



Introdução ao aprendizado de máquina

Aula 8:

- Aprendizado não supervisionado
- K-médias
- Mixturas Gaussianas
- Sistemas de recomendação
- Recomendações baseadas no conteúdo
- Filtro Colaborativo

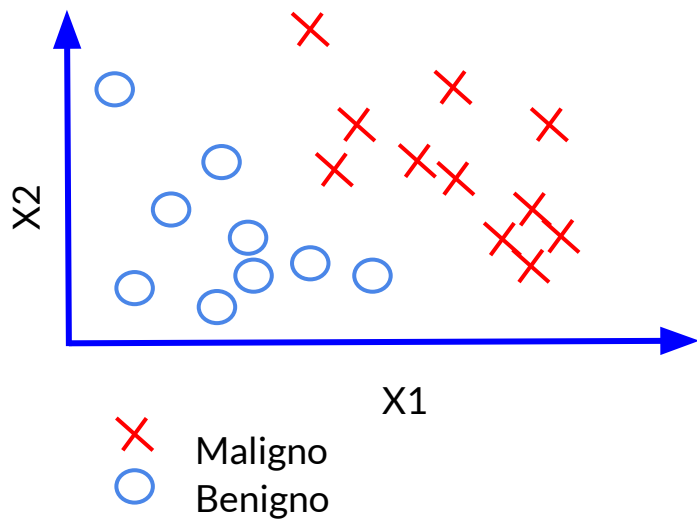


Aprendizado não supervisionado

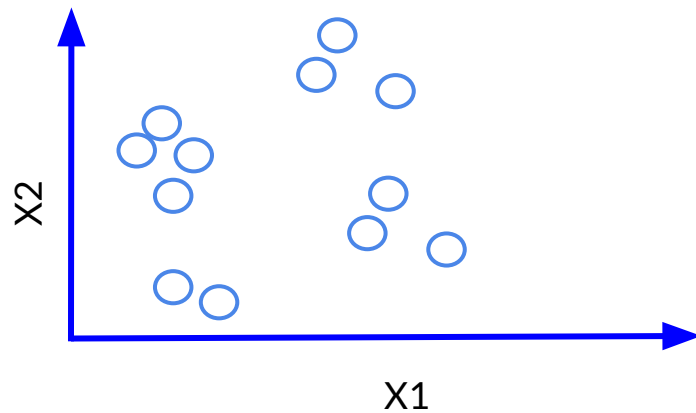
Aprendizado não supervisionado



Aprendizado Supervisionado



Aprendizado Não Supervisionado



Não há rótulos

Perguntas de aprendizado não supervisionado



- Pergunta comum: “ache algum padrão nos dados”
- Esses algoritmos são chamados de agrupamento (clustering)
- Aplicações::
 - Segmentação do mercado
 - Organizar grupos de servidores
 - Agrupar galáxias
 - Analizar redes sociais



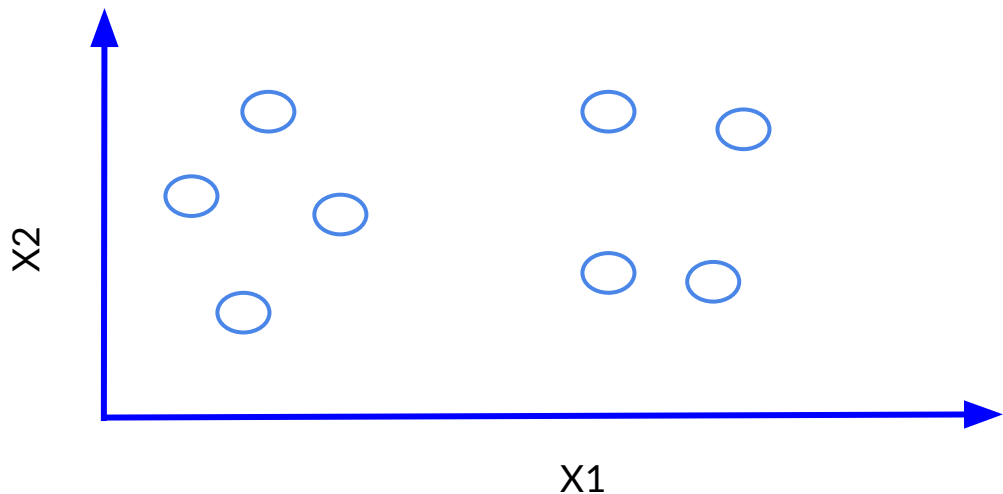
k-médias

Algoritmo de agrupamento de K-médias

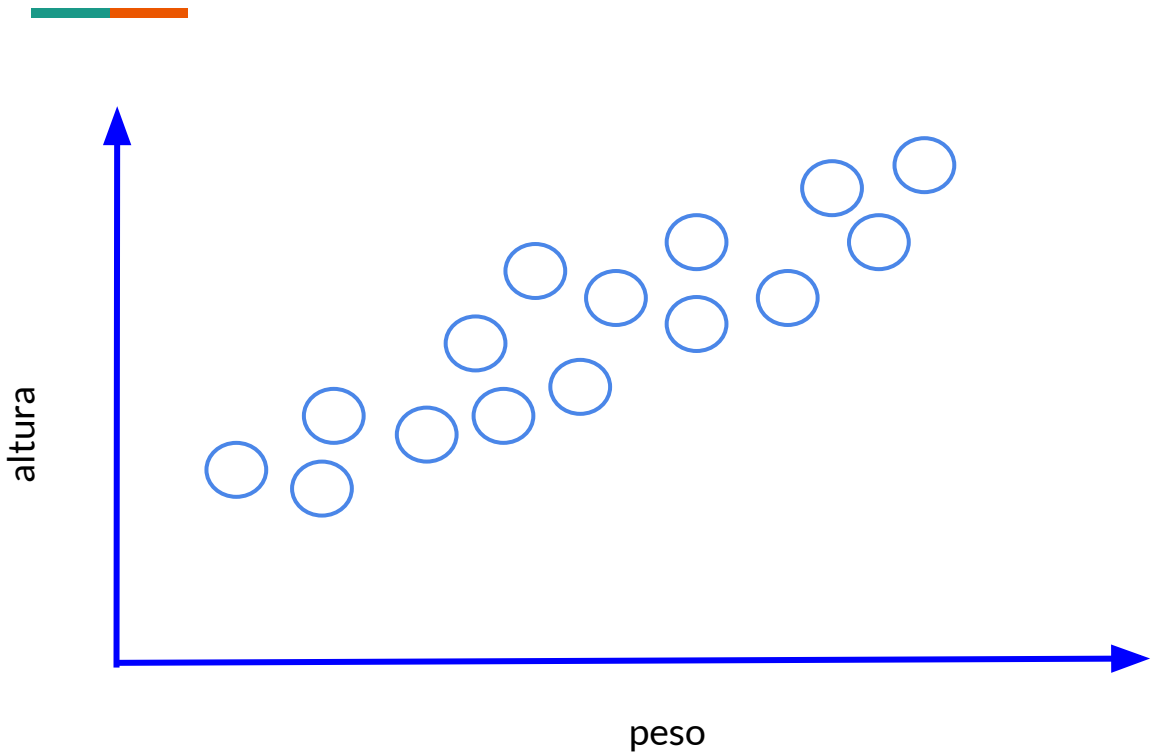


Há 2 passos em cada iteração do algoritmo:

- 1 - Atribuir cada observação a um grupo
- 2- Atualizar o centro do grupo



Exemplo de grupos não separados





Função Custo e Algoritmo

Função custo: Soma dos quadrados dos grupos



$$J(X, \mu) = \frac{1}{N} \sum_{i=1}^N \|x_i - \mu_{c_i}\|^2$$

c_i = grupo ao qual a obs i pertence

μ_{c_i} = centro do grupo que i pertence

Pseudo código de k-médias



KMedia(dados, k):

Continuar Iterando = True

Aleatoriamente selecionar k centroides iniciais

Enquanto Continuar Iterando:

- Para cada ponto, calcular a distância até cada centróide

- Atribuir cada ponto ao grupo com centróide mais próximo

- Calcular a média (em cada dimensão) dos pontos de cada grupo

- Atualizar o centróide do grupo para o valor médio de seus elementos

- Se novos centróides são iguais aos centróides anteriores

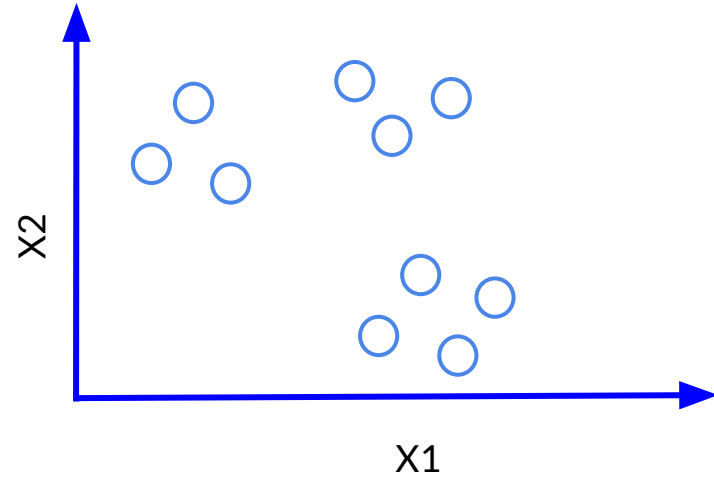
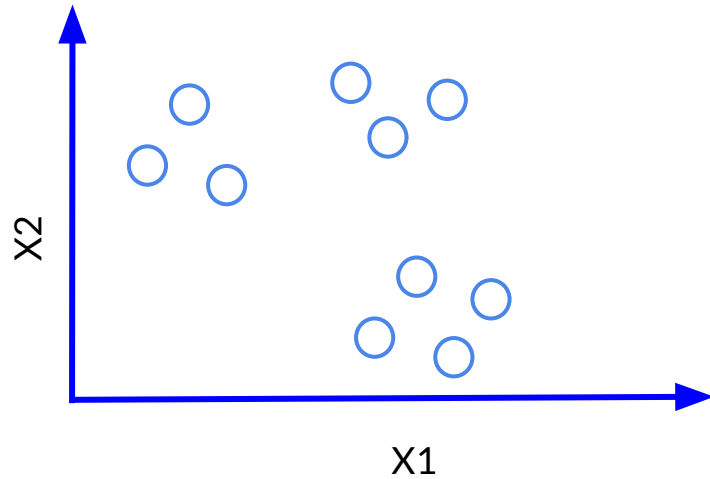
 - Continuar Iterando = False

Retornar valor dos centroides



Detalhes de Implementação

Ótimo Global versus ótimo local



Implementação de K-Médias

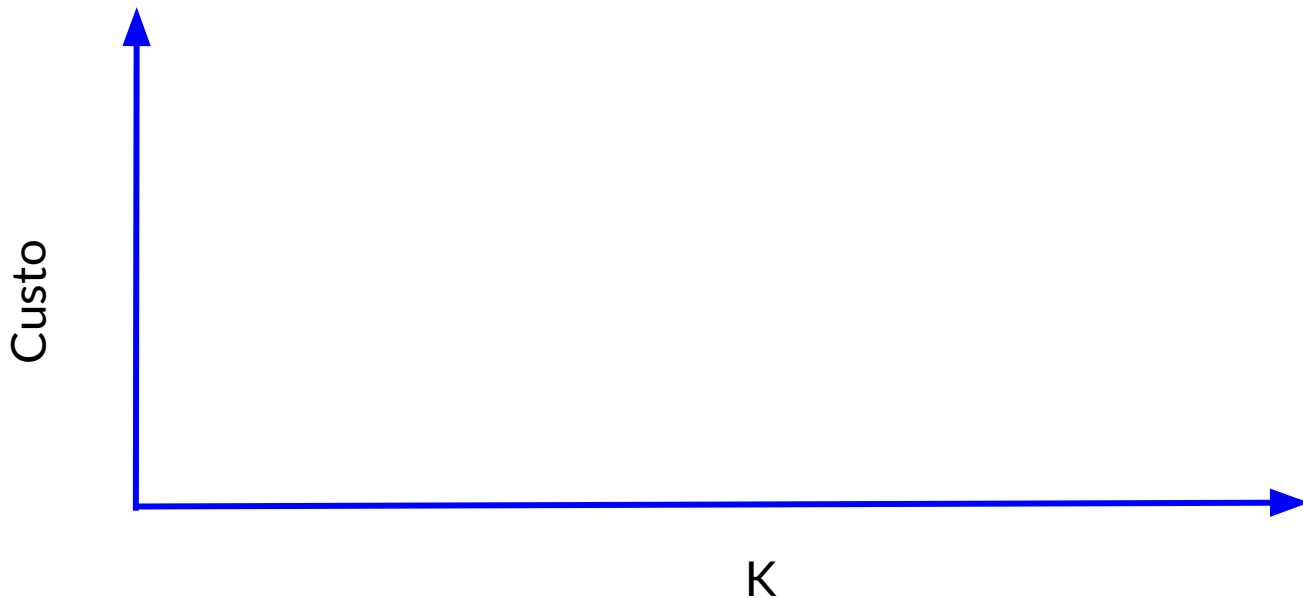


- Depois de cada iteração o custo nunca aumentará
- O algoritmo sempre converge, mas não necessariamente para para a mesma solução
- Podemos melhorar a performance usando iniciações aleatórias diferentes e escolhendo aquela com menor custo final
- Boa prática é iniciar os centróides com valores de observados na base de dados.



Como escolher K

Método da quina para escolher K



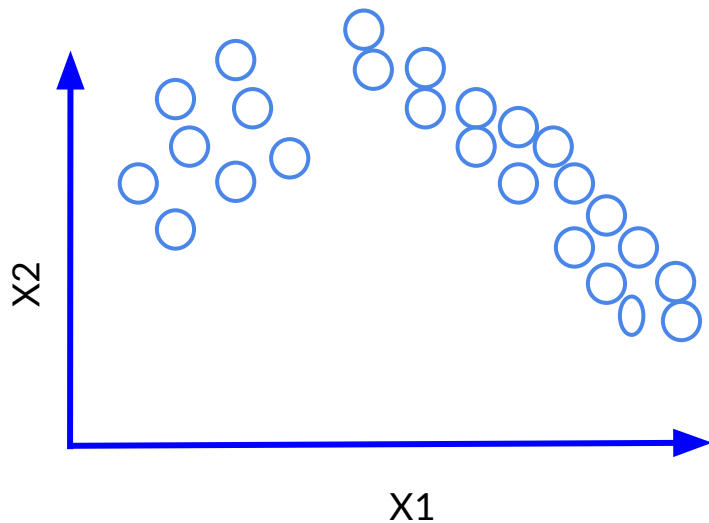


Mixturas Gaussianas

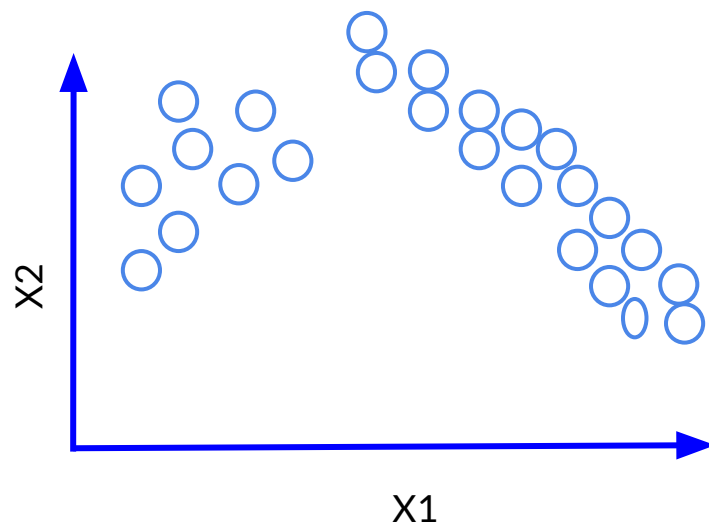
Ótimo Global versus ótimo local



k-médias



Mixturas Gaussianas



Mixturas Gaussianas



- Mixturas Gaussianas é um outro algoritmo de agrupamento.
- Nele presumimos que os dados vem de k distribuições gaussianas
- Para cada uma dessas distribuições estimamos:
 - a média (j dimensões) μ_j
 - a matriz de variância-covariância Ω_j
 - a proporção dos dados que vem de cada gaussiana $\sum_{i=1}^k \pi_i = 1$
- Para tal usamos o método de máxima verosimilhança
- Isso permite que os grupos se pareçam não apenas como bolas nas J dimensões, mas também como elipses.
- Além disso temos uma medida de quão provável é que cada observação venha de cada gaussiana.
- Isso permite detectar anomalias nos dados



Sistemas de Recomendação

Sistemas de recomendação são onipresentes

Related to items you've viewed [See more](#)



New for you [See more](#)



Formulação do problema



| Shows\Pessoa | André | Breno | Ricardo | Rodrigo | Tiago |
|-----------------|-----------|------------|---------|---------|------------|
| Game of Thrones | ★★★★ ★ | ★★★★★ | ★★★★ | | ★★★★ |
| Attack on Titan | ★★★★ | | ★★★★★ | ★★★★ | ★★★★ |
| Billions | | ★★★★★ ★ | | | ★★★★ |
| Suits | | ★★★★★ | | ★★ | |
| The wire | | | | ★★★★★ | ★★★★★ ★ |
| Narcos | | ★ | | | ★★★★★ |

Notação matemática



$$R = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad Y = \begin{bmatrix} 4 & 4 & & 3 & 3 \\ 3 & & 4 & 3 & 3 \\ & 5 & & & 3 \\ & 4 & & 2 & \\ & 1 & & & 4 \end{bmatrix}$$

n_u = número de usuários

n_s = número de shows



Recomendação baseada no conteúdo

Recomendação baseada no conteúdo

| Shows | André | Breno | Ricardo | Rodrigo | Tiago | X1 Fantasia | X2 Trabalho | X3 Violência |
|-----------------|-------|-------|---------|---------|-------|----------------|----------------|-----------------|
| Game of Thrones | 4 | 4 | 3 | | 3 | 0.99 | 0.1 | 0.9 |
| Attack on Titan | 3 | | 4 | 3 | 3 | 0.8 | 0.05 | 0.95 |
| Billions | | 5 | | | 3 | 0.3 | 0.95 | 0.1 |
| Suits | | 4 | | 2 | | 0.4 | 0.85 | 0.2 |
| The wire | | | | 4 | 5 | 0.1 | 0.45 | 0.8 |
| Narcos | | 1 | | | 4 | 0.05 | 0.55 | 0.8 |

Recomendação baseada no conteúdo



- Para cada usuário, podemos ter uma regressão linear
- Estimamos o quão cada pessoa valoriza cada conteúdo baseado nos shows que ela deu nota
- Estimamos o quão Breno valoriza shows de fantasia, trabalho e violência
- Em seguida podemos prever as notas dos shows que ele não viu, baseado em seus conteúdos
- Podemos usar outros modelos de machine learning além de regressão linear.
- Mas precisamos de uma base de dados onde todo mundo deu notas para vários shows!!
- Também precisamos saber o conteúdo dos shows!!



Problema sem conteúdo

Filtro Colaborativo



| Shows | André | Breno | Ricardo | Rodrigo | Tiago | X1 | X2 | X3 |
|-----------------|-------|-------|---------|---------|-------|----|----|----|
| Game of Thrones | 4 | 4 | 3 | | 3 | ?? | ?? | ?? |
| Attack on Titan | 3 | | 4 | 3 | 3 | ?? | ?? | ?? |
| Billions | | 5 | | | 3 | ?? | ?? | ?? |
| Suits | | 4 | | 2 | | ?? | ?? | ?? |
| The wire | | | | 4 | 5 | ?? | ?? | ?? |
| Narcos | | 1 | | | 4 | ?? | ?? | ?? |

Estimado conteúdo iterativamente

- Dados os gostos θ , podemos estimar os conteúdos x que minimizam os custos

$$\min_{x_1, x_2, x_3} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} (\theta_j x_i - y_{ij})^2 + \frac{\lambda}{2} \left(\sum_{i=1}^{n_m} \sum_{k=1}^n x_{ki}^2 \right)$$

- Dados os conteúdos x , podemos estimar os gostos θ que minimizam os custos

$$\min_{\theta_1, \theta_2, \theta_3} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} (\theta_j x_i - y_{ij})^2 + \frac{\lambda}{2} \left(\sum_{i=1}^{n_m} \sum_{k=1}^n \theta_{ki}^2 \right)$$

- Logo podemos iniciar aleatoriamente todos os parâmetros, depois calcular X , depois calcular θ , calcular X , até convergir.

$$\Theta \rightarrow X \rightarrow \Theta \rightarrow X \rightarrow \dots$$



Filtro Colaborativo

Filtro Colaborativo faz isso diretamente



- Basta resolver simultaneamente para gostos e conteúdos

$$\min_{\Theta, X} \frac{1}{2} \sum_{i,j:r(i,j)=1} (\theta_j x_i - y_{ij})^2 + \frac{\lambda}{2} \left(\sum_{i=1}^{n_m} \sum_{k=1}^n x_{ki}^2 \right) + \frac{\lambda}{2} \left(\sum_{i=1}^{n_u} \sum_{k=1}^n \theta_{ki}^2 \right)$$

Mais sobre filtro colaborativo



Alguns detalhes importantes:

- Quando estimamos os conteúdos, não sabemos o que eles significam
- Podemos usar os conteúdos estimados para encontrar shows parecidos. Basta olhar para filmes cuja distância dos conteúdos é pequena
- A média das notas dos filmes nos ajuda a gerar recomendações para pessoas sem nenhuma nota
- Isso é só uma breve introdução a filtro colaborativo. Há várias técnicas de deep learning sendo desenvolvidas nesse tópico.