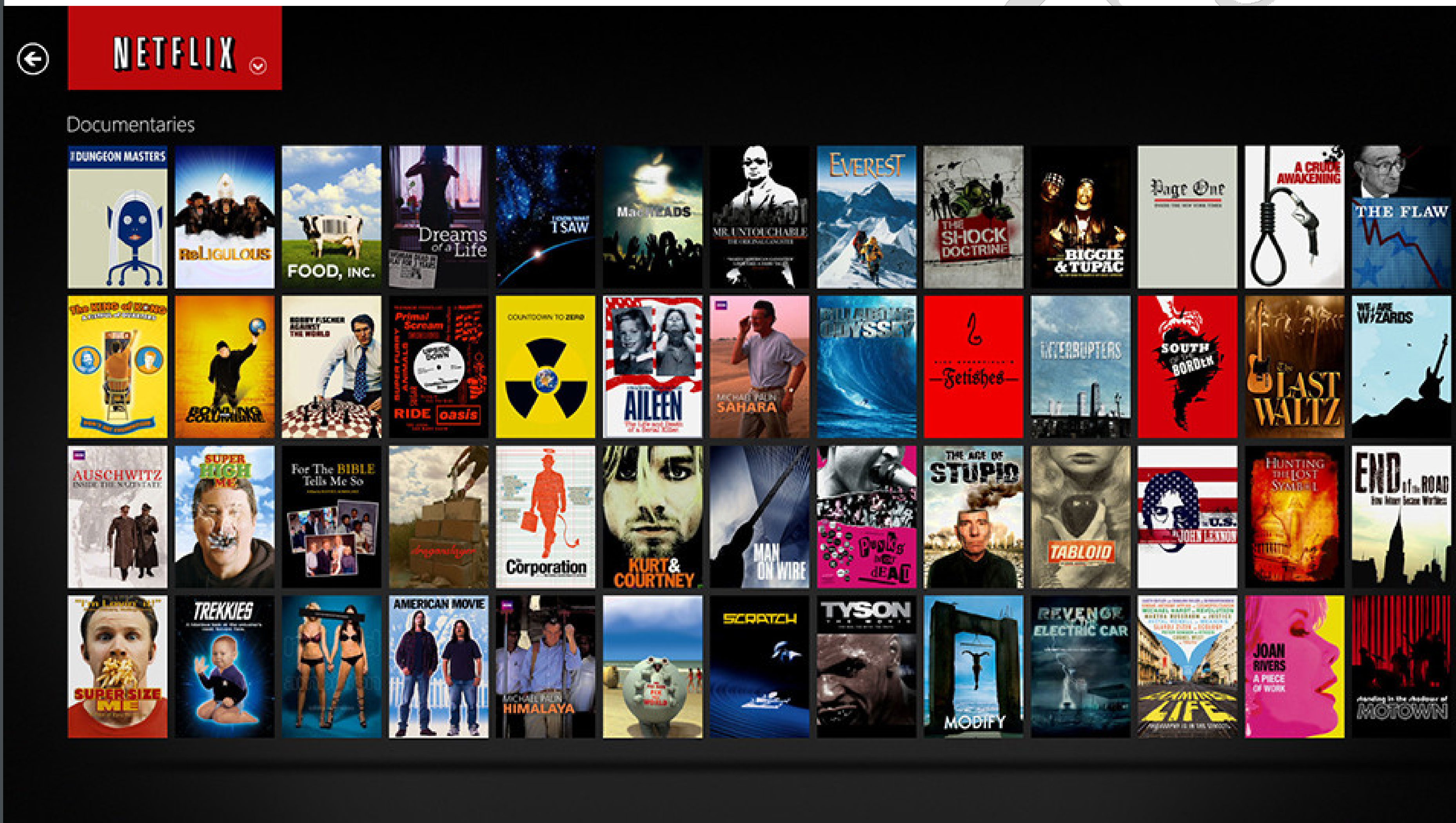
Big Data Analytics & Applications

Bin Li

School of Computer Science

Fudan University

Collaborative Filtering



Memory-based CF

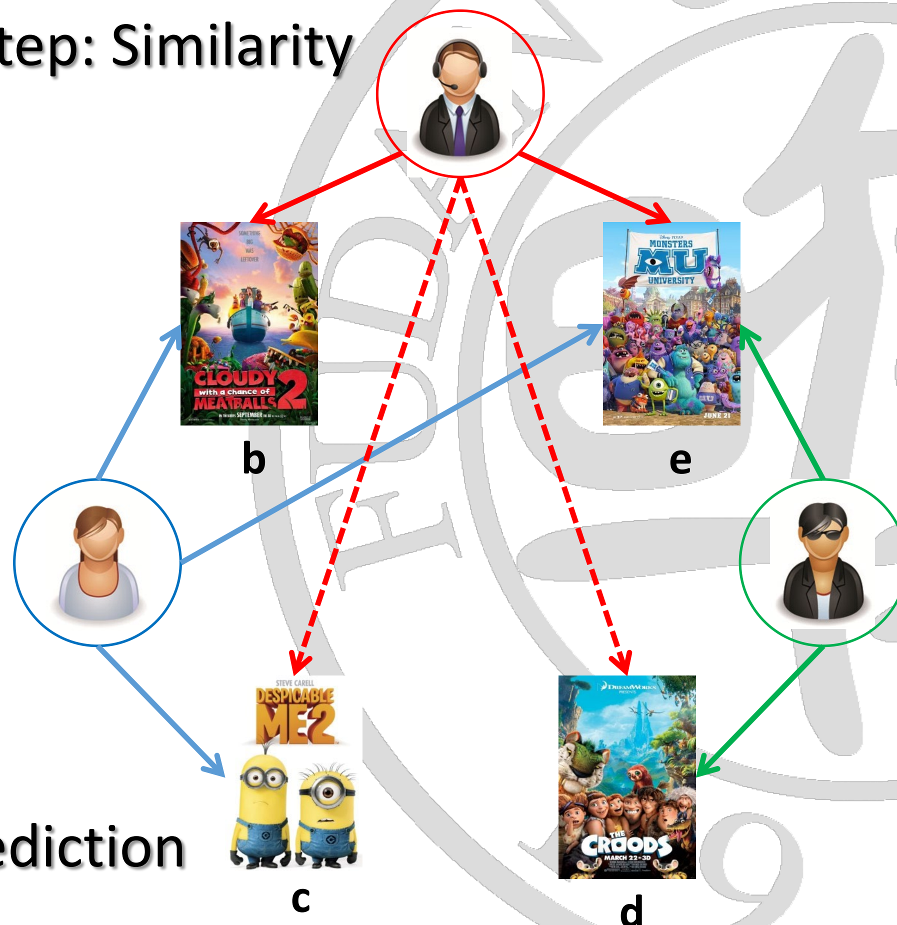
- 1st Step: Collect preference data
  - Represented as a Preference Matrix (bipartite graph)
  - An entry denotes a user’s preference on an item
- 2nd Step: Find neighboring users/items
  - Compute Similarity between users/items
  - Determine neighboring users/items for the target user
- 3rd Step: Recommend unrated items
  - Predict unrated ratings based on neighbors’ ratings
  - Recommend highly ranked items to the target user

User-based CF

1st Step: Data

	a	b	c	d	e	f
A	4					3
B		3	4		3	
C		3			4	
D	4		5			4
E				2	5	

2nd Step: Similarity



3rd Step: Prediction

User-based CF

- Given Preference Matrix **X** and the target user
- Each user is represented as an M-dim vector $\mathbf{x}_u$
  - $\mathbf{x}_u = [x_{u,1}, x_{u,2}, \cdots, x_{u,M}]$ corresponds to the uth row in **X**
  - $x_{u,m}$ denotes the rating user u provides to item m
- User similarity
  - Only calculate on the overlapped items between two users
  - Pearson correlation coefficient and cosine

$$\mathrm{sim}(u,v) = \frac{\sum_{m \in I_u \cap I_v} (x_{u,m} - \bar{x}_u)(x_{v,m} - \bar{x}_v)}{\sqrt{\sum_{m \in I_u \cap I_v} (x_{u,m} - \bar{x}_u)^2} \sqrt{\sum_{m \in I_u \cap I_v} (x_{v,m} - \bar{x}_v)^2}}$$

User-based CF

- User-User Similarity Computation

$$\mathrm{sim}(u,v)=\frac{\sum_{m\in I_u\cap I_v}(x_{u,m}-\bar{x}_u)(x_{v,m}-\bar{x}_v)}{\sqrt{\sum_{m\in I_u\cap I_v}(x_{u,m}-\bar{x}_u)^2}\sqrt{\sum_{m\in I_u\cap I_v}(x_{v,m}-\bar{x}_v)^2}}$$

$$\mathrm{sim}(C,A)=0$$

$$\mathrm{sim}(C,B)=\frac{(3-3.5)(2-3)+(4-3.5)(3-3)}{\sqrt{(3-3.5)^2+(4-3.5)^2}\sqrt{(2-3)^2+(3-3)^2}}=0.7$$

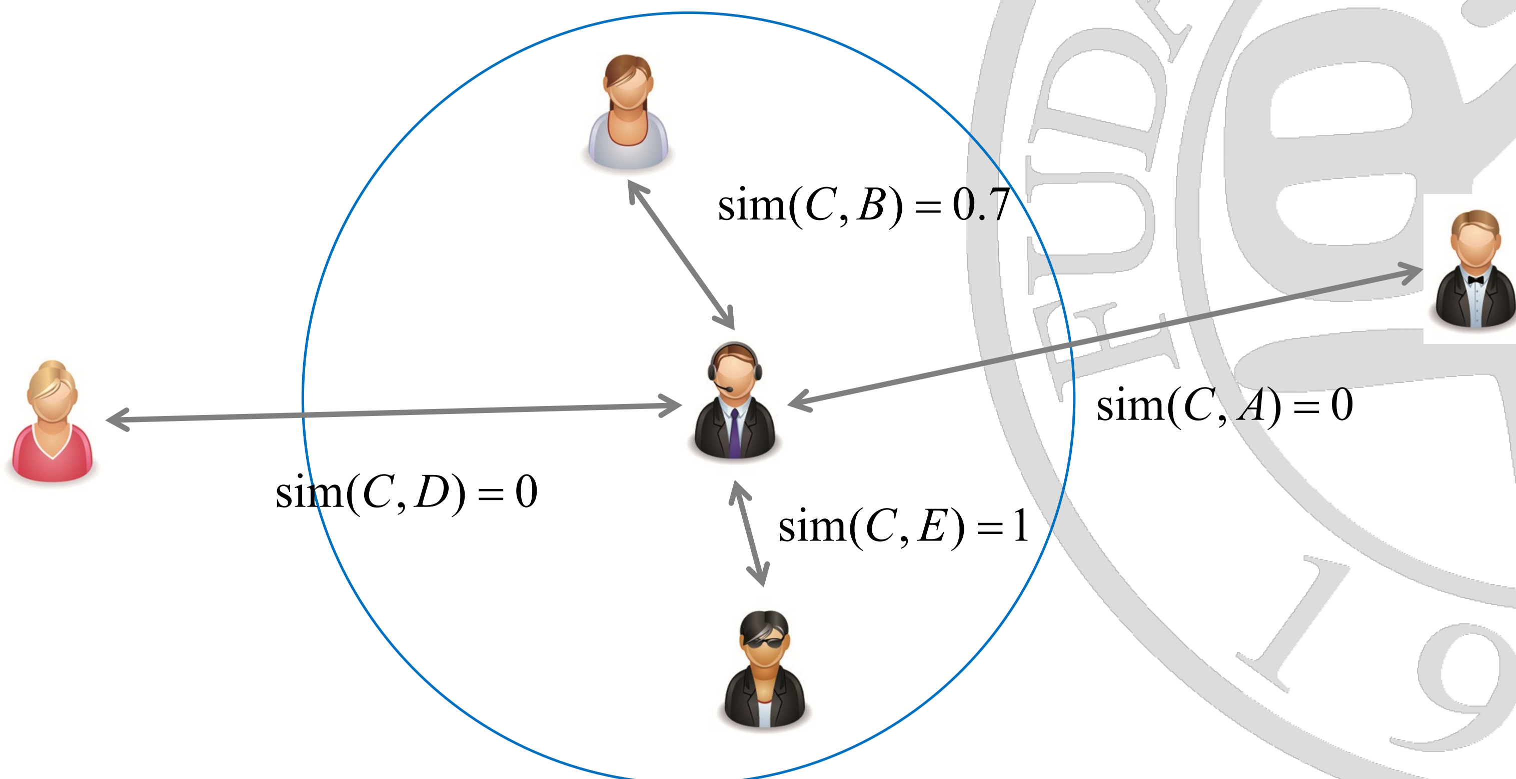
$$\mathrm{sim}(C,D)=0$$

$$\mathrm{sim}(C,E)=\frac{(4-3.5)(5-3.5)}{\sqrt{(4-3.5)^2}\sqrt{(5-3.5)^2}}=1$$

	a	b	c	d	e	f
A	4					3
B		2	4		3	
C		3			4	
D	4		5			4
E				2	5	

User-based CF

- *K*-Nearest Neighbors
  - Top-*K* similar users to the target user



User-based CF

- Rating Prediction

$$\widehat{x}_{u,m} = \bar{x}_u + \frac{\sum_{v \in N_u} \text{sim}(u,v)(x_{v,m} - \bar{x}_v)}{\sum_{v \in N_u} |\,\text{sim}(u,v)\,|}$$

$$\widehat{x}_{C,c} = 3.5 + \frac{0.7(4-3)}{|0.7|} = 4.5$$

$$\widehat{x}_{C,d} = 3.5 + \frac{1(2-3.5)}{|1|} = 2$$

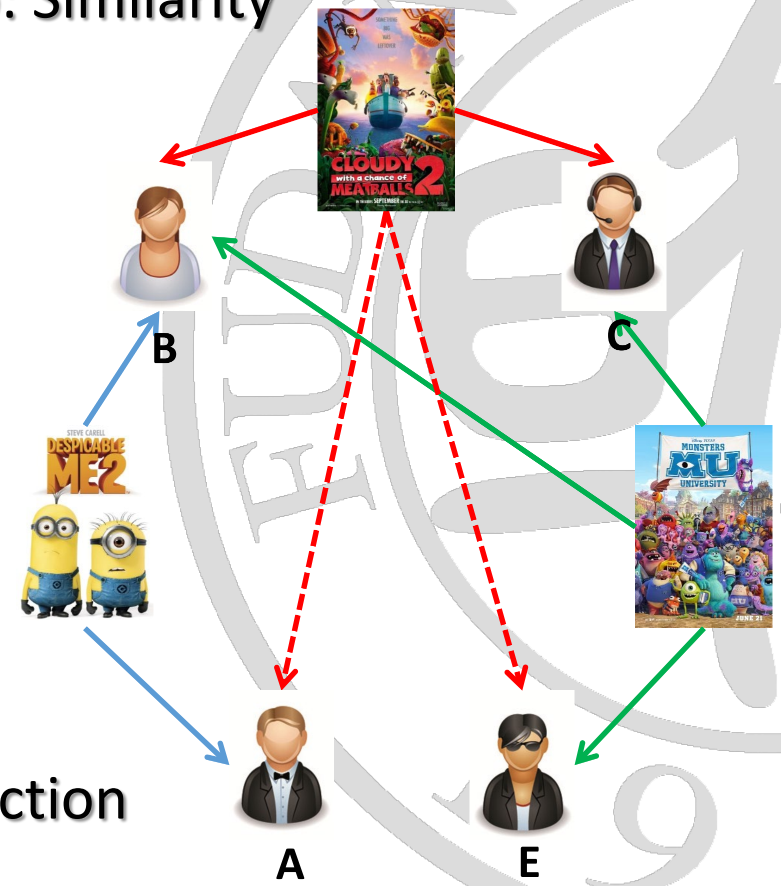
	a	b	c	d	e	f
A	4					3
B		2	4		3	
C		3	4.5	2	4	
D	4		5			4
E				2	5	

Item-based CF

2nd Step: Similarity

1st Step: Data

	a	b	c	d	e	f
A	4					3
B		3	4		3	
C		3			4	
D	4		5			4
E				2	5	



3rd Step: Prediction

Item-based CF

- Given Preference Matrix **X** and the target item
- Each item is represented as an N-dim vector $\mathbf{x}_m$
  - $\mathbf{x}_m = [x_{m,1}, x_{m,2}, \cdots, x_{m,N}]^T$ corresponds to the mth column in **X**
  - $x_{m,u}$ denotes the rating user u provides to item m
- Item similarity
  - Only calculate on the overlapped users between two items
  - Cosine and Pearson correlation coefficient

$$\mathrm{sim}(m,m') = \frac{\sum_{u \in U_m \cap U_{m'}} x_{m,u} x_{m',u}}{\sqrt{\sum_{u \in U_m \cap U_{m'}} x_{m,u}^2} \sqrt{\sum_{u \in U_m \cap U_{m'}} x_{m',u}^2}}$$

Item-based CF

- Item-Item Similarity Computation

$$\mathrm{sim}(m,m') = \frac{\sum_{u \in U_m \cap U_{m'}} x_{m,u} x_{m',u}}{\sqrt{\sum_{u \in U_m \cap U_{m'}} x_{m,u}^2} \sqrt{\sum_{u \in U_m \cap U_{m'}} x_{m',u}^2}}$$

$$\mathrm{sim}(b,a) = 0$$

$$\mathrm{sim}(b,c) = \frac{3 \times 4}{\sqrt{3^2}\sqrt{4^2}} = 1$$

$$\mathrm{sim}(b,d) = 0$$

$$\mathrm{sim}(b,e) = \frac{3 \times 3 + 3 \times 4}{\sqrt{3^2 + 3^2}\sqrt{3^2 + 4^2}} \approx 1$$

$$\mathrm{sim}(b,f) = 0$$

	a	b	c	d	e	f
A	4					3
B		3	4		3	
C		3			4	
D	4		5			4
E				2	5	

Item-based CF

- Rating Prediction

$$\widehat{x}_{m,u} = \frac{\sum_{m' \in I_m} \mathrm{sim}(m,m') x_{m',u}}{\sum_{m' \in I_m} |\mathrm{sim}(m,m')|}$$

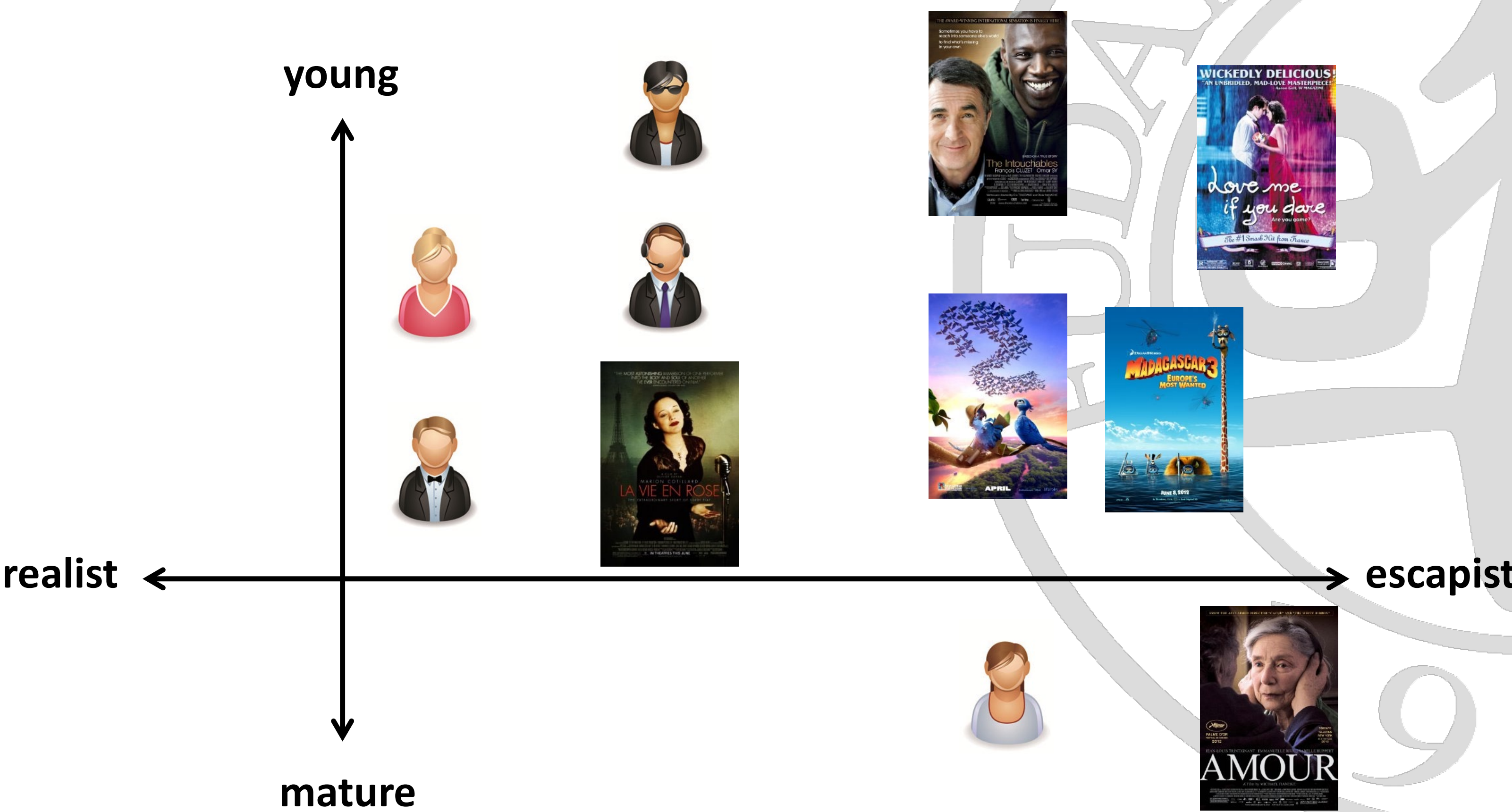
$$\widehat{x}_{b,D} = \frac{1 \times 5}{|1|} = 5$$

$$\widehat{x}_{b,E} = \frac{1 \times 5}{|1|} = 5$$

	a	b	c	d	e	f
A	4					3
B		3	4		3	
C		3			4	
D	4	5	5			4
E		5		2	5	

Model-based CF

- Latent variable view – matrix factorization approach



Model-based CF

- Matrix Factorization

realist/ escapist	young/ mature
-2.1	-0.4
-1.9	-0.6
-2.0	-0.1
-2.1	-0.1
-2.0	1.2

$\times$

-1.9	-1.7	-2.0	-1.7	-1.9	-1.8	realist/ escapist
-0.1	0.2	-0.3	-1.0	1.0	0.1	young/ mature

Assume two latent variables (features)

Rank-2 matrix factorization : $\hat{\mathbf{X}} = \mathbf{F}\mathbf{G}^{\mathrm{T}} \in R^{5\times 6}$

User feature matrix : $\mathbf{F} \in R^{5\times 2}$

Item feature matrix : $\mathbf{G} \in R^{6\times 2}$

Model-based CF

- Preference (Rating) Matrix Reconstruction
  - Predict missing ratings in the rating matrix

$$
\begin{bmatrix}
4 & & 5 & & & 3 \\
 & 3 & 4 & & 3 & \\
3.8 & 3 & & & 4 & \\
4 & & & 3.7 & & 4 \\
 & & & 2 & 5 &
\end{bmatrix}
\sim
\begin{bmatrix}
-2.1 & -0.4 \\
-1.9 & -0.6 \\
-2.0 & -0.1 \\
-2.1 & -0.1 \\
-2.0 & 1.2
\end{bmatrix}
\times
\begin{bmatrix}
-1.9 & -1.7 & -2.0 & -1.7 & -1.9 & -1.8 \\
-0.1 & 0.2 & -0.3 & -1.0 & 1.0 & 0.1
\end{bmatrix}
$$

- Remaining problems
  - Why we can assume rank-K matrices (K latent variables)?
  - How to compute rank-K matrices (user/item feature matrices)?

Model-based CF

- Why K latent variables?
  - We don’t know exact number of features in advance
  - We can assume there are indeed L (>>K) features, so

$$\text{User feature matrix}: \mathbf{F}_0 \in R^{N\times L}$$

$$\text{Item feature matrix}: \mathbf{G}_0 \in R^{M\times L}$$

  - We do a linear projection to $\mathbf{F}_0$ and $\mathbf{G}_0$ (feature reduction)

$$\text{Projection matrix}: \mathbf{A} \in R^{L\times L}, \mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{I}$$

$$\text{User feature matrix}: \mathbf{F} = \mathbf{F}_0\mathbf{A}_{1:K} \in R^{N\times K}$$

$$\text{Item feature matrix}: \mathbf{G} = \mathbf{G}_0\mathbf{A}_{1:K} \in R^{M\times K}$$

  - Now we can directly compute **F** and **G** (without noisy features)

Model-based CF

- How to get rank-K feature matrices
  - Low-rank matrix factorization problem
  - Minimize the reconstruction and the observed preference matrix
- Regularized risk minimization
  - Many ML methods can be applied

Given the preference matrix $\mathbf{X} \in R^{N \times M}$ and rank K

$$\min_{\left\{\mathbf{F} \in R^{N \times K}, \mathbf{G} \in R^{M \times K}\right\}} \left\| \left(\mathbf{X} - \mathbf{F}\mathbf{G}^{\mathrm{T}} \right) \circ \mathbf{W} \right\|_F^2 + \lambda \left(\|\mathbf{F}\|_F^2 + \|\mathbf{G}\|_F^2 \right)$$

where $\mathbf{W} \in \{0,1\}^{N \times M}$ indicates observed entries in $\mathbf{X}$

Model-based CF

- Probabilistic Matrix Factorization (PMF)
  - Assume user/item features and ratings are generated from Gaussians

$$p(\mathbf{X} \mid \mathbf{F}, \mathbf{G}) = \prod_{w_{u,m}=1} p\left(x_{u,m} \mid \mathbf{f}_u^{\mathrm{T}} \mathbf{g}_m, \sigma_R^2\right)$$

$$p(\mathbf{F} \mid \mathbf{0}) = \prod_{u=1}^{N} p\left(\mathbf{f}_u \mid \mathbf{0}, \sigma_F^2\right)$$

$$p(\mathbf{G} \mid \mathbf{0}) = \prod_{m=1}^{M} p\left(\mathbf{g}_m \mid \mathbf{0}, \sigma_G^2\right)$$

  - Probabilistic interpretation of the optimization problem

$$\max_{\{\mathbf{F},\mathbf{G}\}} \ln\left[p(\mathbf{X} \mid \mathbf{F}, \mathbf{G}) p(\mathbf{F} \mid \mathbf{0}) p(\mathbf{G} \mid \mathbf{0})\right]$$

$$\Rightarrow \min_{\{\mathbf{F},\mathbf{G}\}} \sum_{w_{u,m}=1} \left(x_{u,m} - \mathbf{f}_u^{\mathrm{T}} \mathbf{g}_m\right)^2 + c_1 \sum_{u=1}^{N} \left\|\mathbf{f}_u\right\|^2 + c_2 \sum_{m=1}^{M} \left\|\mathbf{g}_m\right\|^2$$

$$\Rightarrow \min_{\{\mathbf{F},\mathbf{G}\}} \left\|\left(\mathbf{X} - \mathbf{F}\mathbf{G}^{\mathrm{T}}\right) \circ \mathbf{W}\right\|_F^2 + c_1 \left\|\mathbf{F}\right\|_F^2 + c_2 \left\|\mathbf{G}\right\|_F^2$$

[1] Salakhutdinov & Mnih: Probabilistic Matrix Factorization, NIPS 2008.

Model-based CF

- Singular Value Decomposition (SVD)
  - Most straightforward way to matrix factorization
  - But SVD is not defined for missing entries
  - Use average rating to stuff missing entries
  - Inaccurate for sparse matrices (tries to fit too many stuff entries)

4	3.7	5	3.7	3.7	3
3.7	3	4	3.7	3	3.7
3.7	3	3.7	3.7	4	3.7
4	3.7	3.7	3.7	3.7	4
3.7	3.7	3.7	2	5	3.7

$$\text{Filled matrix}: \widetilde{\mathbf{X}} \in R^{N\times M}$$

$$\text{User feature matrix}: \mathbf{F} \in R^{N\times K}$$

$$\text{Item feature matrix}: \mathbf{G} \in R^{M\times K}$$

$$\text{SVD}: \widetilde{\mathbf{X}} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}} = \left(\mathbf{U}\sqrt{\mathbf{S}}\right)\left(\sqrt{\mathbf{S}}\mathbf{V}\right)^{\mathrm{T}} = \mathbf{F}\mathbf{G}^{\mathrm{T}}$$

Model-based CF

- Alternative Least Squares (ALS)
  - Optimize **F** assuming **G** is known
  - Optimize **G** assuming **F** is known
  - Each step is a standard least square problem
  - Converge to a local minimum over alternative iterations

$$\min_{\{\mathbf{F}\in R^{N\times K},\mathbf{G}\in R^{M\times K}\}} \left\|\left(\mathbf{X}-\mathbf{F}\mathbf{G}^{\mathrm{T}}\right)\circ\mathbf{W}\right\|_F^2 + \lambda\left(\|\mathbf{F}\|_F^2 + \|\mathbf{G}\|_F^2\right)$$

$$\Rightarrow \begin{cases} \min_{\mathbf{F}\in R^{N\times K}} \left\|(\mathbf{X}-\mathbf{F}\widehat{\mathbf{G}}^{\mathrm{T}})\circ\mathbf{W}\right\|_F^2 + \lambda\|\mathbf{F}\|_F^2 \\ \min_{\mathbf{G}\in R^{M\times K}} \left\|(\mathbf{X}-\hat{\mathbf{F}}\mathbf{G}^{\mathrm{T}})\circ\mathbf{W}\right\|_F^2 + \lambda\|\mathbf{G}\|_F^2 \end{cases}$$

Model-based CF

- Alternative Least Squares (ALS)

$$\min_{\mathbf{F}\in R^{N\times K}} \left\| \left(\mathbf{X}-\mathbf{F}\hat{\mathbf{G}}^{\mathrm{T}}\right)\circ \mathbf{W} \right\|_F^2 + \lambda \left\|\mathbf{F}\right\|_F^2$$

$$\Rightarrow \min_{\mathbf{f}_u \in R^K} \sum_{w_{u,m}=1} \left(x_{u,m} - \mathbf{f}_u^{\mathrm{T}}\hat{\mathbf{g}}_m\right)^2 + \lambda \left\|\mathbf{f}_u\right\|^2, \text{ for } u \in U$$

$$\Rightarrow \mathbf{f}_u \leftarrow \left(\lambda + \sum_{w_{u,m}=1} \hat{\mathbf{g}}_m \hat{\mathbf{g}}_m^{\mathrm{T}}\right)^{-1} \sum_{w_{u,m}=1} \hat{\mathbf{g}}_m x_{u,m}$$

$$\min_{\mathbf{G}\in R^{M\times K}} \left\| \left(\mathbf{X}-\hat{\mathbf{F}}\mathbf{G}^{\mathrm{T}}\right)\circ \mathbf{W} \right\|_F^2 + \lambda \left\|\mathbf{G}\right\|_F^2$$

$$\Rightarrow \min_{\mathbf{g}_m \in R^K} \sum_{w_{u,m}=1} \left(x_{u,m} - \hat{\mathbf{f}}_u^{\mathrm{T}}\mathbf{g}_m\right)^2 + \lambda \left\|\mathbf{g}_m\right\|^2, \text{ for } m \in I$$

$$\Rightarrow \mathbf{g}_m \leftarrow \left(\lambda + \sum_{w_{u,m}=1} \hat{\mathbf{f}}_u \hat{\mathbf{f}}_u^{\mathrm{T}}\right)^{-1} \sum_{w_{u,m}=1} \hat{\mathbf{f}}_u x_{u,m}$$

Model-based CF

- Stochastic Gradient Descent (SGD)
  - Minimize an objective in the form of a sum of differentiable functions
  - All ratings in the rating matrix are shuffled and fed in sequentially
  - Each time a user/item feature vector is optimized on a single rating

$$\min_{\{\mathbf{F}\in R^{N\times K},\mathbf{G}\in R^{M\times K}\}}\left\|\left(\mathbf{X}-\mathbf{F}\mathbf{G}^{\mathrm{T}}\right)\circ\mathbf{W}\right\|_F^2+\lambda\left(\|\mathbf{F}\|_F^2+\|\mathbf{G}\|_F^2\right)$$

$$\Rightarrow\min_{\{\mathbf{F},\mathbf{G}\}}\sum_{w_{u,m}=1}\left(x_{u,m}-\mathbf{f}_u^{\mathrm{T}}\mathbf{g}_m\right)^2+\lambda\left(\sum_{u=1}^{N}\|\mathbf{f}_u\|^2+\sum_{m=1}^{M}\|\mathbf{g}_m\|^2\right)$$

$$\Rightarrow\begin{cases}\mathbf{f}_u\leftarrow(1-\alpha\lambda)\mathbf{f}_u-\alpha\mathbf{g}_m\left(x_{u,m}-\mathbf{f}_u^{\mathrm{T}}\mathbf{g}_m\right)\\ \mathbf{g}_m\leftarrow(1-\alpha\lambda)\mathbf{g}_m-\alpha\mathbf{f}_u\left(x_{u,m}-\mathbf{f}_u^{\mathrm{T}}\mathbf{g}_m\right)\end{cases},\ \text{for all}\ \{x_{u,m}\}$$

Model-based CF

- SVD++[1]: Netflix Winner's Method
  - An improvement of SVD
  - Consider user bias b_u and item bias b_m

$$\min_{\{\mathbf{F}\in R^{N\times K},\mathbf{G}\in R^{M\times K}\}}\left\|\left(\mathbf{X}-\mathbf{F}\mathbf{G}^{\mathrm{T}}\right)\circ\mathbf{W}\right\|_F^2+\lambda\left(\|\mathbf{F}\|_F^2+\|\mathbf{G}\|_F^2\right)$$

$$\Rightarrow \min_{\{\mathbf{F},\mathbf{G}\}}\sum_{w_{u,m}=1}\left(x_{u,m}-\left(\mu+b_u+b_m+\mathbf{f}_u^{\mathrm{T}}\mathbf{g}_m\right)\right)^2$$

$$+\lambda\left(\sum_{u=1}^{N}\|\mathbf{f}_u\|^2+\sum_{m=1}^{M}\|\mathbf{g}_m\|^2+\sum_{u=1}^{N}\|b_u\|^2+\sum_{m=1}^{M}\|b_m\|^2\right)$$

[1] Koren: Factorization meets the neighborhood: a multifaceted collaborative filtering model, KDD 2008

Model-based CF

- SVD++: Netflix Winner's Method
  - Stochastic Gradient Descent solution

$$\min_{\{\mathbf{F},\mathbf{G}\}} \sum_{w_{u,m}=1} \left(x_{u,m} - \left(\mu + b_u + b_m + \mathbf{f}_u^{\mathrm{T}} \mathbf{g}_m\right)\right)^2$$

$$+ \lambda \left(\sum_{u=1}^{N} \left\|\mathbf{f}_u\right\|^2 + \sum_{m=1}^{M} \left\|\mathbf{g}_m\right\|^2 + \sum_{u=1}^{N} \left\|b_u\right\|^2 + \sum_{m=1}^{M} \left\|b_m\right\|^2 \right)$$

$$\Rightarrow \begin{cases} \mathbf{f}_u \leftarrow (1-\alpha\lambda)\mathbf{f}_u - \alpha \mathbf{g}_m \delta_{u,m} \\ \mathbf{g}_m \leftarrow (1-\alpha\lambda)\mathbf{g}_m - \alpha \mathbf{f}_u \delta_{u,m} \\ b_u \leftarrow (1-\alpha\lambda) b_u - \alpha \delta_{u,m} \\ b_m \leftarrow (1-\alpha\lambda) b_m - \alpha \delta_{u,m} \end{cases}, \text{for all } \{x_{u,m}\}$$

$$\text{where } \delta_{u,m} = x_{u,m} - \left(\mu + b_u + b_m + \mathbf{f}_u^{\mathrm{T}} \mathbf{g}_m\right)$$

Project: Collaborative Filtering

- Dataset:
  - Public available datasets for collaborative filtering (e.g., https://movielens.org/)
  - Or user rating data collected by yourself
- Method:
  - Use User-based CF or Probabilistic Matrix Factorization for collaborative filtering
- Experiments:
  - Obtain the rating prediction results for evaluate the performance
  - And discuss the limitations of the method you used based on the observations from the results

Thanks

Email: libin@fudan.edu.cn