

SCC0284 – Sistemas de Recomendação

Aula 02: Filtragem Colaborativa
Parte 1

(mmanzato@icmc.usp.br)

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



Filtragem Colaborativa (FC)

- 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

FC tradicional

- Tipos de entrada e saída



Sobre avaliações

- Avaliações explícitas
 - Escalas bem definidas (1 a 5, -10 a +10, etc.)
 - Reflete precisamente as preferências do usuário
 - Porém requer ação do usuário
 - Número de avaliações pode ser escasso → matrizes esparsas → recomendações imprecisas



Sobre avaliações

- Avaliações implícitas
 - Coletadas pelo sistema durante a navegação do usuário
 - Cliques, visualizações, compras, etc.
 - Não requer esforço adicional do usuário, por isso são abundantes
 - Problema: como interpretar corretamente o comportamento do usuário no sistema?



FC tradicional

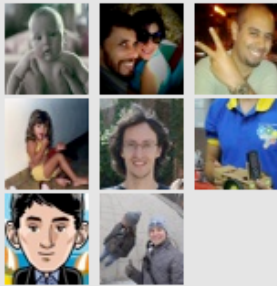
- A FC pode ser dividida em:
 - Baseada em memória
 - Baseada em modelo
- FC baseada em memória pode ser subdividida em:
 - Vizinhança entre usuários
 - Vizinhança entre itens

FC baseada em vizinhança de usuários

Recomendação

Friends' Favorites

Based on these friends:



FC baseada em vizinhança de usuários

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

Exemplo

- Dada uma base de avaliações (escala de 1 a 5):

	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

- Determinar se **Alice** irá gostar ou não de **Item5**

Exemplo

- Algumas questões iniciais:
 - Como saber quais usuários tem gostos parecidos com Alice?
 - Como calcular a similaridade?
 - Quantos vizinhos devemos considerar?
 - Como calcular uma predição com base nas avaliações dos vizinhos?

	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

Similaridade entre usuários

- Uma métrica de similaridade bastante popular é a Correlação de Pearson
- Dados:
 - u, v : usuários
 - r_{ui} : avaliação do usuário u para um item i
 - I_{uv} : conjunto de itens avaliados por ambos u e v

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

Possíveis valores
entre -1 e +1

Similaridade entre usuários

- Dados:

- u, v : usuários
- r_{ui} : avaliação do usuário u para um item i
- I_{uv} : conjunto de itens avaliados por ambos u e v

	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 = 0,85

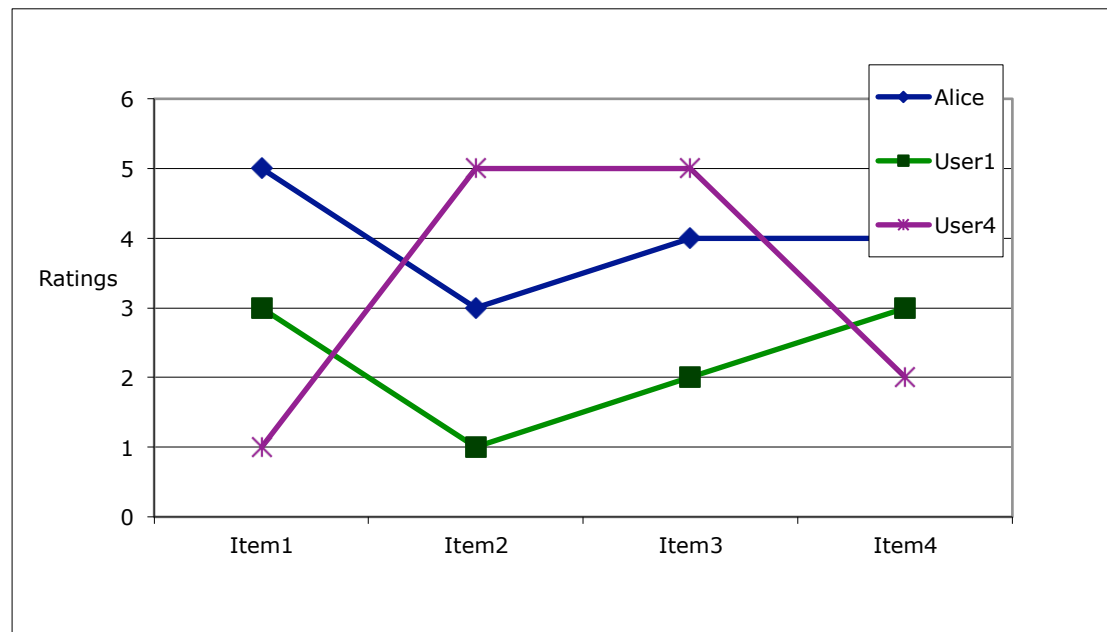
sim = 0,00

sim = 0,70

sim = -0,79

Similaridade entre usuários

- A Correlação de Pearson leva em consideração o comportamento de avaliações de cada usuário



Predição de avaliações

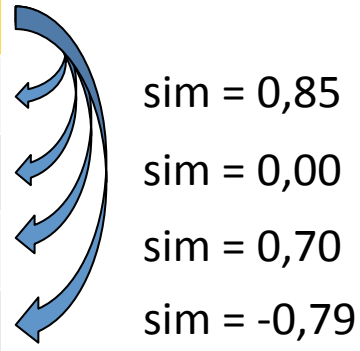
- Uma função bastante conhecida:

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

- onde U_u são os k usuários mais próximos de u
- Função acima considera o viés de cada usuário

Exemplo

	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 = 0,85
sim = 0,00
sim = 0,70
sim = -0,79

- Assim, a predição de **Alice** para **Item5** será (assumindo $k = 2$):

$$\begin{aligned} & \text{pred}(\text{Alice}, \text{Item5}) \\ &= 4 + 1 / (0,85 + 0,70) * (0,85 * (3 - 2,4) + 0,70 * (5 - 3,8)) \\ &= 4,87 \end{aligned}$$

Alguns cuidados...

- Nem todas avaliações de vizinhos podem ser igualmente importantes
 - Concordância em itens comumente apreciados não é tão informativo quanto a concordância em itens controversos
- Número de itens co-avaliados
 - Em especial para bases muito esparsas, esse número pode ser insuficiente

Alguns cuidados...

- Escolha do número 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



FC baseada em vizinhança de itens

- Dado um usuário u e um item i ainda não visto por u :
 - Encontre um conjunto de itens que tenham avaliações parecidas com i e que tenham sido avaliados por u
 - Use (por exemplo) a média de avaliações de u desses itens para predizer o nível de satisfação de u por i
 - Faça isso para todos os itens que u ainda não conhece, e recomende os melhores avaliados

FC baseada em vizinhança de itens

- Abordagens que exploram a similaridade entre itens para calcular previsões
- Similaridade é calculada no espaço de itens:

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

- onde U_{ij} é o conjunto de usuários que avaliaram ambos itens i e j

FC baseada em vizinhança de itens

- Função de predição:

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

- onde I_u é o conjunto dos k itens mais similares a i que foram avaliados por u
- Valor de k geralmente é uma constante (cerca de 20 a 50 normalmente)
- Pode-se considerar o viés de cada item na fórmula acima (como?)

Exemplo

$\text{sim}(\text{Item5}, \text{Item4})$
 $\text{sim}(\text{Item5}, \text{Item3})$
 $\text{sim}(\text{Item5}, \text{Item2})$
 $\text{sim}(\text{Item5}, \text{Item1})$

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

k = 2 itens mais similares a Item5



$\text{pred}(\text{Alice}, \text{Item5})$



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

	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

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

Requisitos de memória

- Na teoria, até N^2 similaridades precisam ser armazenadas
 - N = número de itens
- Na prática, esse valor é menor (itens sem nenhuma co-avaliação)
- Outras otimizações em memória:
 - Limiar mínimo para co-avaliações
 - Limitar o valor de k (pode afetar a acurácia da recomendação)

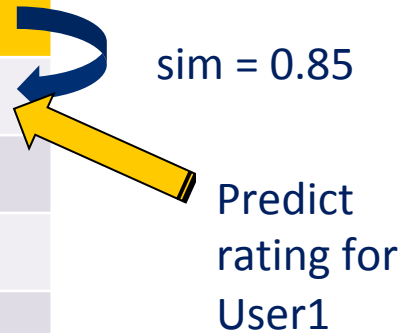
Problemas relacionados com esparsidade

- Partida-fria
 - Como recomendar novos itens? O que recomendar para novos usuários?
- Abordagens triviais
 - Pedir/forçar que o usuário avalie um conjunto inicial de itens
 - Usar outro método de recomendação (baseado em conteúdo, demográfico, não-personalizado, etc.)
 - Usar uma avaliação padrão (e.g. média) para aqueles itens que somente um dos dois usuários a serem comparados avaliou

Variações da FC para lidar com esparsidade

- Recursive CF (Zhang & Pu 2007)
 - Selecionar vizinho **v** mais próximo de **u**, mesmo que não tenha avaliado o item-alvo **i**
 - Aplicar a FC recursivamente e prever uma avaliação de **v** para **i**
 - Usar essa avaliação predita ao invés da avaliação de um vizinho mais longe

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

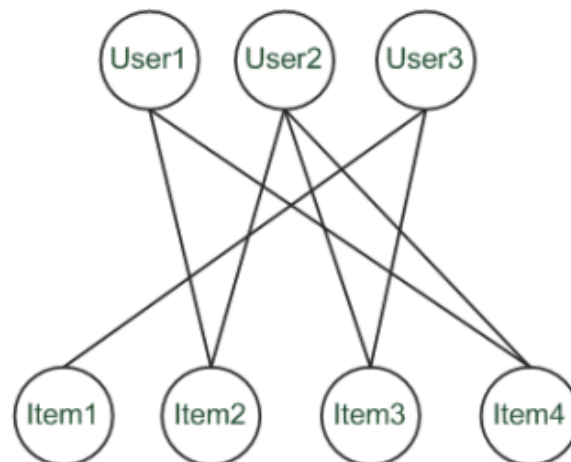


sim = 0.85

Predict rating for User1

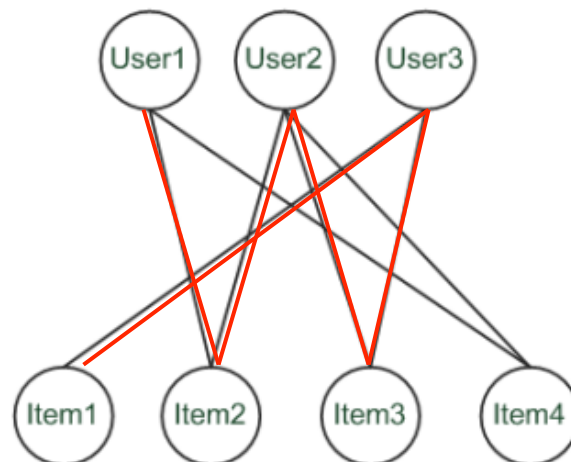
Variações da FC para lidar com esparsidade

- Spreading activation (Huang et al., 2004)
 - Explora a suposta “transitividade” de gostos de usuários para enriquecer a matriz com informação adicional
 - Assuma que estamos calculando uma recomendação para **User1**
 - Ao usar a FC tradicional, **User2** será selecionado como mais próximo de **User1** (ambos consumiram **Item2** e **Item4**)
 - Portanto, **Item3** será recomendado para **User1**, pois **User2** também consumiu esse item



Variações da FC para lidar com esparsidade

- Spreading activation (Huang et al., 2004)
 - FC tradicional usa caminhos de comprimento 3, i.e., **Item3** é relevante para **User1** porque existe o caminho (User1->Item2->User2->Item3)
 - Em bases esparsas, tais caminhos são poucos, então a idéia é considerar caminhos mais longos para computar as recomendações
 - Por exemplo, caminho de comprimento 5
 - Neste caso, **Item1** também pode ser recomendado a **User1**



Regras de Associação

- Comumente usado para análise de comportamento de compra
 - Procura detectar regras como: “se um cliente comprar cerveja então 70% de chance dele também comprar amendoim”
- Algoritmos:
 - Podem detectar regras da forma $X \rightarrow Y$ (e.g. cerveja \rightarrow amendoim) de um conjunto de transações de compras $D=\{t_1, t_2, \dots, t_n\}$
 - Métricas de qualidade: suporte e confiança

Regras de Associação

- Suporte e confiança
 - Métricas usadas como **limiar** para filtrar regras irrelevantes
 - Seja:

$$\begin{array}{ll} \text{suporte} = \frac{\sigma(X \cup Y)}{|D|} & \longrightarrow \frac{\text{no. transações contendo X e Y}}{\text{no. transações}} \\ \text{confiança} = \frac{\sigma(X \cup Y)}{\sigma(X)} & \longrightarrow \frac{\text{no. transações contendo X e Y}}{\text{no. transações contendo X}} \end{array}$$

onde:

$$\sigma(X) = |\{X \mid X \subseteq t_i, t_i \in D\}|$$

$D = \{t_1, t_2, \dots, t_n\}$ (transações de compras \rightarrow e.g. qtde. de usuários)

Regras de Associação

- Exemplo: Recomendações para Alice
 - Determinar regras “relevantes” com base nas transações de Alice
 - I.e., regras que são disparadas a partir das transações de Alice
 - regra Item1 → Item5 é relevante pois Alice consumiu Item1
 - Determinar itens que não foram consumidos ainda por Alice
 - Ordenar os itens com base nos valores de confiança das regras

Matriz binarizada (1 = acima da média do usuário)

	Item1	Item2	Item3	Item4	Item5
Alice	1	0	?	0	?
User1	1	0	0	0	1
User2	1	0	1	0	1
User3	0	0	0	1	1
User4	0	1	1	0	0

Excluindo Alice, temos:

Regra: Item1 → Item5

suporte = 2/4

confiança = 2/2

Regra: Item1 → Item3

suporte = 1/4

confiança = 1/2

Métodos Probabilísticos

- Idéia básica
 - Dada a matriz de interações usuário/item
 - Determinar a probabilidade em que o usuário u irá gostar de um item i
 - Realizar a recomendação com base nessas probabilidades
- Cálculo das probabilidades é feito com base no Teorema de Bayes

Métodos Probabilísticos

- Teorema de Bayes
 - Qual a probabilidade do **Item5** ser acessado por **Alice**?
 - Probabilidade condicional: $P(\text{Item5} \mid X)$, onde
 - X = histórico de acessos / avaliações de Alice
 - Estimativa:

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)} \quad P(Y \mid X) = \frac{\prod_{i=1}^d P(X_i \mid Y)P(Y)}{P(X)}$$

- Suposição: acessos / avaliações são independentes (?)

Métodos Probabilísticos

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

$X = (\text{Item1}=1, \text{Item2}=3, \text{Item3}=3, \text{Item4}=2)$

$P(X) \rightarrow$ constante (omitir)

$P(Y)$:

$$P(\text{Item5}=1) = 2/4 = 0.5$$

$$P(\text{Item5}=2) = 0/4 = 0$$

$$P(\text{item5}=3) = 1/4 = 0.25$$

$$P(\text{Item5}=4) = 1/4 = 0.25$$

$$P(\text{item5}=5) = 0/4 = 0$$

$P(X \mid Y)$:

$P(X \mid \text{Item5}=1)$

$$\begin{aligned} &= P(\text{Item1}=1 \mid \text{Item5}=1) \times P(\text{Item2}=3 \mid \text{Item5}=1) \\ &\times P(\text{Item3}=3 \mid \text{Item5}=1) \times P(\text{Item4}=2 \mid \text{Item5}=1) \\ &= 2/2 \times 1/2 \times 1/2 \times 1/2 \approx 0.125 \end{aligned}$$

$P(X \mid \text{Item5}=2)$

$$\begin{aligned} &= P(\text{Item1}=1 \mid \text{Item5}=2) \times P(\text{Item2}=3 \mid \text{Item5}=2) \\ &\times P(\text{Item3}=3 \mid \text{Item5}=2) \times P(\text{Item4}=2 \mid \text{Item5}=2) \\ &= 0/0 \times \dots \times \dots \times \dots = 0 \end{aligned}$$

...

Outros métodos

- Slope One (Lemiere & Maclachlan 2005)
 - Baseado na diferença das avaliações de cada usuário

	Item1	Item5
Alice	2	?
User1	1	2

-

$$\text{pred}(\text{Alice}, \text{Item5}) = 2 + (2 - 1) = 3$$

- Esquema básico: usar a média dessas diferenças das co-avaliações para calcular uma predição
- De modo geral, encontrar uma função da forma $f(x) = x + b$

Outros métodos

- RF-Rec (Gedikli et al. 2011)
 - Predição baseada nas frequências de avaliações de usuários e itens
 - Esquema básico:

$$\hat{r}_{ui} = \arg \max_{r \in R} f_{user}(u, i) \times f_{item}(i, r)$$

– onde

- R : conjunto de todas as avaliações, i.e. $R = \{1, 2, 3, 4, 5\}$
- $f_{user}(u, r)$ e $f_{item}(i, r)$ representam a (frequência + 1) de uma avaliação r ter sido usada pelo usuário u ou atribuída ao item i

Outros métodos

- RF-Rec (Gedikli et al. 2011)

	Item1	Item2	Item3	Item4	Item5
Alice	1	1	?	5	4
User1	2		5	5	5
User2			1	1	
User3		5	2		2
User4	3		1	1	
User5	1	2	3		4

Rating 1: $(2 + 1) \times (2 + 1) = 9$

Rating 2: $(0 + 1) \times (1 + 1) = 2$

Rating 3: $(0 + 1) \times (1 + 1) = 2$

Rating 4: $(1 + 1) \times (0 + 1) = 2$

Rating 5: $(1 + 1) \times (1 + 1) = 4$

pred(Alice, Item3) = 1

Referências

- [Adomavicius and Tuzhilin 2005] Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (2005), no. 6, 734–749
- [Zhang and Pu 2007] A recursive prediction algorithm for collaborative filtering recommender systems, Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys '07) (Minneapolis, MN), ACM, 2007, pp. 57–64
- [Huang et al. 2004] Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering, ACM Transactions on Information Systems 22 (2004), no. 1, 116–142
- [Lemire and Maclachlan 2005] Slope one predictors for online rating-based collaborative filtering, Proceedings of the 5th SIAM International Conference on Data Mining (SDM '05) (Newport Beach, CA), 2005, pp. 471–480
- [Gedikli et al. 2011] RF-Rec: Fast and accurate computation of recommendations based on rating frequencies, Proceedings of the 13th IEEE Conference on Commerce and Enterprise Computing - CEC 2011, Luxembourg, 2011, forthcoming