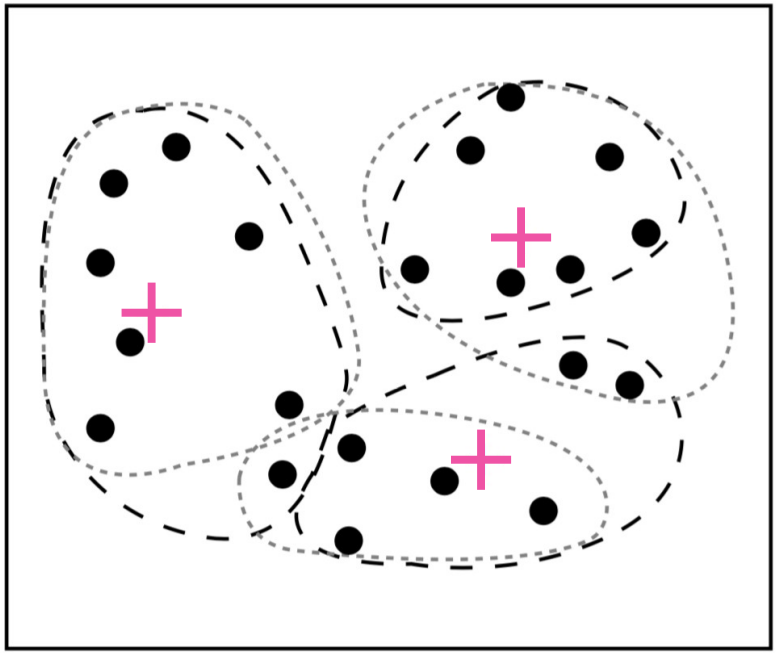
atualizam-se os centroides

média dos pontos do grupo

e reagrupam-se os pontos

atribuídos aos grupos dos centroides mais próximos

(tracejado fino)



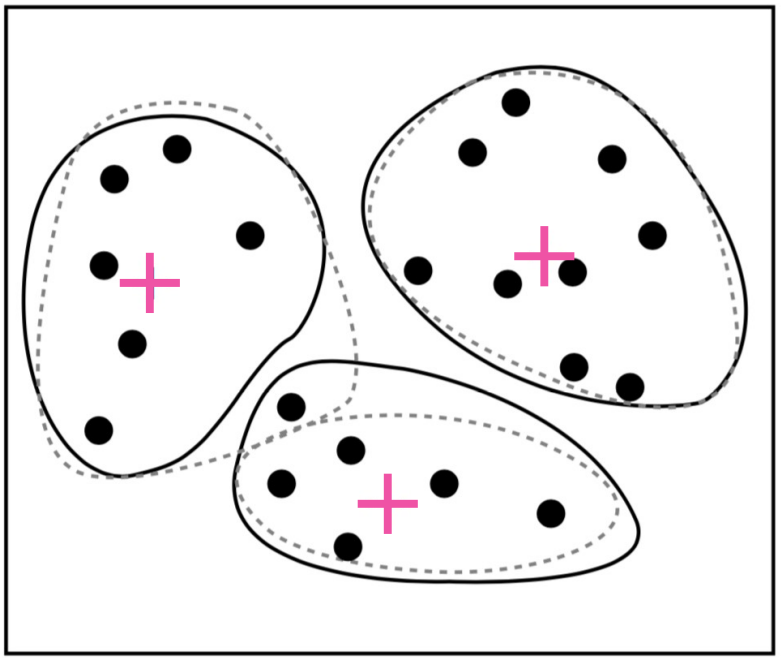
exemplo – c)

após algumas iterações

deixa de haver alterações

retorna os grupos e centroides definidos nessa altura

(linha contínua e cruzes)



melhoramentos

k-means é sensível aos centroides iniciais

- solução simples: correr várias vezes e retornar a melhor

preferível: *k-means*++

- centroides iniciais aleatórios mas com probabilidade proporcional ao quadrado da distância aos centroides já definidos
- (é a initialização por defeito no Scikit-learn)

que valor de *k* ?

muito pequeno – pode manter no mesmo grupo exemplos pouco semelhantes

muito grande – no limite, cada exemplo é o seu centroide *overfit*

método de escolha automática testa vários valores de *k* e escolhe aquele em que avaliação é melhor

- atualmente o mais usado é o Silhouette – combina medida de coesão intra-cluster com separação inter-cluster