CeMEAI
CEPID - Centro de Ciências Matemáticas Aplicadas à Indústria

ICMC USP SÃO CARLOS

Algoritmos baseados em filtragem colaborativa

Prof. Dr. Marcelo G. Manzato e Arthur Fortes da Costa

Filtragem Colaborativa (FC)

Abordagem mais conhecida para se gerar recomendações

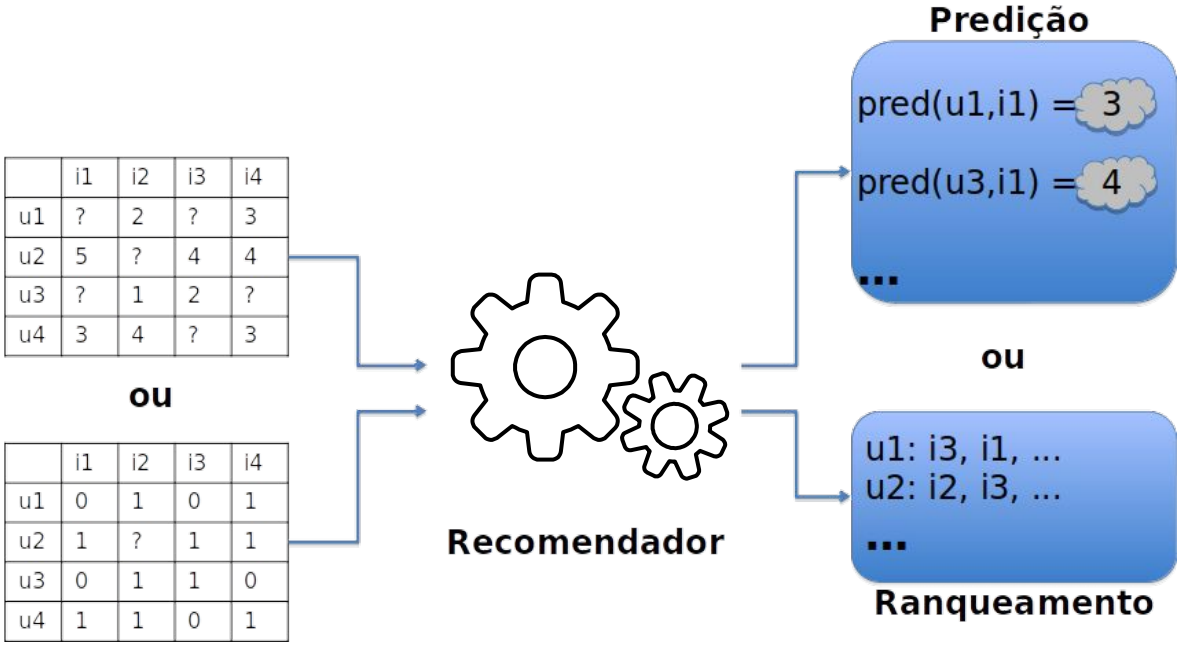
- Usada pela maioria dos sistemas comerciais
- Bem entendida, vários algoritmos e versões
- Aplicável em praticamente qualquer domínio (livros, filmes, jogos, ...)

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)



	i1	i2	i3	i4
u1	?	2	?	3
u2	5	?	4	4
u3	?	1	2	?
u4	3	4	?	3

	i1	i2	i3	i4
u1	0	1	0	1
u2	1	?	1	1
u3	0	1	1	0
u4	1	1	0	1

Tipos de Filtragem Colaborativa

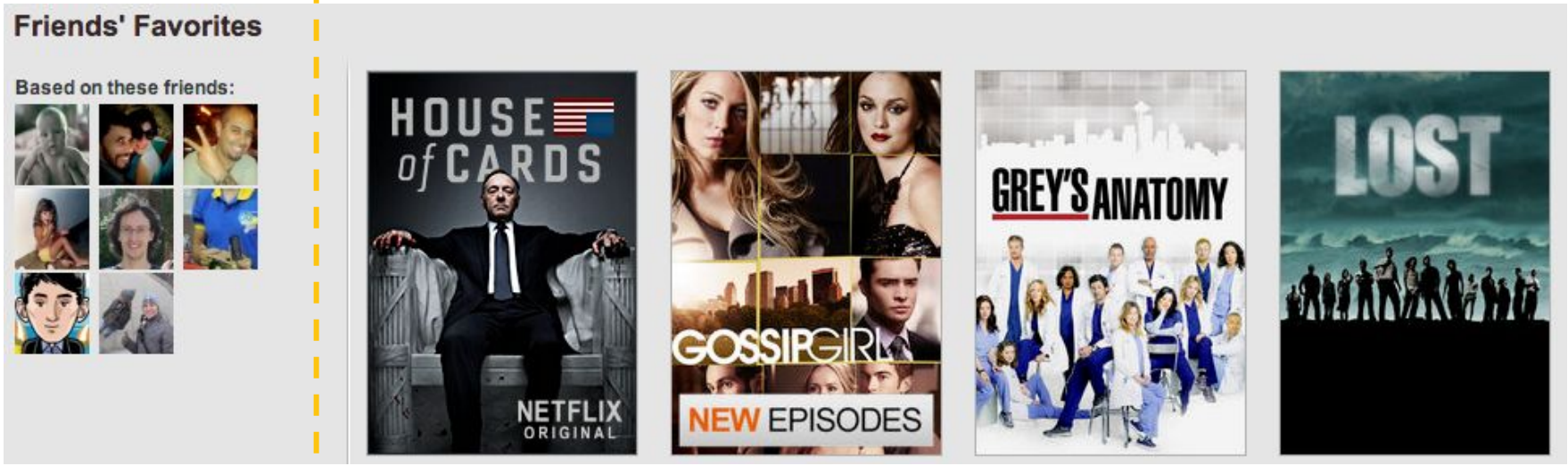
A FC pode ser dividida em:

- 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



Algoritmo

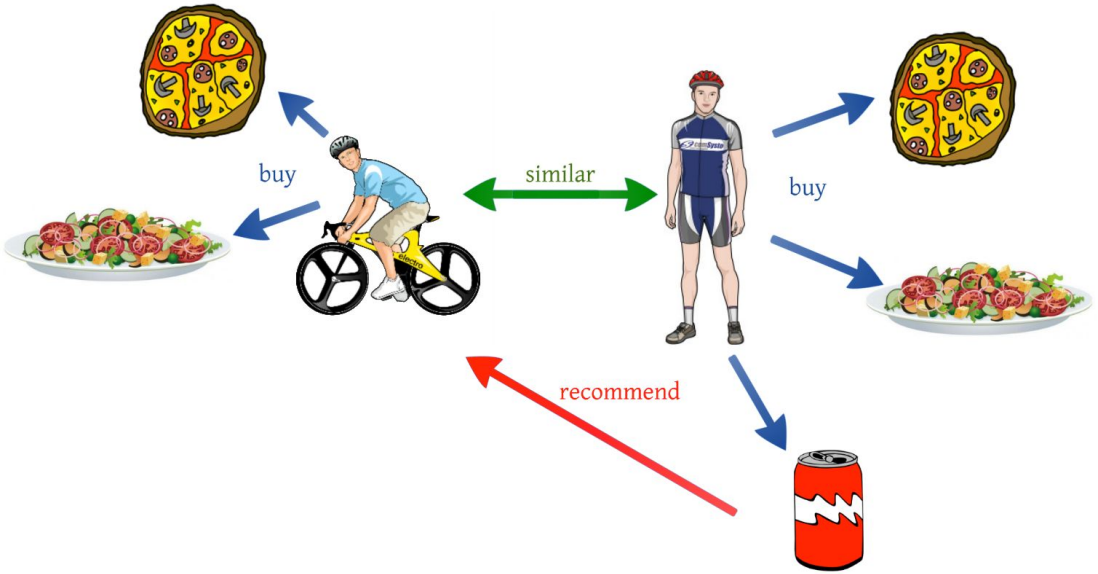
Dado um usuário *u* e um item *i* ainda não visto por *u*:

1. Encontre um conjunto de usuários que tenham preferências parecidas com ***u*** e que tenham avaliado ***i***;

2. Use (por exemplo) a média de suas avaliações para predizer o nível de satisfação de ***u*** por ***i***;

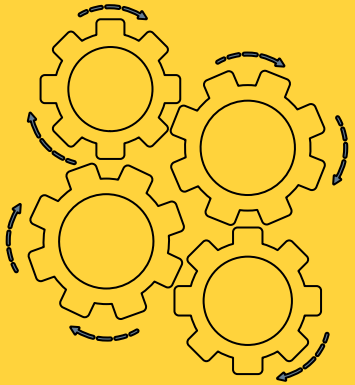
3. Faça isso para todos os itens que ***u*** ainda não conhece, e recomende os melhores avaliados.

Exemplo



Algumas questões iniciais

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



Na prática

Similaridades: Pearson, Cosseno, Jaccard, etc.

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

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

sim = 0,83

sim = 0,60

sim = 0,00

sim = -0,76

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

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

K = 2

$\bar{r}_{Alice} = 4$ $\bar{r}_{User1} = 2.4$ $\bar{r}_{User2} = 3.8$ ...

→ pred(Alice, Item5) = 4 + [1 / (0.83 + 0.60)] * [0.83 * (3 – 2.4) + 0.60 * (5 – 3.8)] = 4.85 (média ponderada)

→ score(Alice, Item5) = 0.83 + 0.60 = 1.43 (soma das similaridades)

Cuidados

Número de itens co-avaliados

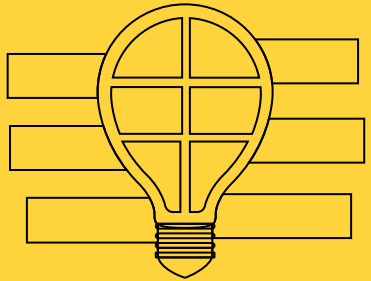
- Em especial para bases muito esparsas, esse número pode ser insuficiente

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

Usuário assistiu:

Recomendação

Algoritmo

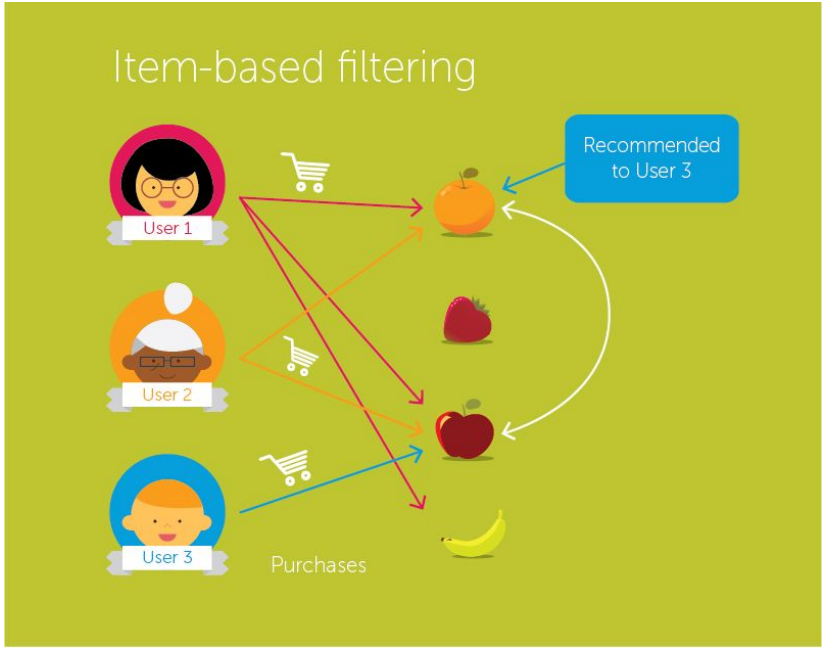
Dado um usuário *u* e um item *i* ainda não visto por *u*:

1. Encontre um conjunto de itens que tenham avaliações parecidas com ***i*** e que tenham sido avaliados por ***u***;

2. Use (por exemplo) a média de avaliações de ***u*** desses itens para predizer o nível de satisfação de ***u*** por ***i***;

3. Faça isso para todos os itens que ***u*** ainda não conhece, e recomenda os melhores avaliados

Exemplo


Item-based filtering
Recommended to User 3
User 1
User 2
User 3
Purchases

Na prática

sim(Item5,Item4)
sim(Item5,Item3)
sim(Item5,Item2)
sim(Item5,Item1)

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

k = 2 itens mais similares a Item5

pred(Alice, Item5)

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

$$pred(u,i) = \frac{\sum_{j \in I_u} sim(i,j) r_{uj}}{\sum_{j \in I_u} sim(i,j)}$$

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

Pré-processamento para FC

FC baseada em vizinhança de itens não resolve por si só o problema da escalabilidade.

Por outro lado

- Possibilidade de calcular antecipadamente (off-line) a similaridade entre todos os pares de itens
- Similaridade de itens tende a ser mais estável do que a similaridade de usuários
- Em tempo de execução, vizinhança usada é pequena, já que contém apenas itens que o usuário avaliou

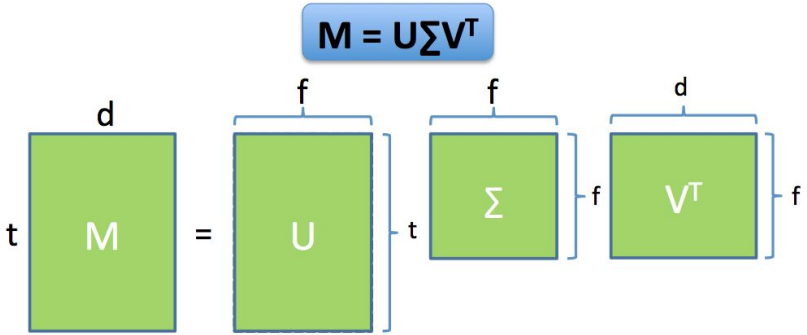
Abordagens de FC baseadas em modelo

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:



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

Σ_k	Dim1	Dim2
Dim1	5.63	0
Dim2	0	3.23

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

  = 3 + 0.84 = 3.84

Singular Value Decomposition

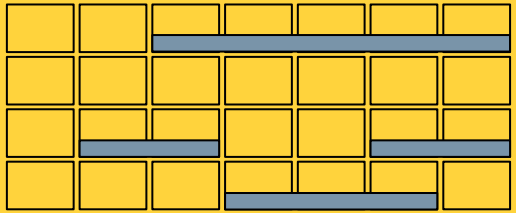
Fatoração de matrizes

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

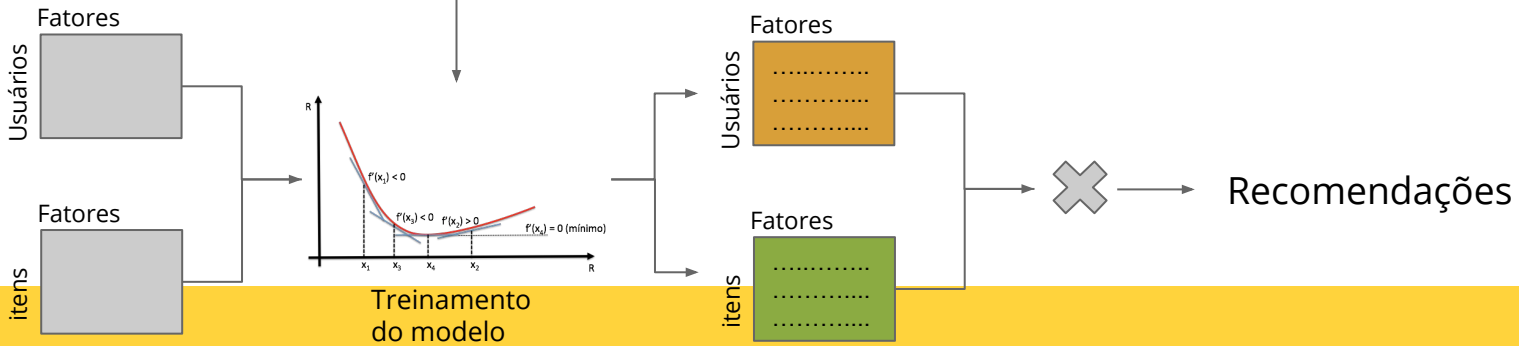
Jessica	5	2	4	3	2	3
Marta	4	3	5	4	3	2
Jose	1	5	3	4	4	5
Dave	1	?	2	3	4	2

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: $\sum_{(u,i) \in K} (r_{ui} - \hat{r}_{ui})^2$

$$\min_{b_*, p_*, q_*} \sum_{(u,i) \in K} \left(r_{ui} - \mu - b_u - b_i - \sum_{f=1}^{k} p_{uf} q_{if} \right)^2$$



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