CS247: ADVANCED DATA MINING

Recommender Systems II

Instructor: Yizhou Sun

yzsun@cs.ucla.edu

May 25, 2020

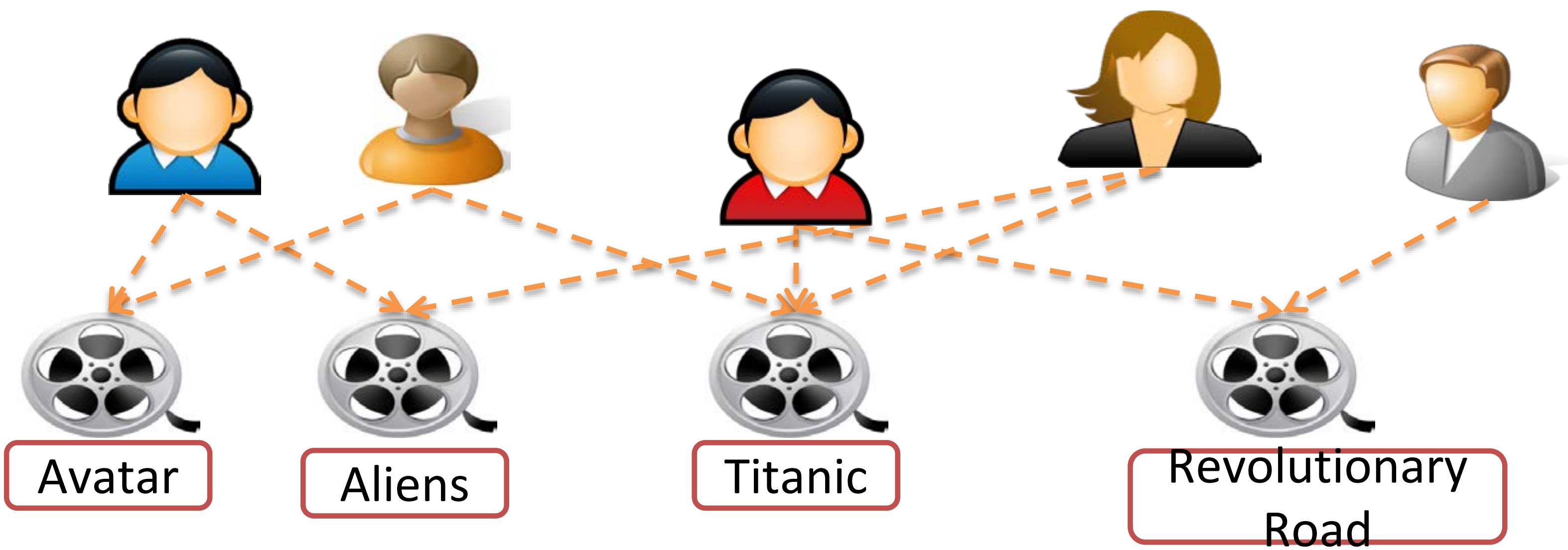
Methods to Learn

	Vector Data	Text Data	Graph & Network	Recommender Systems
Classification	Naïve Bayes; Logistic Regression; NN		Label Propagation	
Clustering	K-means; kernel k-means; Mixture Models	PLSA; LDA	Spectral Clustering	Matrix Factorization
Prediction	NN			Collaborative Filtering; Factorization machine; Hybrid CF; Recommendation with graph regularization
Ranking			PageRank	
Similarity Search			P-PageRank	
Representation Learning		Word embedding	Network Embedding	Deep collaborative learning

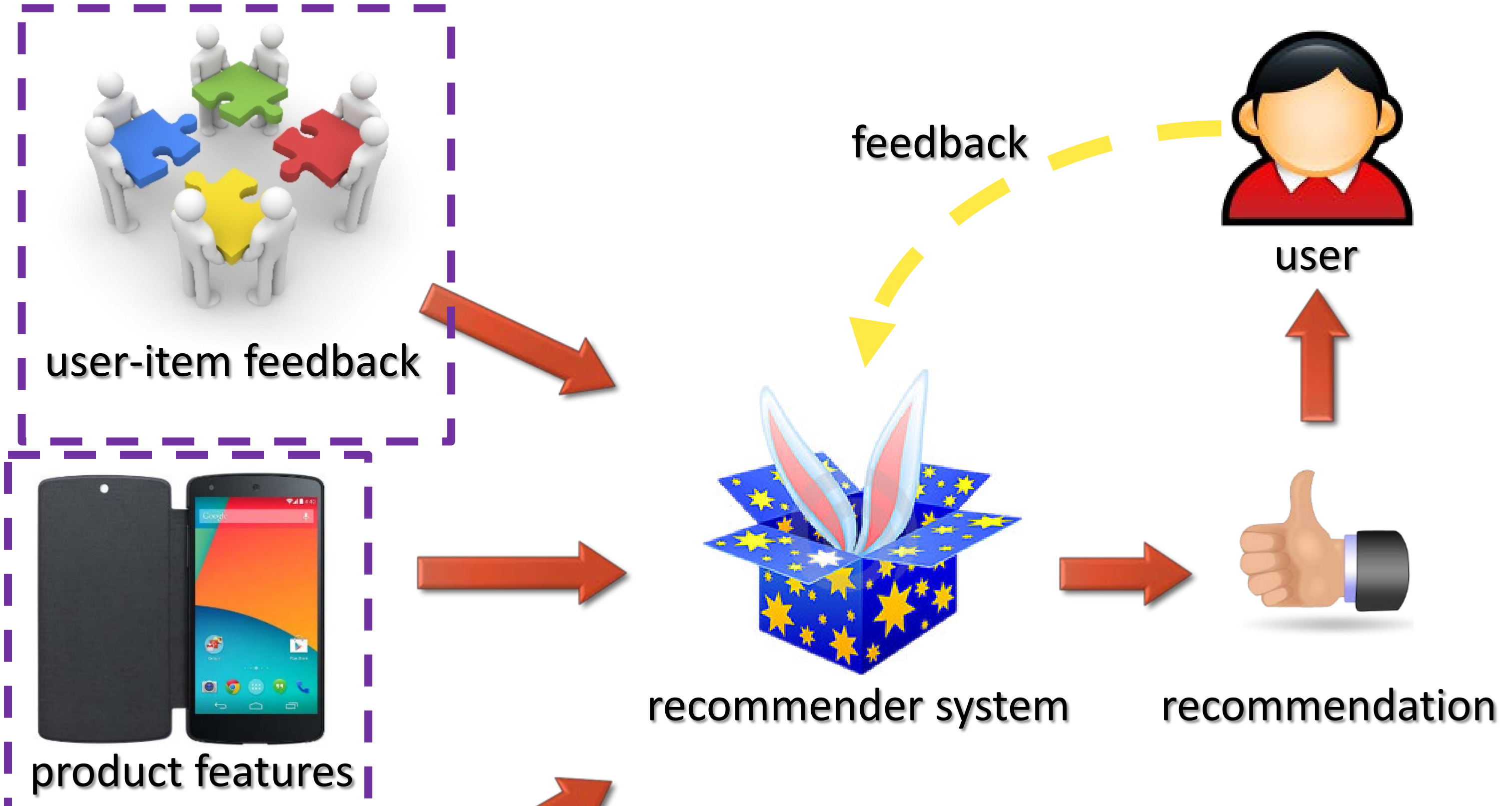
Recommender Systems

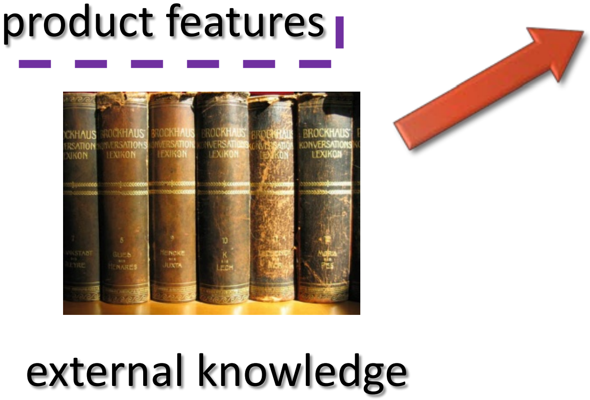
- Recommendation via Information Network Analysis
- Hybrid Collaborative Filtering with Information Networks
- Graph Regularization for Recommendation
- Summary

Traditional View of Recommendation



Recommendation Paradigm





Probabilistic Model (Hofmann SIGIR'03)

An Example of Traditional Method: Matrix Factorization

R: Rating Matrix

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	5	2		3		4		
u_2	4	3			5			
u_3	4		2				2	4
u_4								
u_5	5	1	2		4	3		
u_6	4	3		2	4		3	5

$\hat{R}$: Estimated Rating Matrix

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	5	2	**2.5**	3	**4.8**	4	**2.2**	**4.8**
u_2	4	3	**2.4**	**2.9**	5	**4.1**	**2.6**	**4.7**
u_3	4	**1.7**	2	**3.2**	**3.9**	**3.0**	2	4
u_4	**4.8**	**2.1**	**2.7**	**2.6**	**4.7**	**3.8**	**2.4**	**4.9**
u_5	5	1	2	**3.4**	4	3	**1.5**	**4.6**
u_6	4	3	**2.9**	2	4	**3.4**	3	5

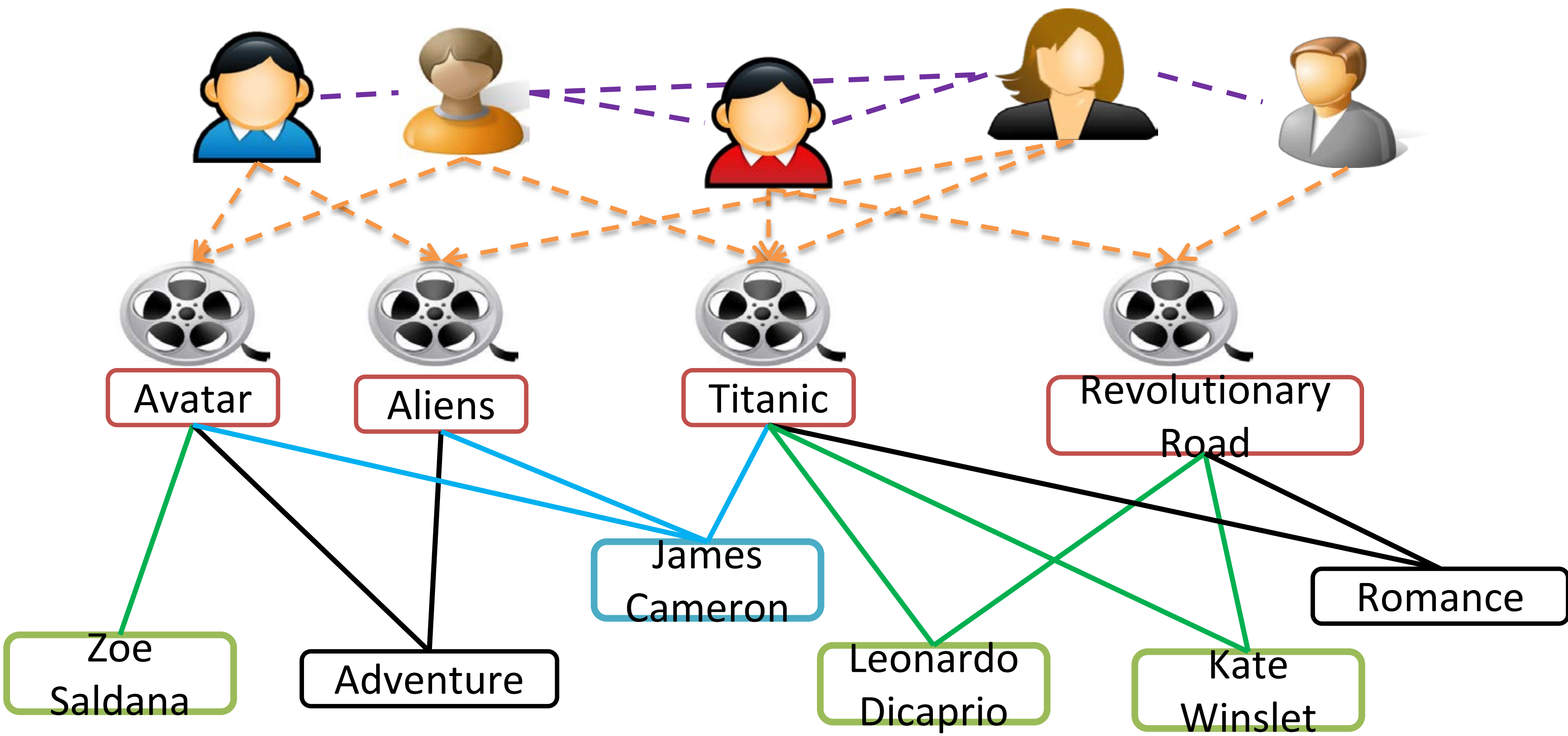
$$U = \begin{bmatrix} 1.55 & 1.22 & 0.37 & 0.81 & 0.62 & -0.01 \\ 0.36 & 0.91 & 1.21 & 0.39 & 1.10 & 0.25 \\ 0.59 & 0.20 & 0.14 & 0.83 & 0.27 & 1.51 \\ 0.39 & 1.33 & -0.43 & 0.70 & -0.90 & 0.68 \\ 1.05 & 0.11 & 0.17 & 1.18 & 1.81 & 0.40 \end{bmatrix} \quad V = \begin{bmatrix} 1.00 & -0.05 & -0.24 & 0.26 & 1.28 & 0.54 & -0.31 & 0.52 \\ 0.19 & -0.86 & -0.72 & 0.05 & 0.68 & 0.02 & -0.61 & 0.70 \\ 0.49 & 0.09 & -0.05 & -0.62 & 0.12 & 0.08 & 0.02 & 1.60 \\ -0.40 & 0.70 & 0.27 & -0.27 & 0.99 & 0.44 & 0.39 & 0.74 \\ 1.49 & -1.00 & 0.06 & 0.05 & 0.23 & 0.01 & -0.36 & 0.80 \end{bmatrix}$$

Challenges

- How to address the data sparsity and cold start issues?

- How to leverage different sources of information?

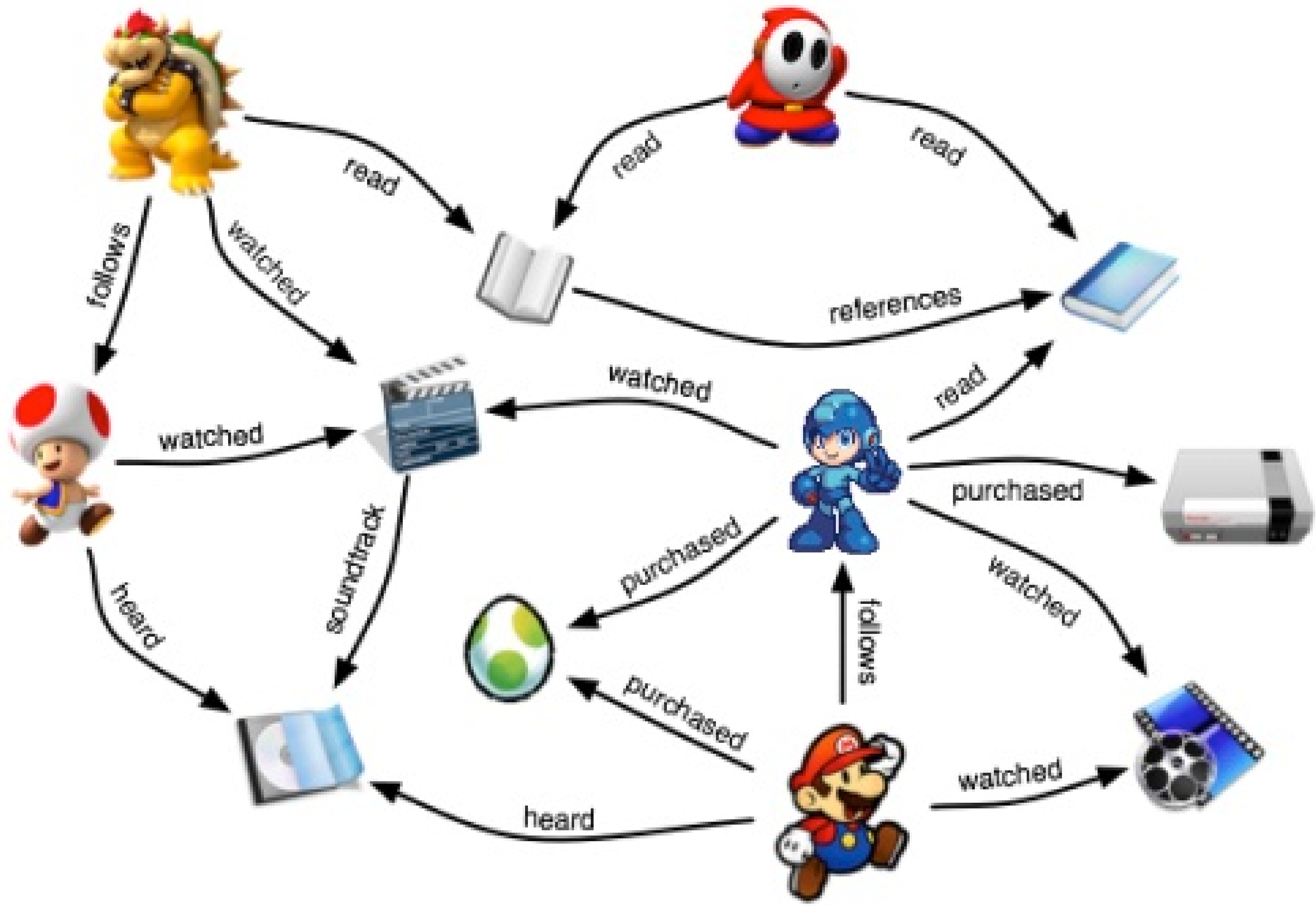
Solution: A Heterogeneous Information Network View of Recommendation



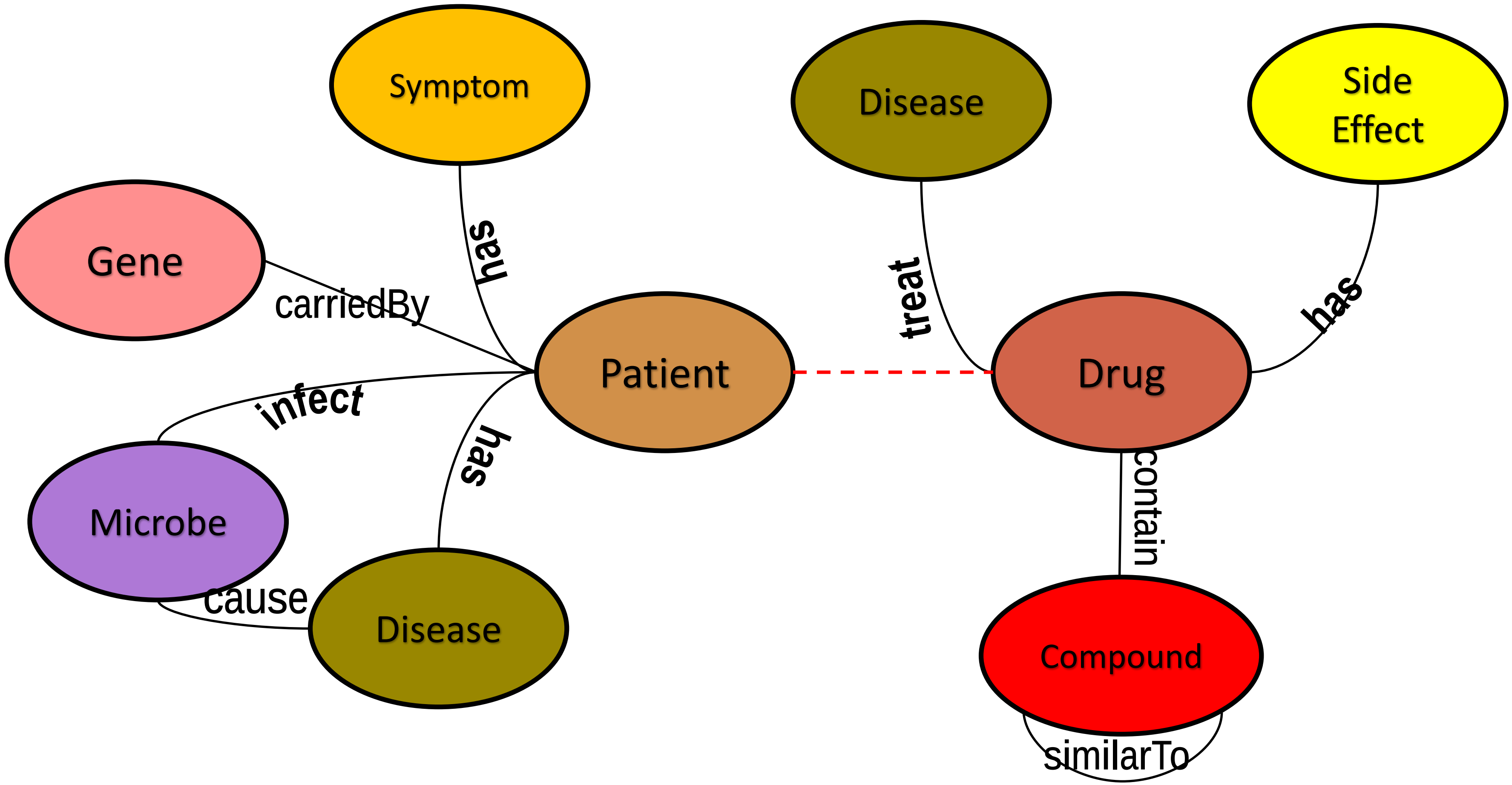
What Are Information Networks?

- A network where each node represents an entity (e.g., user in a social network) and each link (e.g., friendship) a relationship between entities.
  - Nodes/links may have attributes, labels, and weights.
  - Links may carry rich semantic information.

We are living in a connected world!



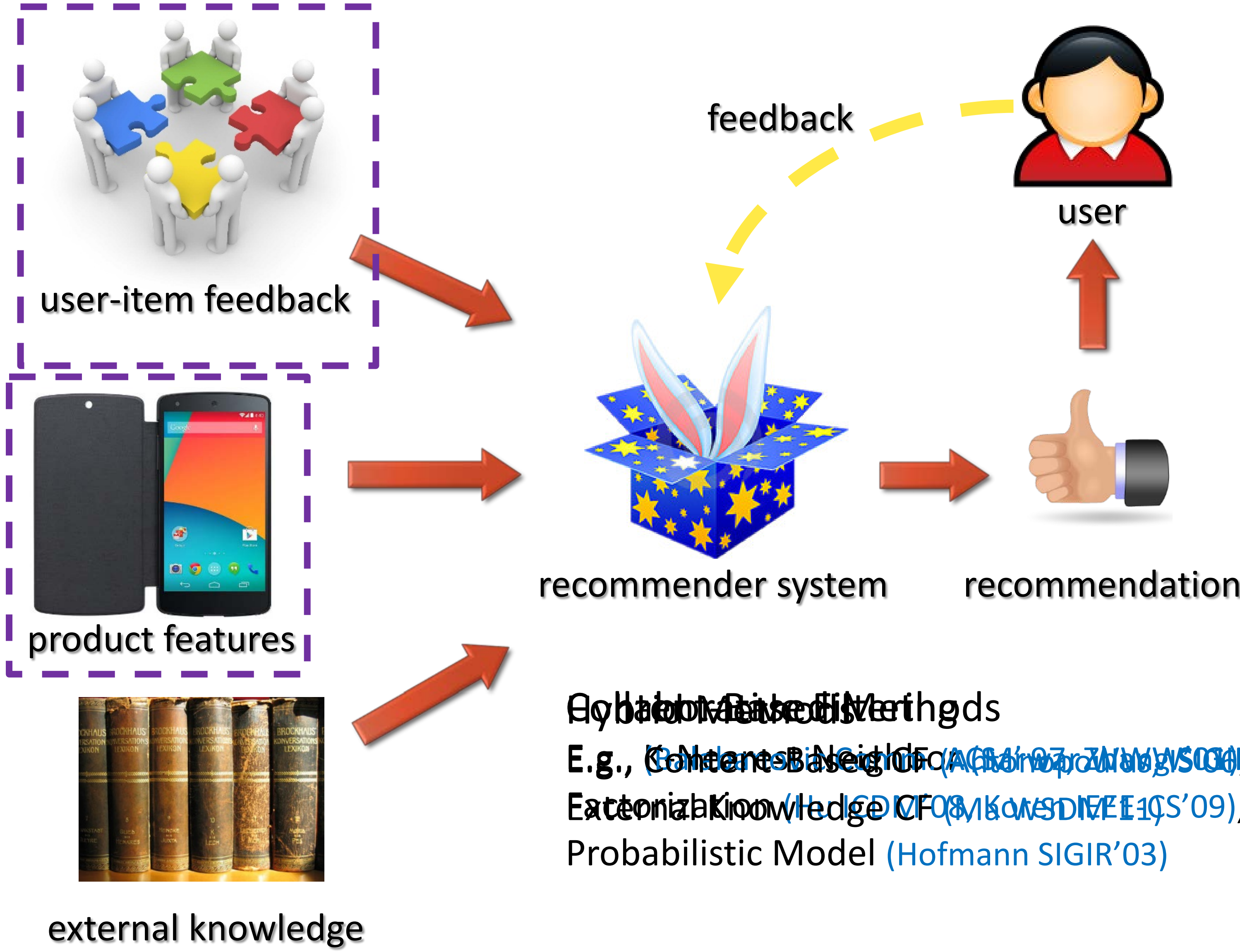
Even in Biomedical Domain



Recommender Systems

- Recommendation via Information Network Analysis
- Hybrid Collaborative Filtering with Information Networks
- Graph Regularization for Recommendation
- Summary

Recommendation Paradigm



Collaborative Filtering

E.g., K-Nearest Neighbor, Matrix Factorization (Hu ICDM'08, Koren IEEE-CS'09),

Probabilistic Model (Hofmann SIGIR'03)

Problem Definition

implicit user feedback

feedback

user

recommender system

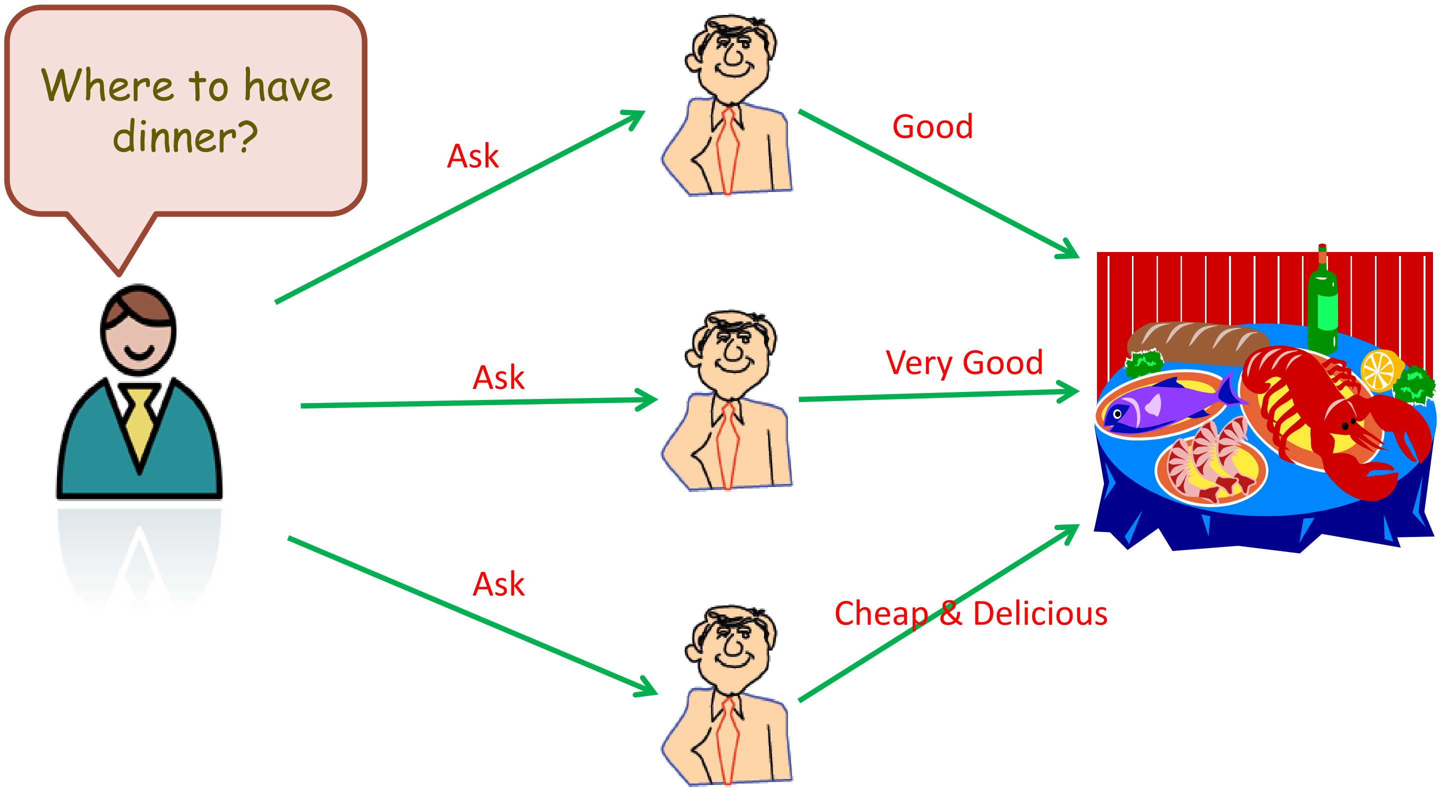
recommendation

information network

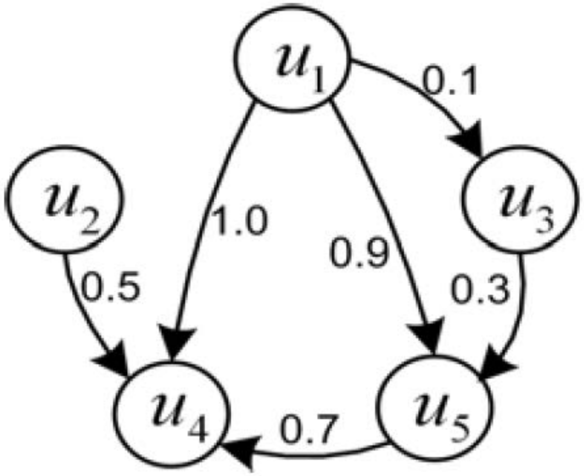
hybrid collaborative filtering with information networks

Recommend with Trust and Distrust Relationships [Ma et al., RecSys'09]

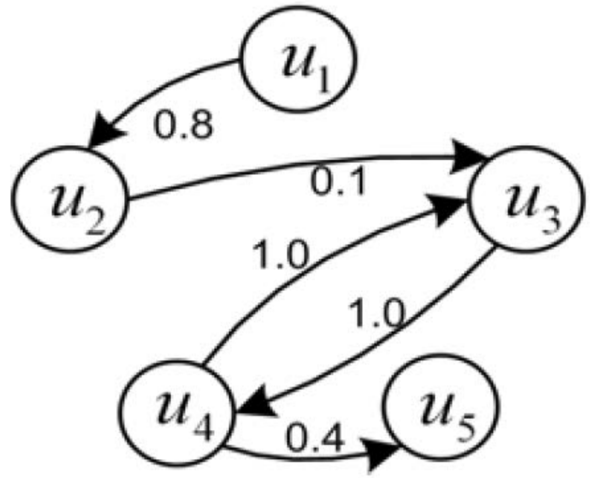
- Users can be easily influenced by the friends they trust, and prefer their friends' recommendations.



Trust and Distrust Graph



S^T: Trust Graph



S^D: Distrust Graph

	v_1	v_2	v_3	v_4	v_5
u_1	5		3		5
u_2				1	
u_3		4			
u_4	3		4	2	
u_5		5			4

R: User Item Rating Matrix

Recommendation with Trust and Distrust Relationships

$$\min_{U,V} \mathcal{L}_{\mathcal{T}}(R, S^{\mathcal{T}}, U, V) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(R_{ij} - g(U_i^T V_j))^2 + \frac{\alpha}{2}\sum_{i=1}^{m}\sum_{t\in\mathcal{T}+(i)} (S_{it}^{\mathcal{T}} \|U_i - U_t\|_F^2) + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2. \quad (7)$$

S^T: Trust Graph

$$\min_{U,V} \mathcal{L}_{\mathcal{D}}(R, S^{\mathcal{D}}, U, V) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(R_{ij} - g(U_i^T V_j))^2 + \frac{\beta}{2}\sum_{i=1}^{m}\sum_{d\in\mathcal{D}+(i)} (-S_{id}^{\mathcal{D}} \|U_i - U_d\|_F^2) + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2. \quad (3)$$

S^D: Distrust Graph

Results

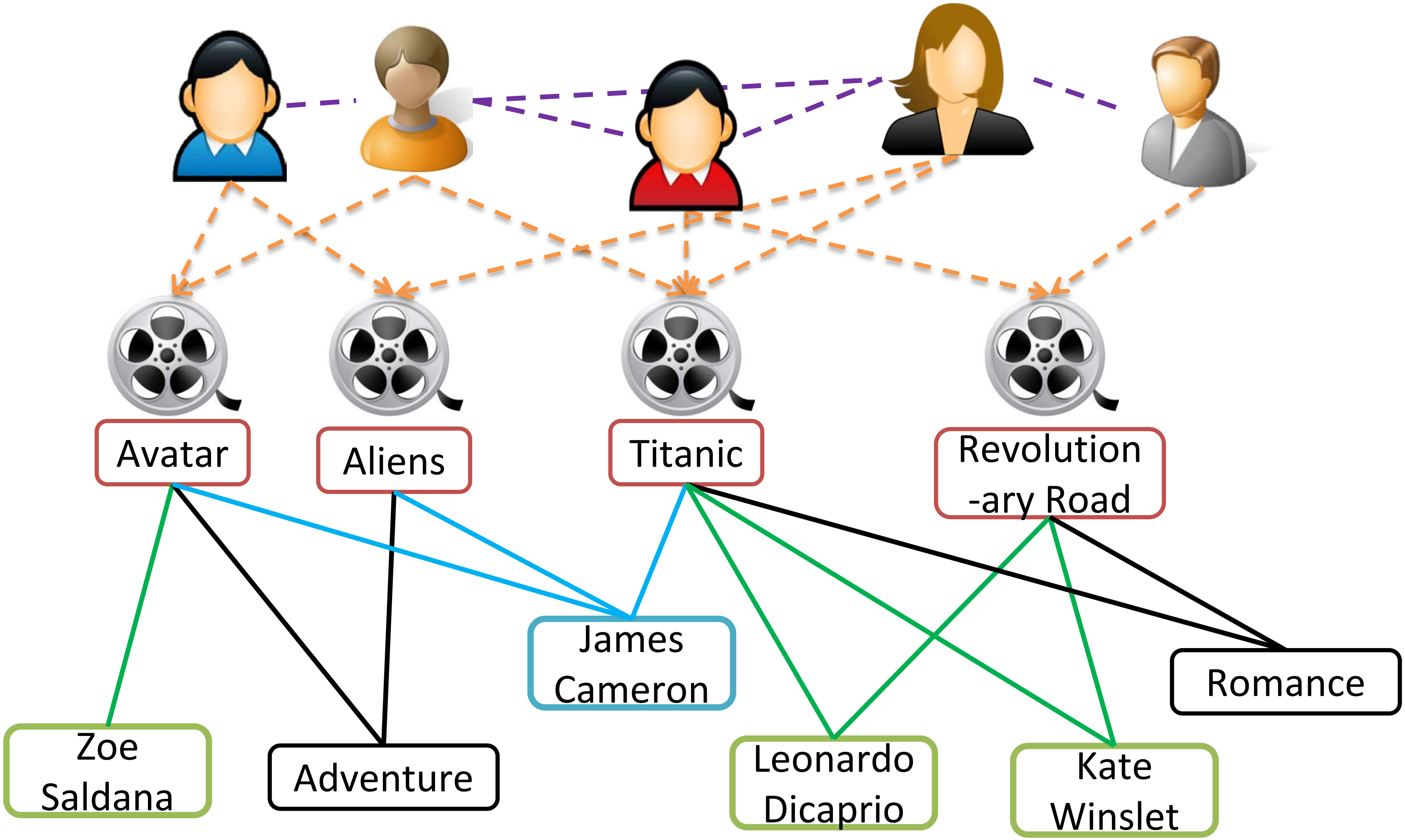
- Dataset: Epinions
- Metric: RMSE

Dataset	Traning Data	Dimensionality	PMF	SoRec	RWD	RWT
Epinions	5%	5D	1.228	1.199	1.186	**1.177**
		10D	1.214	1.198	1.185	**1.176**
	10%	5D	0.990	0.944	0.932	**0.924**
		10D	0.977	0.941	0.931	**0.923**
	20%	5D	0.819	0.788	0.723	**0.721**
		10D	0.818	0.787	0.723	**0.720**

Hybrid Collaborative Filtering with Networks

- Utilizing network relationship information can enhance the recommendation quality
- However, most of the previous studies only use single type of relationship between users or items (e.g., social network Ma,WSDM'11, trust relationship Ester, KDD'10, service membership Yuan, RecSys'11)

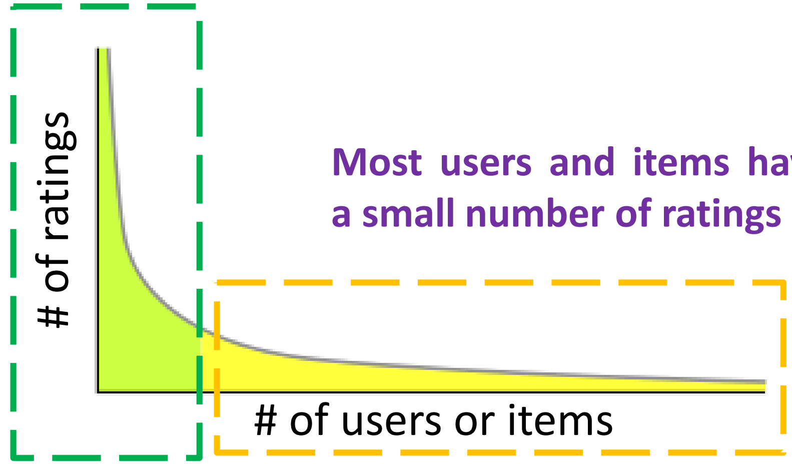
The Heterogeneous Information Network View of Recommender System



Relationship Heterogeneity Alleviates Data Sparsity

Collaborative filtering methods suffer from data sparsity issue

A small number of users and items have a large number of ratings

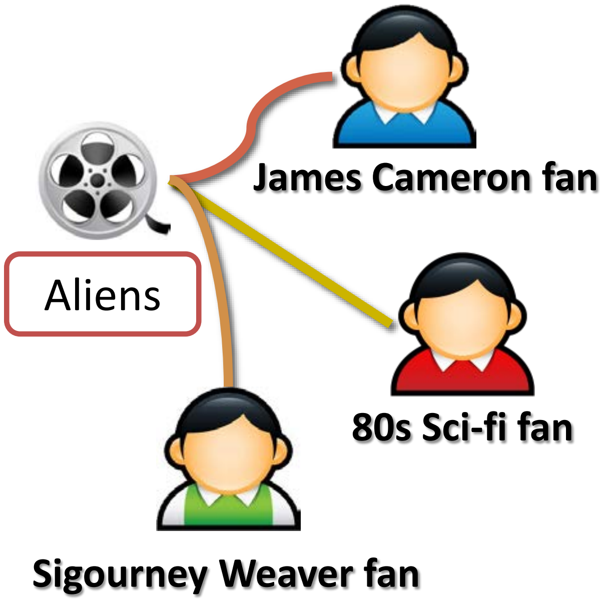


Most users and items have a small number of ratings

- Heterogeneous relationships complement each other
- Users and items with limited feedback can be connected to the network by different types of paths
  - Connect new users or items (cold start) in the information network

Relationship Heterogeneity Based Personalized Recommendation Models (Yu et al., WSDM'14)

Different users may have different behaviors or preferences



Different users may be interested in the same movie for different reasons

Two levels of personalization

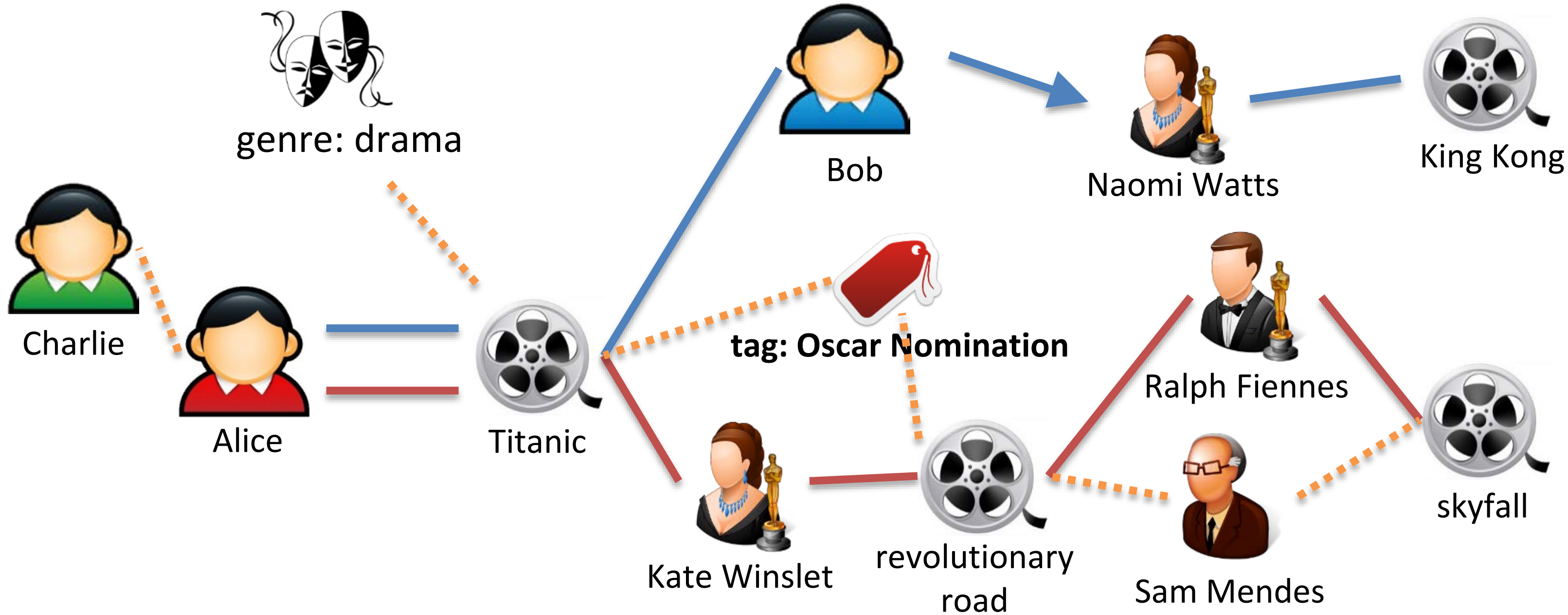
Data level

- Most recommendation methods use one model for all users and rely on personal feedback to achieve personalization

Model level

- With different entity relationships, we can learn personalized models for different users to further distinguish their differences

Preference Propagation-Based Latent Features



Generate *L* different **meta-path** (path types) connecting users and items

→

Propagate user implicit feedback along each meta-path

→

Calculate latent-features for users and items for each meta-path with **NMF** related method

Recommendation Models

Observation 1: Different meta-paths may have different importance

Global Recommendation Model

ranking score

features for user *i* and item *j*

$$\boxed{\hat{r}(u_i, e_j)} = \sum_{q=1}^{L} \theta_q \cdot \boxed{\hat{U}_i^{(q)} \hat{V}_j^{(q)T}} \quad (1)$$

the q-th meta-path

Observation 2: Different users may require different models

Personalized Recommendation Model

user-cluster similarity

$$\hat{r}_p(u_i, e_j) = \sum_{k=1}^{c} \boxed{sim(C_k, u_i)} \sum_{q=1}^{L} \theta_q^{\{k\}} \cdot \hat{U}_i^{(q)} \hat{V}_j^{(q)T} \quad (2)$$

c total soft user clusters

Parameter Estimation

- Bayesian personalized ranking (Rendle UAI'09)
- Objective function

sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$.

$$\min_{\Theta} \; -\sum_{u_i \in \mathcal{U}} \boxed{\sum_{(e_a > e_b) \in R_i}} \ln \sigma(\hat{r}(u_i, e_a) - \hat{r}(u_i, e_b)) + \frac{\lambda}{2}\|\Theta\|_2^2 \qquad (3)$$

for each correctly ranked item pair
i.e., u_i gave feedback to e_a but not e_b

Soft cluster users with NMF + k-means → **For each user cluster, learn one model with Eq. (3)** → **Generate personalized model for each user on the fly with Eq. (2)**

Learning Personalized Recommendation Model

Experiment Setup

- Datasets

Name	#Items	#Users	#Ratings	#Entities	#Links
IM100K	943	1360	89,626	60,905	146,013
Yelp	11,537	43,873	229,907	285,317	570,634

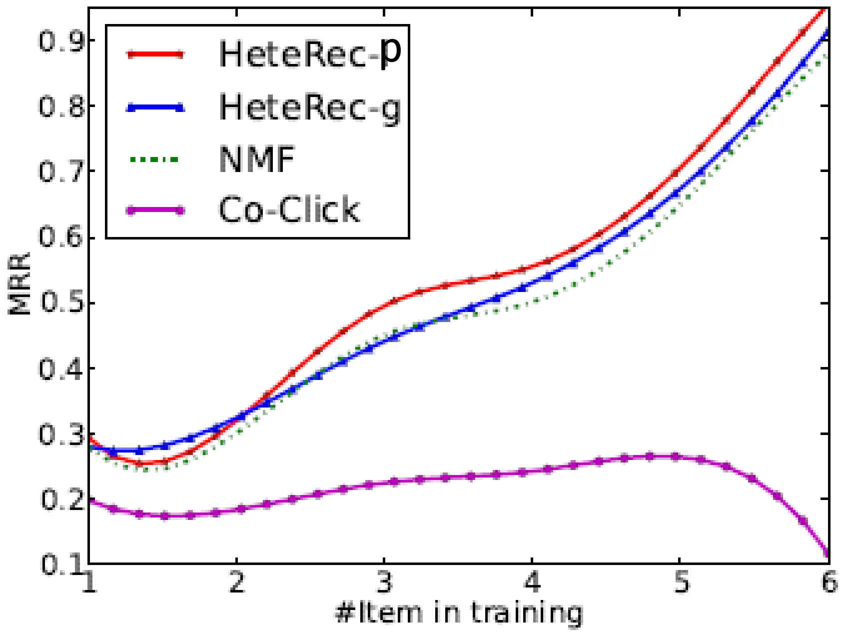
- Comparison methods:
  - Popularity: recommend the most popular items to users
  - Co-click: conditional probabilities between items
  - NMF: non-negative matrix factorization on user feedback
  - Hybrid-SVM: use Rank-SVM with plain features (utilize both user feedback and information network)

Performance Comparison

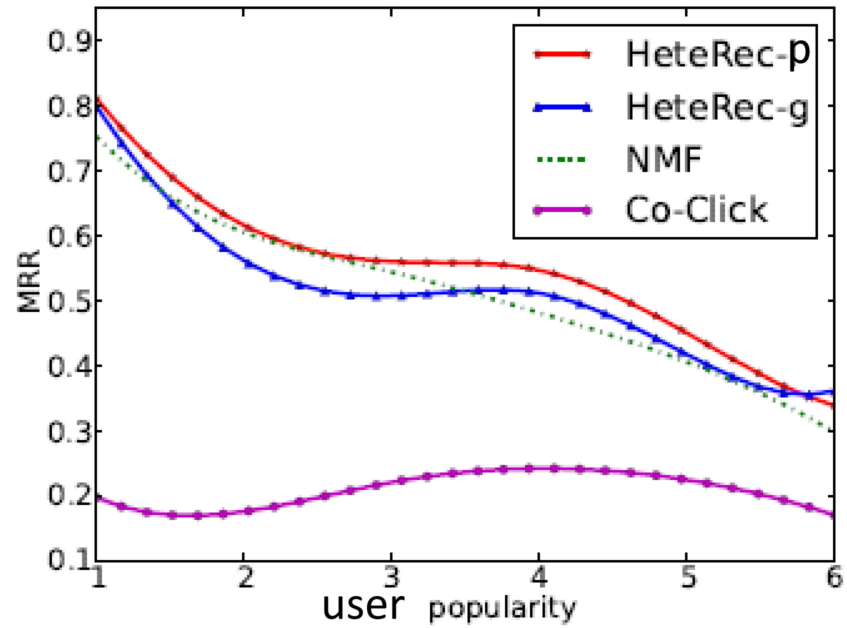
Method	IM100K				Yelp			
	Prec1	Prec5	Prec10	MRR	Prec1	Prec5	Prec10	MRR
Popularity	0.0731	0.0513	0.0489	0.1923	0.00747	0.00825	0.00780	0.0228
Co-Click	0.0668	0.0558	0.0538	0.2041	0.0147	0.0126	0.01132	0.0371
NMF	0.2064	0.1661	0.1491	0.4938	0.0162	0.0131	0.0110	0.0382
Hybrid-SVM	0.2087	0.1441	0.1241	0.4493	0.0122	0.0121	0.0110	0.0337
HeteRec-g	0.2094	0.1791	0.1614	0.5249	0.0165	0.0144	0.0129	0.0422
HeteRec-p	**0.2121**	**0.1932**	**0.1681**	**0.5530**	**0.0213**	**0.0171**	**0.0150**	**0.0513**

HeteRec personalized recommendation (HeteRec-p) provides the best recommendation results

Performance under Different Scenarios



(a) Performance Change with User Feedback Number



(b) Performance Change with User Feedback Popularity

HeteRec–p consistently outperform other methods in different scenarios

better recommendation results if users provide more feedback

better recommendation for users who like less popular items

Recommender Systems

- Recommendation via Information Network Analysis
- Hybrid Collaborative Filtering with Information Networks
- Graph Regularization for Recommendation
- Summary

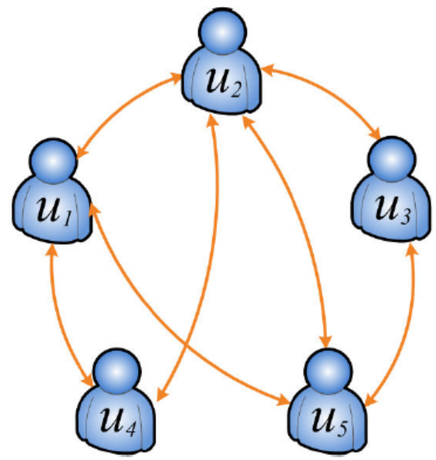
From Graph Regularization Point of View

- Why additional links help?
  - They define new similarity metrics between users or items.
- How to integrate this assumption into recommendation?
  - Use graph regularization to force two entities to be similar in latent space, if they are similar in graph
- The original form of graph regularization
  - $\frac{1}{2}\sum w_{ij}\left(f_i - f_j\right)^2 = f'Lf$
    - w_{ij} : $similarity\ of\ node\ i\ and\ j$
    - f_i: some latent representation for node i
    - L: Laplacian matrix of W, i.e., $L = D - W$,
      - $where\ D\ is\ a\ diagonal\ matrix\ and\ D_{ii} = \sum_j w_{ij}$

Recommender Systems with Social Regularization [Ma et al., WSDM'11]

- Input: Social Relation + Rating Matrix

(a) Real World Social Recommendation



(b) Social Network

	v_1	v_2	v_3	v_4	v_5
u_1	1		2	3	
u_2		3			1
u_3		4		5	
u_4	5			4	
u_5		2	5		4

(c) User-Item Rating Matrix

Two Regularization Forms

- Model 1: Average-based Regularization
  - We are similar to the average of our friends

$$
\begin{aligned}
\min_{U,V} \mathcal{L}_1(R,U,V) &= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2 \\
&+ \frac{\alpha}{2}\sum_{i=1}^{m} \|U_i - \frac{1}{|\mathcal{F}^+(i)|}\sum_{f\in\mathcal{F}^+(i)} U_f\|_F^2 \\
&+ \frac{\lambda_1}{2}\|U\|_F^2 + \frac{\lambda_2}{2}\|V\|_F^2, \qquad (5)
\end{aligned}
$$

- Model2: Individual-based Regularization
  - We are similar to each of our friends

$$
\begin{aligned}
\min_{U,V} \mathcal{L}_2(R,U,V) &= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2 \\
&+ \frac{\beta}{2}\sum_{i=1}^{m}\sum_{f\in\mathcal{F}^+(i)} Sim(i,f)\|U_i - U_f\|_F^2 \\
&+ \lambda_1\|U\|_F^2 + \lambda_2\|V\|_F^2. \qquad (11)
\end{aligned}
$$

Similarity can be propagated via friends: transitivity!

How to compute similarity between two users?

- Cosine similarity (VSS)

$$Sim(i,f) = \frac{\sum_{j \in I(i) \cap I(f)} R_{ij} \cdot R_{fj}}{\sqrt{\sum_{j \in I(i) \cap I(f)} R_{ij}^2} \cdot \sqrt{\sum_{j \in I(i) \cap I(f)} R_{fj}^2}}.$$

- Pearson correlation coefficient (PCC)

$$Sim(i,f) = \frac{\sum_{j \in I(i) \cap I(f)} (R_{ij} - \overline{R}_i) \cdot (R_{fj} - \overline{R}_f)}{\sqrt{\sum_{j \in I(i) \cap I(f)} (R_{ij} - \overline{R}_i)^2} \cdot \sqrt{\sum_{j \in I(i) \cap I(f)} (R_{fj} - \overline{R}_f)^2}}, \tag{14}$$

Results

Table 5: Performance Comparisons (Dimensionality = 10)

Dataset	Training	Metrics	UserMean	ItemMean	NMF	PMF	RSTE	$SR1_{vss}$	$SR1_{pcc}$	$SR2_{vss}$	$SR2_{pcc}$
Douban	80%	MAE	0.6809	0.6288	0.5732	0.5693	0.5643	0.5579	0.5576	0.5548	**0.5543**
		Improve	18.59%	11.85%	3.30%	2.63%	1.77%				
		RMSE	0.8480	0.7898	0.7225	0.7200	0.7144	0.7026	0.7022	0.6992	**0.6988**
		Improve	17.59%	11.52%	3.28%	2.94%	2.18%				
	60%	MAE	0.6823	0.6300	0.5768	0.5737	0.5698	0.5627	0.5623	0.5597	**0.5593**
		Improve	18.02%	11.22%	3.03%	2.51%	1.84%				
		RMSE	0.8505	0.7926	0.7351	0.7290	0.7207	0.7081	0.7078	0.7046	**0.7042**
		Improve	17.20%	11.15%	4.20%	3.40%	2.29%				
	40%	MAE	0.6854	0.6317	0.5899	0.5868	0.5767	0.5706	0.5702	0.5690	**0.5685**
		Improve	17.06%	10.00%	3.63%	3.12%	1.42%				
		RMSE	0.8567	0.7971	0.7482	0.7411	0.7295	0.7172	0.7169	0.7129	**0.7125**
		Improve	16.83%	10.61%	4.77%	3.86%	2.33%				
Epinions	90%	MAE	0.9134	0.9768	0.8712	0.8651	0.8367	0.8290	0.8287	0.8258	**0.8256**
		Improve	9.61%	15.48%	5.23%	4.57%	1.33%				
		RMSE	1.1688	1.2375	1.1621	1.1544	1.1094	1.0792	1.0790	1.0744	**1.0739**
		Improve	8.12%	13.22%	7.59%	6.97%	3.20%				
	80%	MAE	0.9285	0.9913	0.8951	0.8886	0.8537	0.8493	0.8491	0.8447	**0.8443**
		Improve	9.07%	14.83%	5.68%	4.99%	1.10%				
		RMSE	1.1817	1.2584	1.1832	1.1760	1.1256	1.1016	1.1013	1.0958	**1.0954**
		Improve	7.30%	12.95%	7.42%	6.85%	2.68%				

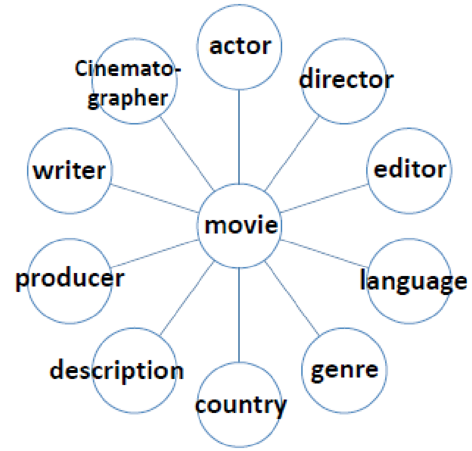
Meta-Path-based Regularization [Yu et al., IJCAI-HINA'13]

- What if it is more than one type of relation?

	E1	e2	…	em
u1	0	0	0	1
u2	0	2	0	5
…	0	0	0	0
un	3	4	0	0

Rating Data

+



Heterogeneous Information Network

- Solution:
  - Use meta-path to generate similarity relation between items, e.g., movie-director-movie
  - Learn the importance score for each meta-path

Notations

- We have *n* users and *m* items.
  - $\mathcal{U} = \{u_1, ..., u_n\}$ $\mathcal{I} = \{e_1, ..., e_m\}$
- By computing similarity scores of all item pairs along certain meta-path, we can get a similarity matrix
  - $S^{(l)} \in \mathbb{R}^{n \times n}$
- With *L* different meta-paths, we can calculate *L* similarity matrices as
  - $S^{(1)}, S^{(2)}, ..., S^{(L)}$

Objective Function

Approximate R with U V product

Regularization on U V

$$\min_{U,V,\boldsymbol{\theta}} \quad \|Y \odot (R - UV^T)\|_F^2 + \lambda_0(\|U\|_F^2 + \|V\|_F^2) +$$

$$\frac{\lambda_1}{2} \cdot \sum_{i,j} \sum_{l=1}^{L} \theta_l S_{ij}^{(l)} \|V_i - V_j\|_2^2 + \lambda_2 \|\boldsymbol{\theta}\|_2^2,$$

Similar items measured from HIN should have similar low-rank representations

Regularization on *ϑ, which is the importance score for each meta-path*

$$\text{s.t.} \quad U \geq 0, \;\; V \geq 0, \;\; \boldsymbol{\theta} \geq 0, \;\; \text{and} \;\; \sum_{l=1}^{L} \theta_l = 1,$$

Equivalent Objective Function Using Graph Laplacian

$$D_{ii}^{(l)} = \sum_{j=1}^{n} S_{ij}^{(l)} \qquad L^{(l)} = D^{(l)} - S^{(l)}$$

$$\min_{U,V,\boldsymbol{\theta}} \quad \|Y \odot (R - UV^T)\|_F^2 + \lambda_0(\|U\|_F^2 + \|V\|_F^2) + \boxed{\lambda_1 \cdot \mathrm{Tr}\left(V^T\left(\sum_l \theta_l L^{(l)}\right)V\right)} + \lambda_2\|\boldsymbol{\theta}\|_F^2,$$

Similar items measured from HIN should have similar low-rank representations

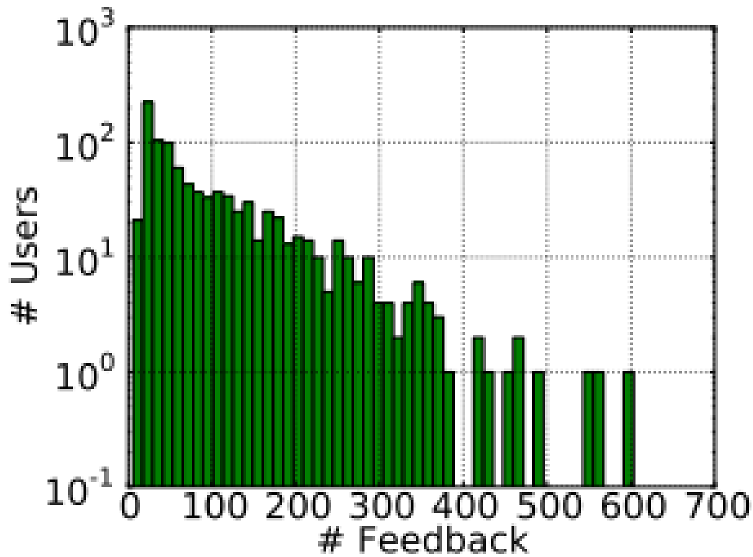
$$\text{s.t.} \quad U \geq 0, \;\; V \geq 0, , \;\; \boldsymbol{\theta} \geq 0, \;\; \text{and} \;\; \sum_{l=1}^{L} \theta_l = 1.$$

Dataset

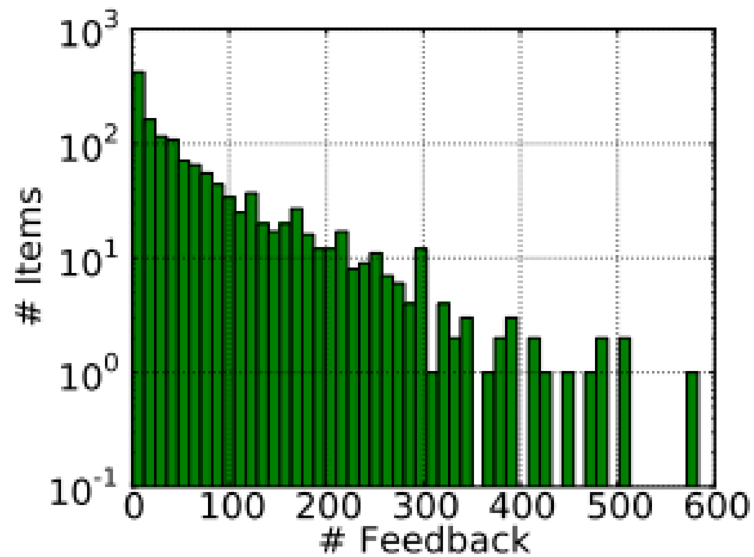
- We combine IMDb + MovieLens100K

Name	#Items	#Users	#Ratings	#Entities	#Links
IM100K	943	1360	89,626	60,905	146,013

(a) Datasets Description



(b) #Ratings vs. #Users



(c) #Ratings vs. Item Popularity

We random sample training datasets of different sizes (0.4, 0.6, and 0.8)

Results

Metric	MAE			RMSE		
Training Size	40%	60%	80%	40%	60%	80%
UserMean	0.8400	0.8409	0.8324	1.0479	1.0482	1.0407
ItemMean	0.8167	0.8237	0.8130	1.0281	1.0354	1.0235
NMF (d=40)	2.1944	2.1862	2.0162	2.4459	2.4391	2.2915
WNMF (d=10)	0.7919	0.7879	0.7589	1.0055	1.0028	0.9677
WNMF (d=20)	0.7917	0.7875	0.7591	1.0060	1.0026	0.9681
WNMF (d=40)	0.7886	0.7833	0.7569	1.0027	0.9991	0.9655
Hete-MF (d=10)	0.7838	0.7800	0.7530	0.9950	0.9931	0.9683
Hete-MF (d=20)	0.7818	0.7802	0.7528	0.9941	0.9938	0.9593
Hete-MF (d=40)	**0.7780**	**0.7772**	**0.7400**	**0.9900**	**0.9905**	**0.9503**

Recommender Systems

- Recommendation via Information Network Analysis
- Hybrid Collaborative Filtering with Information Networks
- Graph Regularization for Recommendation
- Summary

Summary

- Recommendation via Information Network Analysis
  - Users and items are embedded in a heterogeneous information network
  - Recommendation can be considered as a link prediction problem
- Hybrid Collaborative Filtering with Information Networks
  - Propagate the feedback via meta-paths
- Graph Regularization for Recommendation
  - Similar items/users should have similar latent vectors

Peer Evaluation Questions

1. Is the proposed problem interesting and novel?	2. Is the problem formalization reasonable?	3. Is the solution solid and reasonable?	4. To what extent the project achieves the claimed goal?	5. How good is the presentation?