

Sistemas de Recomendação

Filtragem Colaborativa *Memory-based*



PUC Minas

Filtragem Colaborativa (CF)

Premissa: as preferências passadas dos usuários revelam seus interesses.

- *Memory-based*: usa os ratings dos usuários para computar similaridades entre usuários e itens.
- *Model-based*: constrói-se modelos de mineração de dados, machine learning e outros com base nas interações entre usuários e itens.

Utiliza as informações passadas para correlacionar 'coisas'.

- **User-based CF:**
 - Como os usuários similares a mim avaliaram o item i ?
- **Item-based CF:**
 - Como eu avaliei itens similares ao item i ?

User-based CF

Existem 5 passos fundamentais para definir modelos user-based:

1. Definir a vizinhança
2. Agregar os ratings
3. Normalizar os dados
4. Computar similaridades
5. Avaliar opções adicionais

1. Definir a vizinhança

Vizinhança refere-se aos usuários que serão utilizados para gerar a recomendação

Podemos considerar como vizinhos:

- Todos os usuários.
- Usuários aleatórios.
- Todos os usuários com um nível de similaridade acima de um threshold pré-definido.
- Os *top-k* usuários ranqueados pela similaridade.











Como encontrar vizinhos?

Quem são os vizinhos de u (i.e., que avaliaram o item i)?










			i			
x			1	3		2
y	1	2	5		4	1
	4			3		
u		3	?	5	4	
z		3	4		5	3

Como encontrar vizinhos?

			<i>i</i>							
										
<i>x</i> 			1	3		2				$sim(\vec{u}, \vec{x}) = 0.27$
<i>y</i> 	1	2	5		4		1			$sim(\vec{u}, \vec{y}) = 0.45$
	4			3	5		4			
<i>u</i> 		2	?		5	4		4		
<i>z</i> 		3	4		5		3			$sim(\vec{u}, \vec{z}) = 0.52$

Como encontrar vizinhos?

i

								
			1	3		2		
<i>y</i> 	1	2	5		4		1	
	4			3	5		4	
<i>u</i> 		2	?		5	4		4
<i>z</i> 		3	4		5		3	













$$\text{sim}(\vec{u}, \vec{y}) = 0.45$$

2-NN

$$\text{sim}(\vec{u}, \vec{z}) = 0.52$$

Como encontrar vizinhos?

E se nós precisamos prever os itens i e j ?

			i	j					
									
					1	3		2	
y		1	2	5	X	4		1	
		4			3	5		4	
u			2	?	?	5	4		4
z			3	4	X	5		3	














Global nearest neighbors don't necessarily cover all items that could be recommended

Quantos vizinhos?

- Na teoria, quanto mais melhor...
 - ... se você tiver uma boa métrica de similaridade!
 - O custo computacional é alto.
- Na prática:
 - Mais vizinhos significa mais ruídos.
 - Menos vizinhos significa menos cobertura.
- É comum definir entre 25 a 100 vizinhos.
 - 30 a 50 é suficiente para o cenário de filmes.

2. Agregar ratings






Como predizer se o usuário u vai gostar do item i ?

		i							
									
				1	3		2		
y		1	2	5		4		1	
		4			3	5		4	
u			2	?		5	4		4
z			3	4		5		3	

$$\text{sim}(\vec{u}, \vec{y}) = 0.45 \quad r_{yi} = 5$$

$$\text{sim}(\vec{u}, \vec{z}) = 0.52 \quad r_{zi} = 4$$

Como prever o rating?

		<i>i</i>							
									
<i>y</i>				1	3		2		
		1	2	5		4		1	
		4			3	5		4	
	<i>u</i> 		2	4.5		5	4		4
	<i>z</i> 		3	4		5		3	

$$\hat{r}_{ui} = \frac{\sum_{v \in N} \text{sim}(\vec{u}, \vec{v}) \times r_{vi}}{\sum_{v \in N} |\text{sim}(\vec{u}, \vec{v})|}$$

$$\hat{r}_{ui} = \frac{0.45 \times 5 + 0.52 \times 4}{0.45 + 0.52}$$

$$\hat{r}_{ui} = 4.5$$

3. Normalizar os Dados

Em geral, as notas dos usuários têm diferentes significados:

- Alguns usuários dão notas altas, outros baixas
- Alguns usam mais opções da escala de notas que outros

Técnicas de normalização compensam essas diferenças:

- Mean-center normalization
- Z-score normalization

Mean-centering

$$\hat{r}_{ui} = \frac{\sum_{v \in N} \text{sim}(\vec{u}, \vec{v}) \times r_{vi}}{\sum_{v \in N} |\text{sim}(\vec{u}, \vec{v})|}$$



$$\hat{r}_{ui} = \frac{\sum_{v \in N} \text{sim}(\vec{u}, \vec{v}) \times (r_{vi} - \bar{r}_v)}{\sum_{v \in N} |\text{sim}(\vec{u}, \vec{v})|}$$

(subtract neighbor's mean)



$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N} \text{sim}(\vec{u}, \vec{v}) \times (r_{vi} - \bar{r}_v)}{\sum_{v \in N} |\text{sim}(\vec{u}, \vec{v})|}$$

(add target user's mean)

Normalização Z-score

Consiste em duas etapas:

- Subtrai o valor pela média dos ratings
- Divide pelo desvio padrão dos ratings

$$Z = \frac{X - E[X]}{\sigma(X)}$$

Possui um melhor desempenho que o mean-centering

4. Computar Similaridades

Existem diversas abordagens para definir similaridades:

- Correlação de Pearson
- Similaridade de Cosseno

Correlação de Pearson

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

- Usada apenas sobre ratings em comum
- Considera a normalização pela média do usuário

Similaridade de Cosseno

Cosseno do ângulo entre os vetores dos usuários:

$$\text{sim}(\vec{u}, \vec{v}) = \cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| |\vec{v}|} = \frac{\sum_{i \in I} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I} r_{ui}^2} \sqrt{\sum_{i \in I} r_{vi}^2}}$$

- Em geral essa similaridade varia de -1 a 1.
- Para ratings não negativos, a similaridade varia de 0 a 1.
- Com ratings normalizados é quase equivalente a Pearson.

Problemas?

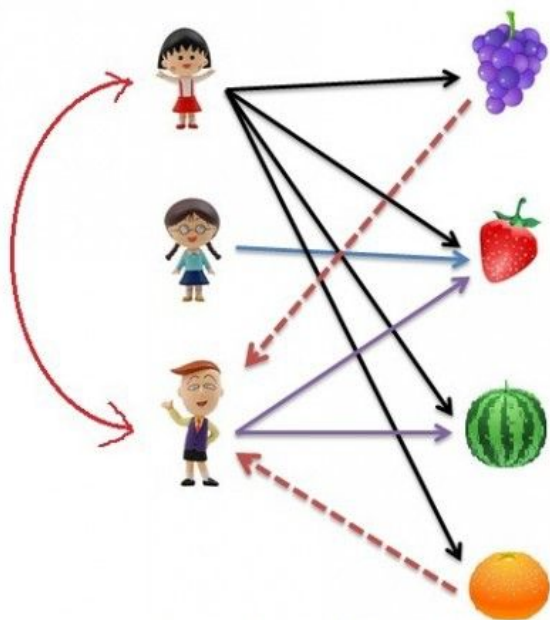
- Como lidar com poucos dados?
 - Similaridade de dois usuários com um rating em comum = 1.
 - Eles são realmente similares?
- **Solução:** penalizar a similaridade pela *confiança*
 - Abordagem simples: multiplicar por $\min(c, 50) / 50$
 - c é o número de ratings em comum
 - $c < 50$: as similaridades são penalizadas por $c / 50$
 - $c > 50$: as similaridades não são penalizadas

5. Opções Adicionais

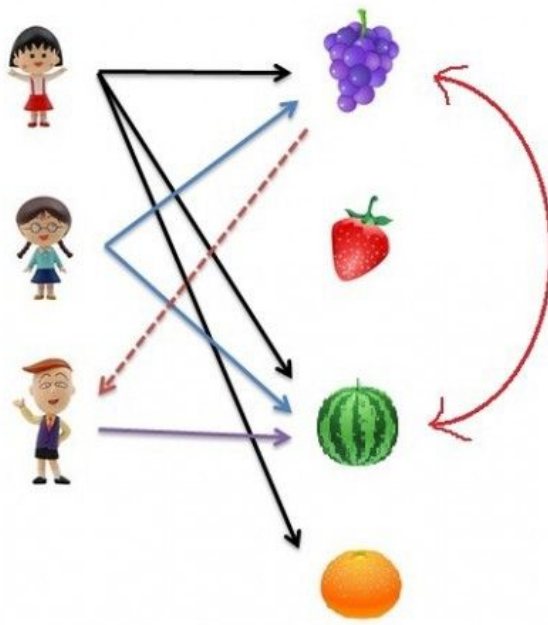
- Pode-se utilizar estratégias de clustering
 - Clusterizar usuários e utilizar esses grupos para predição.
 - Não funciona muito bem na maioria dos casos.
- Pré-computar as similaridades entre todos os usuários
 - É caro computacionalmente.
 - É instável, pois um sistema real se atualiza diariamente.

- O *user-based* sofre com o mal da esparsidade
 - Existem muitos itens ($\sim 10M$) e poucas avaliações (~ 100)
 - É difícil definir bons vizinhos, devido a esse viés
- O *user-based* é pouco eficiente
 - Computar as correlações é $O(m^2n)$
 - É totalmente inviável fazer isso em tempo real (online)
 - Correlações offline são instáveis

Memory-based



User-based filtering



Item-based filtering

"Diga-me o que é popular entre meus amigos."

Item-based CF

O processo é muito similar ao user-based:

1. Normalizar os dados
2. Computar similaridades
3. Selecionar os melhores vizinhos
4. Agregar as informações

1. Normalizar os dados

- Mean-centering (para ratings em uma escala)
 - Subtrair a média das notas do usuário
 - Subtrair a média das notas recebidas pelo item
- Unit-centering (para feedback binário)
 - Dividir pela norma Euclidiana do usuário

2. Computar Similaridades

- Correlação de Pearson
 - Utilizada para valores em uma escala de ratings
- Similaridade de Cosseno
 - Utilizada para uma escala binária de valores
 - Após um processo de normalização

2. Computar Similaridades

Avaliamos os outros itens consumidos por u .

		x	i		y	z		w	
									
			1	3		2			$\text{sim}(\vec{i}, \vec{x}) = 0.82$
	1	2	5		4		1		$\text{sim}(\vec{i}, \vec{y}) = 0.65$
	4			3	5		4		$\text{sim}(\vec{i}, \vec{z}) = 0.07$
u 		2	?		5	4		4	$\text{sim}(\vec{i}, \vec{w}) = 0.00$
		3	4		5		3		

3. Selecionar a Vizinhaça

- Encontrar os k itens mais similares aos...
 - ... itens avaliados por u
 - ... itens visualizados por u
- Qual o melhor valor para k ?
 - Poucos vizinhos geram previsões incoerentes
 - Muitos vizinhos introduzem muito ruído
 - $k = 20$, é um bom ponto de partida!

3. Selecionar a Vizinha

Exemplo com dois vizinhos ($k = 2$)

		x	i		y			
								
			1	3		2		
	1	2	5		4		1	
	4			3	5		4	
u 		2	?		5	4		4
		3	4		5		3	

$$\text{sim}(\vec{i}, \vec{x}) = 0.82$$

$$\text{sim}(\vec{i}, \vec{y}) = 0.65$$














2-NN

4. Agregar informações

Existem algumas abordagens clássicas:

- Min / max / média / mediana dos ratings
- Média ponderada pela similaridade
- Estratégias de agregação supervisionada













4. Agregar informações

		<i>x</i>	<i>i</i>		<i>y</i>				
									
			1	3		2			
	1	2	5		4		1		
	4			3	5		4		
<i>u</i> 		2	?		5	4		4	
		3	4		5		3		

$$\text{sim}(\vec{i}, \vec{x}) = 0.82 \quad r_{ux} = 2$$

$$\text{sim}(\vec{i}, \vec{y}) = 0.65 \quad r_{uy} = 5$$

4. Agregar informações

		<i>x</i>	<i>i</i>		<i>y</i>			
								
			1	3		2		
	1	2	5		4		1	
	4			3	5		4	
<i>u</i> 		2	3.3		5	4		4
		3	4		5		3	

$$\hat{r}_{ui} = \frac{\sum_{j \in N} \text{sim}(\vec{i}, \vec{j}) \times r_{uj}}{\sum_{j \in N} |\text{sim}(\vec{i}, \vec{j})|}$$

$$\hat{r}_{ui} = \frac{0.82 \times 2 + 0.65 \times 5}{0.82 + 0.65}$$

$$\hat{r}_{ui} = 3.33$$

- Item-based é mais **efetivo**
 - O modelo é mais resiliente a esparsidade dos dados
- Item-based é mais **eficiente**
 - A similaridade entre os itens é mais estável que entre usuários
- Item-based é mais **flexível**
 - Pode ser aplicado em:
 - cenários de *profile-based*
 - cenários de *session-based*
 - cenários de *basket-based*

- Modelos memory-based são simples e eficazes estratégias de CF
 - a. User-based
 - b. Item-based