

APRENDIZAGEM DE MÁQUINA

(usando Python)

Thiago Marzagão

CLUSTERIZAÇÃO

regressão/classificação vs clusterização

- Regressão/classificação: temos x_1, x_2, \dots, x_p e queremos prever y .
- Clusterização: temos x_1, x_2, \dots, x_p e queremos dividir as amostras em grupos (clusters).

regressão/classificação vs clusterização

- Regressão/classificação: aprendizagem supervisionada.
- Clusterização: aprendizagem não-supervisionada.

clusterização

- Clusterização: temos x_1, x_2, \dots, x_p e queremos dividir as amostras em clusters.
- Exemplo: agrupar usuários do Netflix conforme os gêneros preferidos (romance, comédia, terror, ficção científica, etc).
- Seja:
- x_1 = quantos filmes/seriados de romance o usuário já assistiu no Netflix
- x_2 = quantos filmes/seriados de comédia o usuário já assistiu no Netflix
- x_3 = quantos filmes/seriados de terror o usuário já assistiu no Netflix
- x_4 = quantos filmes/seriados de ficção científica o usuário já assistiu no Netflix
- Queremos dividir os usuários do Netflix em clusters, de acordo com os gêneros preferidos.

k-means

- Vários algoritmos de clusterização são possíveis. O mais conhecido é o k-means.
- Objetivo: encontrar a divisão das amostras que minimiza a distância euclidiana quadrada média entre os pares de um mesmo cluster.

- Matematicamente:

- $$\operatorname{argmin}_{C_1, \dots, C_k} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}$$

- K é o número de clusters (você escolhe K)
- $|C_k|$ é o número de amostras no cluster k
- $j..p$ são as variáveis
- i, i' é um par de amostras

- Resolver $\operatorname{argmin}_{C_1, \dots, C_k} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}$ é difícil: existem quase K^n de particionar n amostras em K clusters.
- (NP-difícil: pelo menos tão difícil quanto os problemas mais difíceis em tempo polinomial não-determinístico.)
- Em vez de resolver o problema exatamente nós usamos o algoritmo de Lloyd p/ encontrar uma solução aproximada:
- 1) Designe cada amostra, aleatoriamente, a um dos K clusters.
- 2) Compute o centróide de cada cluster.
- 3) Designe cada amostra ao cluster cujo centróide esteja mais próximo.
- 4) Repita 2 e 3 até que as amostras permaneçam nos mesmos clusters.

k-means

<http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html>

k-means

- K-means tende a convergir p/ um ótimo local em vez de global.
- Duas soluções (não-excludentes):
- 1) Executamos k-means várias vezes e escolhemos o resultado que minimiza a soma das distâncias euclidianas quadradas médias.
- 2) k-means++:
 - a) escolhemos uma amostra aleatoriamente, c/ igual probabilidade p/ cada amostra ($p_i = 1/n$); chamemos essa amostra de c_1 ; ela será nosso primeiro centróide
 - b) calculamos a distância euclidiana quadrada entre cada amostra e c_1 :
$$D_i^2 = (x_i - c_1)^2$$
 - c) escolhemos uma amostra aleatoriamente, c/ probabilidade
$$p_i = D_i^2 / \sum_{i=1}^n D_i^2$$
; chamemos essa amostra de c_2 ; ela será nosso segundo centróide
 - d) recalculamos D_i^2 : $D_i^2 = \operatorname{argmin} \{ (x_i - c_1)^2, (x_i - c_2)^2 \}$
 - e) repetimos b) e c) até obtermos k centróides iniciais
 - f) executamos k-means

k-means

- Como escolher K ?
- Às vezes o K é óbvio dada a natureza do problema.
- Por exemplo, se sabemos de antemão que há 10 autores e queremos clusterizar 1000 obras por autor.
- Outras vezes não: existe um cluster "comédia" e outro "romance"? Ou apenas um cluster "comédia romântica"?
- E como saber se o K escolhido é o "correto"?
- Não há uma resposta simples. Só mesmo inspecionando, "no olho", a consistência interna de cada cluster. I.e., o cluster k "faz sentido"?
- Às vezes o problema é a seleção de variáveis. A inclusão de variáveis irrelevantes pode levar a uma clusterização subótima.
- Em suma: quando se trata de aprendizado *não-supervisionado* a vida é mais difícil; não dá p/ usar validação cruzada.