

Algoritmos baseados em filtragem colaborativa

Prof. Dr. Marcelo G. Manzato



Filtragem Colaborativa (FC)

Abordagem mais conhecida para se gerar recomendações

- Usada pela maioria dos sistemas comerciais
- Bem entendida, vários algoritmos e versões
- Aplicável em praticamente qualquer domínio (livros, filmes, jogos, ...)

Usar a “sabedoria da multidão” para recomendar itens.

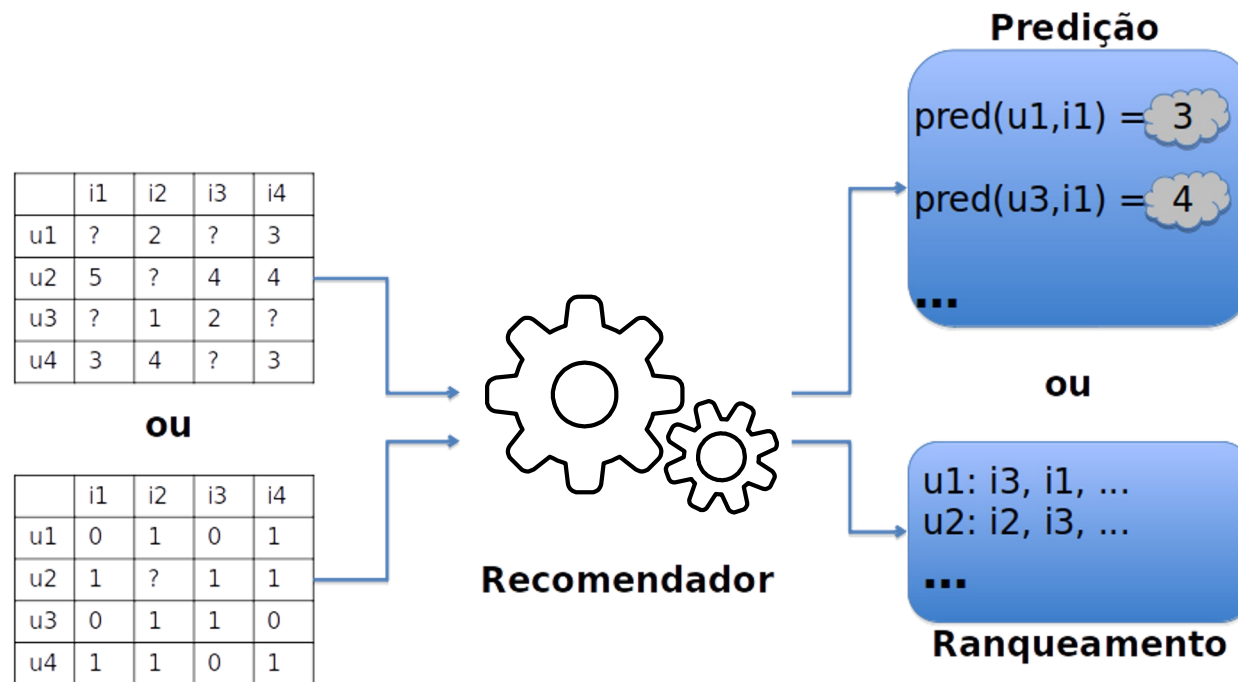
Suposições

- Usuários fornecem avaliações para itens visitados
- Indivíduos que tinham gostos similares no passado continuarão tendo gostos similares no futuro
- Preferências permanecem estáveis e consistentes ao longo do tempo



Tipos de entradas e saídas

(Abordagens tradicionais)



Tipos de Filtragem Colaborativa

A FC pode ser dividida em:

- Baseada em memória
- Baseada em modelo

Abordagens baseadas em memória, podem ser subdivididas em:

1

Vizinhança de usuários

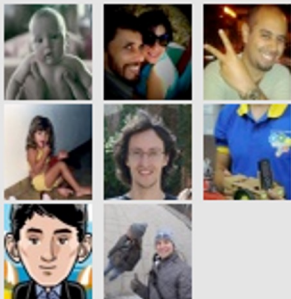
2

Vizinhança de itens

FC baseada em vizinhança de usuários

Friends' Favorites

Based on these friends:

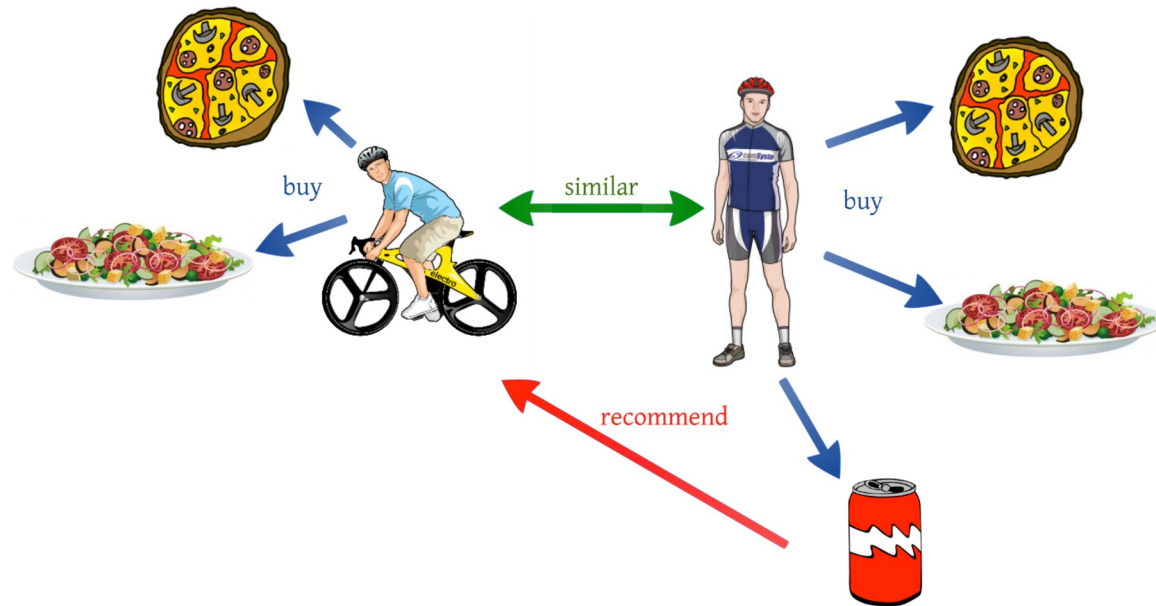


Algoritmo

Dado um usuário u e um item i ainda não visto por u :

1. Encontre um conjunto de usuários que tenham preferências parecidas com u e que tenham avaliado i ;
1. Use (por exemplo) a média de suas avaliações para prever o nível de satisfação de u por i ;
1. Faça isso para todos os itens que u ainda não conhece, e recomende os melhores avaliados.

Exemplo

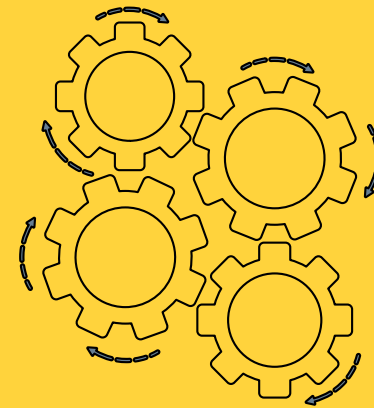


Exemplo

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice	4		3	4	?
Bob	1	2	5		3
Carlos	1			5	
Débora		3	4	5	3
Érica	2		5	4	5

Algumas questões iniciais

- Como saber quais usuários são similares?
- Como calcular a similaridade?
- Quantos vizinhos devemos considerar?
- Como calcular uma previsão ou ranking com base nas avaliações dos vizinhos?



Na prática

Similaridades:
Pearson,
Cosseno,
Jaccard, etc.

$$\begin{aligned} r_{\text{Alice}} &= (4+3+4)/3 = 3.66 \\ r_{\text{Bob}} &= (1+2+5+3)/4 = 2.75 \\ r_{\text{Carlos}} &= (1+5)/2 = 3 \\ r_{\text{Débora}} &= (3+4+5+3)/4 = 3.75 \\ r_{\text{Érica}} &= (2+5+4+5)/4 = 4 \end{aligned}$$

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice	4		3	4	?
Bob	1	2	5		3
Carlos	1			5	
Débora		3	4	5	3
Érica	2		5	4	5

$$\text{sim}(\text{Alice}, \text{Bob}) = -0.98$$

$$\text{sim}(\text{Alice}, \text{Carlos}) = 0$$

$$\text{sim}(\text{Alice}, \text{Débora}) = 0.27$$

$$\text{sim}(\text{Alice}, \text{Érica}) = -0.73$$

$$\text{sim}(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

U_u : conj. de
usuários mais
similares a u
que avaliaram
item i

$$\text{pred}(u, i) = \bar{r}_u + \frac{\sum_{v \in U_u} \text{sim}(u, v)(r_{vi} - \bar{r}_v)}{\sum_{v \in U_u} \text{sim}(u, v)}$$

Predição (k = 2):

$$\text{pred}(\text{Alice}, \text{Item 5}) = 3.66 + 0.27 \cdot (3 - 3.75) / 0.27 = 2.91$$

Score (k = 2):

$$\text{score}(\text{Alice}, \text{Item 5}) = 0.27$$

Cuidados

Número de itens co-avaliados

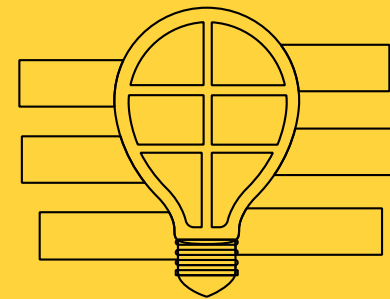
- Em especial para bases muito esparsas, esse número pode ser insuficiente

Escolha do no. de vizinhos mais próximos (k)

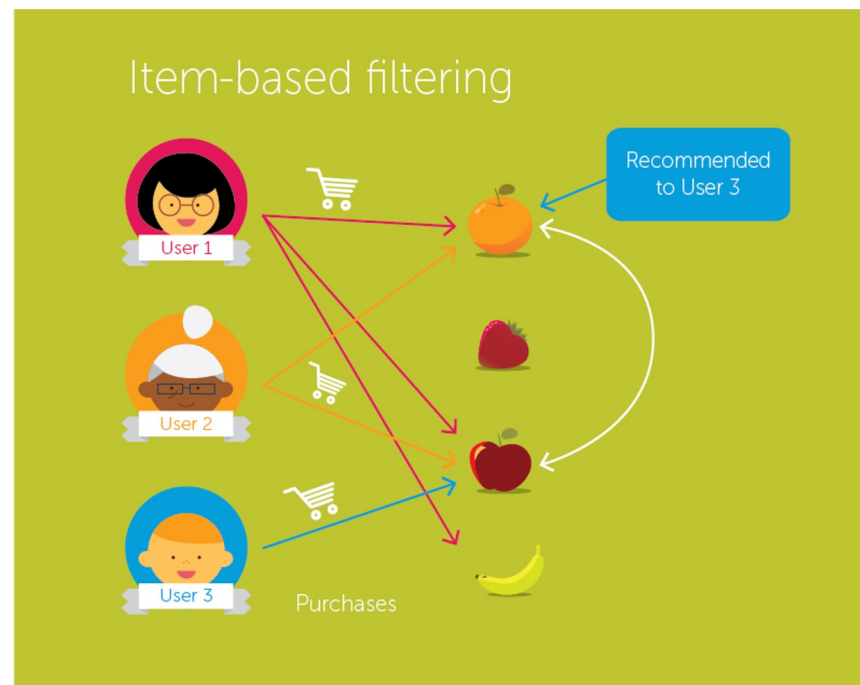
- Valores muito baixos ou muito altos podem reduzir a acurácia do sistema

Escalabilidade

- Normalmente sistemas têm milhares de usuários e milhares de produtos



FC baseada em vizinhança de itens



Algoritmo


Dado um usuário u e um item i ainda não visto por u :

1. Encontre um conjunto de itens que tenham avaliações parecidas com i e que tenham sido avaliados por u ;
1. Use (por exemplo) a média de avaliações de u desses itens para prever o nível de satisfação de u por i ;
1. Faça isso para todos os itens que u ainda não conhece, e recomenda os melhores avaliados

Pearson,
Cosseno,
Jaccard, etc.

Na prática

I_u : conj. de
itens mais
similares a i
que foram
avaliados por
 u



	Item 1	Item 2	Item 3	Item 4	Item 5
Alice	4		3	4	?
Bob	1	2	5		3
Carlos	1			5	
Débora		3	4	5	3
Érica	2		5	4	5

$$\begin{aligned} r_{\text{Item 1}} &= (4+1+1+2)/4 = 2 \\ r_{\text{Item 2}} &= (2+3)/2 = 2.5 \\ r_{\text{Item 3}} &= (3+5+4+5)/4 = 4.25 \\ r_{\text{Item 4}} &= (4+5+5+4)/4 = 4.5 \\ r_{\text{Item 5}} &= (3+3+5)/3 = 3.66 \end{aligned}$$

Similaridade entre itens:

$$sim(i, j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_i)^2} \sqrt{\sum_{u \in U_{ij}} (r_{uj} - \bar{r}_j)^2}}$$

$$\begin{aligned} sim(\text{Item 5}, \text{Item 1}) &= 0.44 \\ sim(\text{Item 5}, \text{Item 2}) &= 0 \\ sim(\text{Item 5}, \text{Item 3}) &= 0.37 \\ sim(\text{Item 5}, \text{Item 4}) &= -0.94 \end{aligned}$$

Predição (k = 2):

$$pred(\text{Alice}, \text{Item 5}) = (0.44 \cdot 4 + 0.37 \cdot 3) / (0.44 + 0.37) = 3.54$$

Score (k = 2):

$$score(\text{Alice}, \text{Item 5}) = 0.44 + 0.37 = 0.81$$

Abordagens de FC baseadas em modelo

Algoritmos mais conhecidos

- Fatoração de matrizes via:
 - Singular Value Decomposition
 - Gradiente Descendente
- FunkSVD
- SVD++
- Factorization Machines
- Etc.

Singular Value Decomposition

Técnica algébrica que decompõe uma matriz M em um produto de três matrizes:

$$M = U \Sigma V^T$$

The diagram illustrates the SVD decomposition of matrix M . Matrix M is represented by a green square with dimensions t (rows) and d (columns). It is equal to the product of three matrices: U , Σ , and V^T . Matrix U is a green square with dimensions t (rows) and f (columns). Matrix Σ is a green square with dimensions f (rows) and f (columns). Matrix V^T is a green square with dimensions d (rows) and f (columns). Brackets indicate the dimensions of each matrix.

Usando apenas os k primeiros valores singulares (fatores mais importantes), é possível aproximar M .

Singular Value Decomposition

- SVD: $M_k = U_k \times \Sigma_k \times V_k^T$

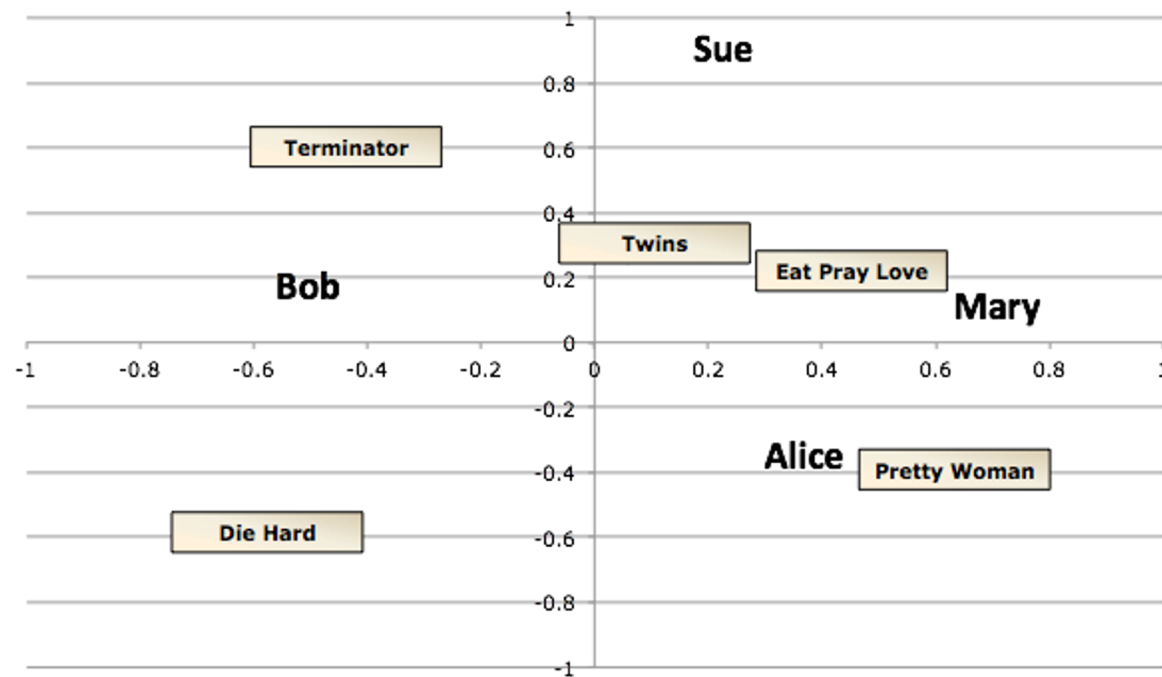
U_k	Dim1	Dim2
Alice	0.47	-0.30
Bob	-0.44	0.23
Mary	0.70	-0.06
Sue	0.31	0.93

V_k^T	Terminator	Die Hard	Twins	Eat Pray Love	Pretty Woman
Dim1	-0.44	-0.57	0.06	0.38	0.57
Dim2	0.58	-0.66	0.26	0.18	-0.36

- Prediction: $\hat{r}_{ui} = \bar{r}_u + U_k(Alice) \times \Sigma_k \times V_k^T(EPL)$
 $= 3 + 0.84 = 3.84$

Σ_k	Dim1	Dim2
Dim1	5.63	0
Dim2	0	3.23

Singular Value Decomposition



Fatoração de matrizes

Problemas

- Lentidão na decomposição
- Valores desconhecidos (ratings) são interpretados como "zero"

Solução:

- Usar apenas valores conhecidos da matriz de interações
- Treinar as matrizes U e V com gradiente descendente, minimizando o erro entre a nota real e a predita

Fatoração de Matrizes

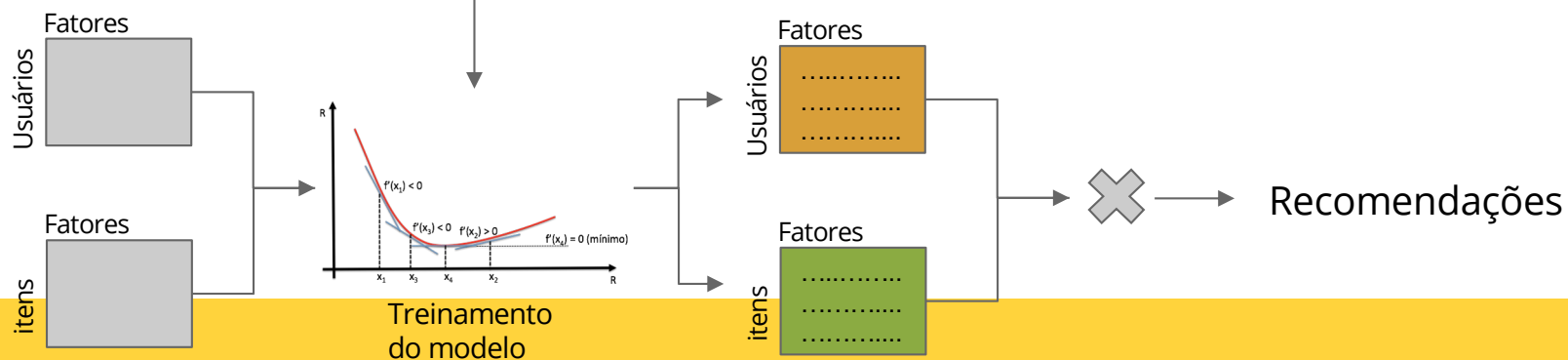


Jessica	5	2	4	3	2	3
Marta	4	3	5	4	3	2
Jose	1	5	3	4	4	5
Dave	1	?	2	3	4	2

Base de treino

Predição: $\hat{r}_{ui} = b_{ui} + \sum_{f=1}^k p_{uf} q_{if}$ onde $b_{ui} = \mu + b_u + b_i$

Função custo: $\sum_{(u,i) \in K} (r_{ui} - \hat{r}_{ui})^2$
 $\min_{b_u, p_u, q_u} \sum_{(u,i) \in K} \left(r_{ui} - \mu - b_u - b_i - \sum_{f=1}^k p_{uf} q_{if} \right)^2$





Filtragem Colaborativa

Baseada em memória

- Boa para detectar relacionamentos fortes entre itens próximos entre si (visão local)

Baseada em modelo

- Boa para capturar relações não aparentes na base de dados (visão global)

Filtragem colaborativa

Vantagens

- Técnica bem estudada e entendida
- Funciona bem em vários domínios
- Não precisa de conhecimento especializado

Desvantagens

- Requer colaboração da comunidade
- Esparsidade dos dados
- Sem integração com outras fontes de conhecimento
- Na baseada em modelos, é difícil explicar as recomendações