Overlapping Community Regularization for Social Recommender Systems

Hui Li, Dingming Wu, Wenbin Tang, Nikos Mamoulis

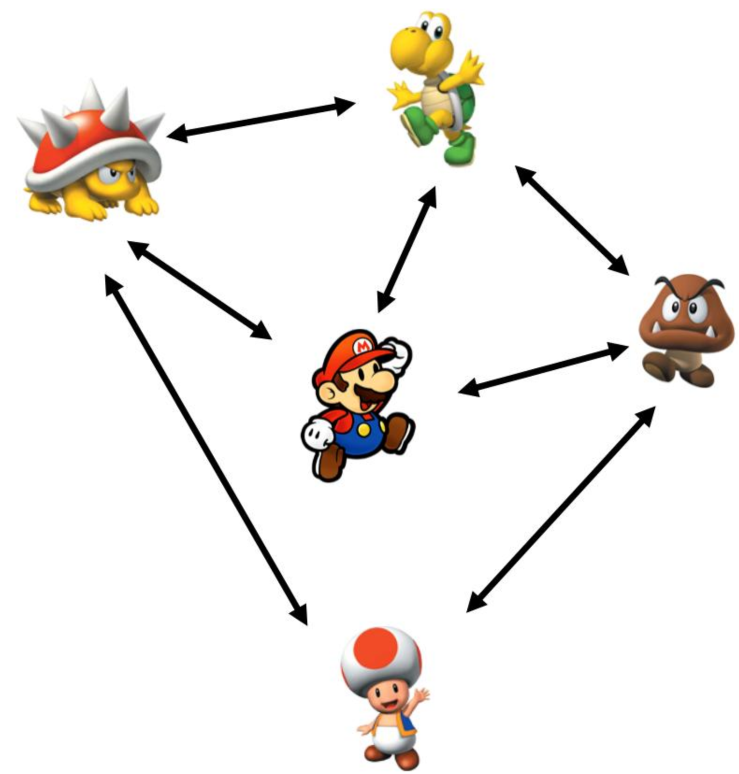
The University of Hong Kong

Outline

- Rating Based Social Recommender Systems in Retrospect
- Community Based Recommender Systems MFC & MFC^+
- Future Work

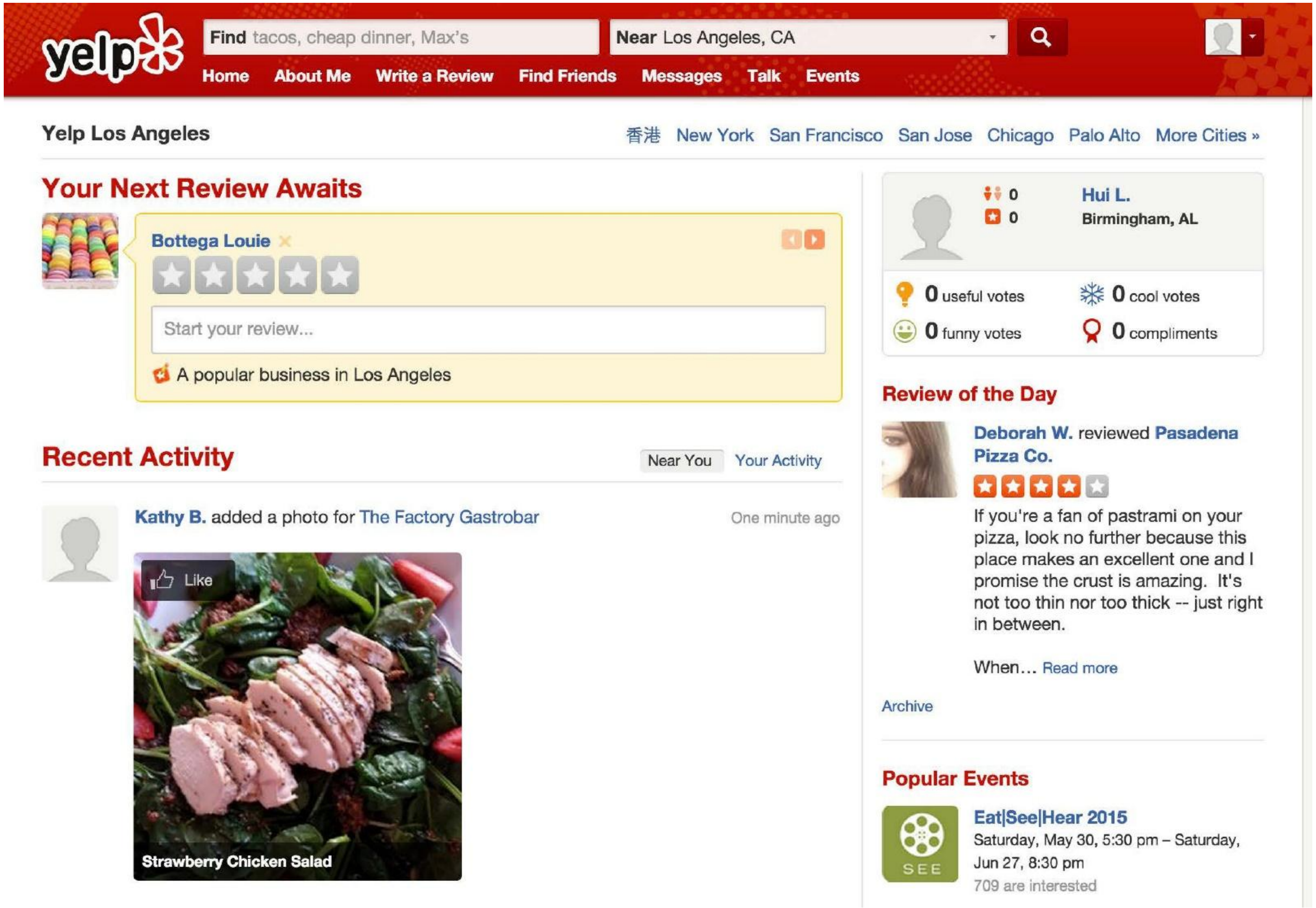
Social Recommender

User-Item Rating Matrix



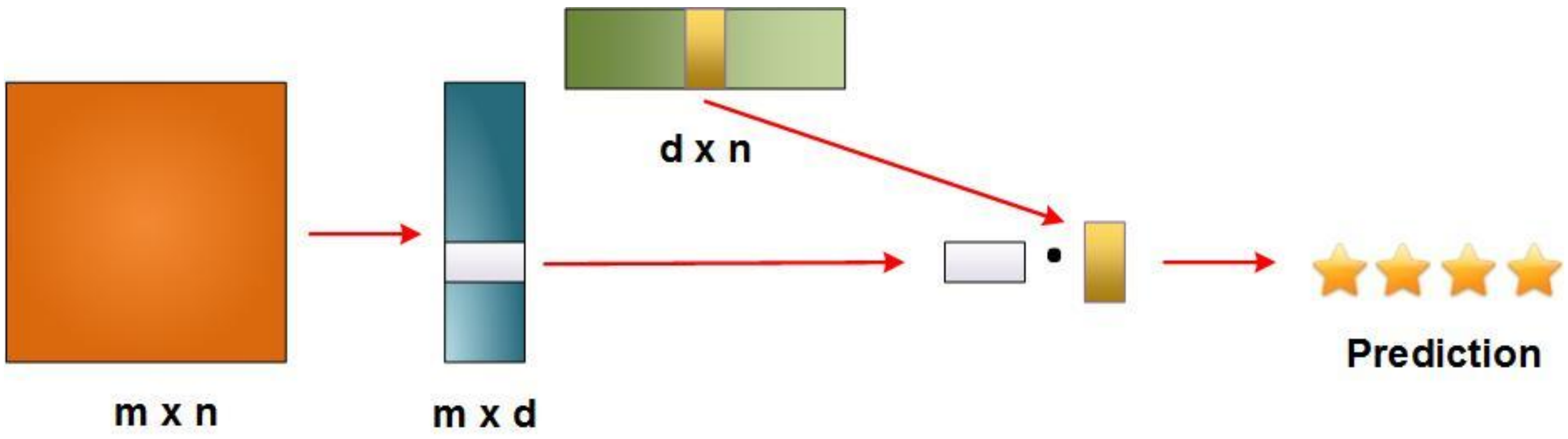
Social Network

Social Recommender

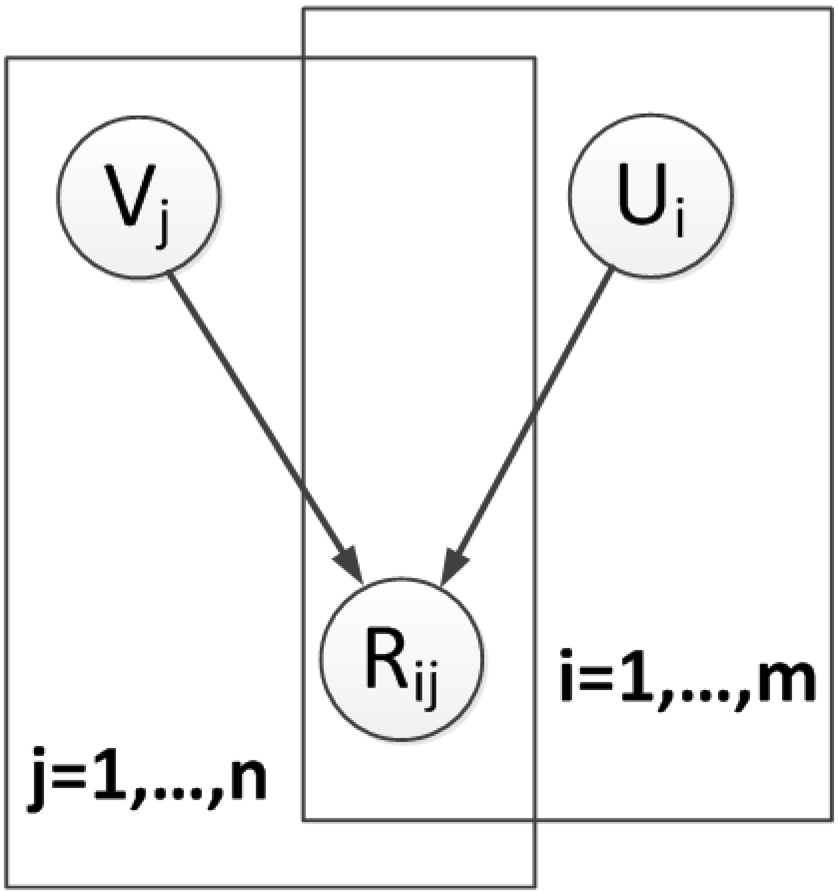


Basic Matrix Factorization

- Given a rank d, factorize user-item rating matrix: $R = U^T V$.
- U and V are the user latent vectors and item latent vectors. d is the feature dimension.



BaseMF



BaseMF

- Loss Function

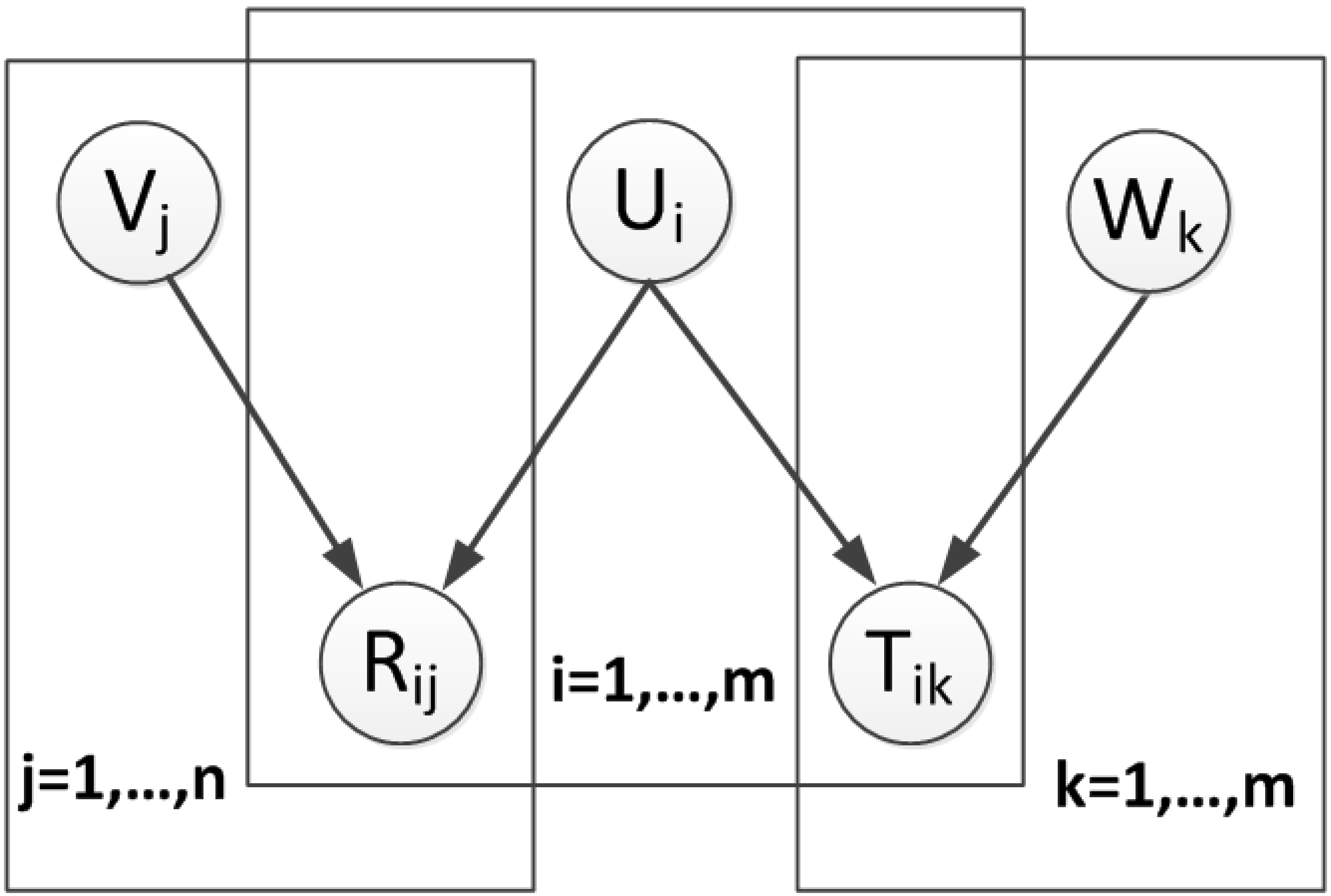
$$L = \min_{U,V} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (r_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2 + \frac{\lambda_1}{2} ||U||_F^2 + \frac{\lambda_2}{2} ||V||_F^2,$$

SoRec (CIKM'08)

- Consider the trust matrix $T = \{T_{ik}\}$. For a pair of friends, $T_{ik} \in (0,1]$ denotes how much user u_i trusts user u_k.
- Then factorize the trust matrix into user-specific and factor-specific vectors.

- Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In CIKM, pages 931–940, 2008.

SoRec (CIKM'08)



SoRec (CIKM'08)

- Loss Function

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_T}{2}\sum_{i=1}^{m}\sum_{k=1}^{m} I_{ik}^{T}(T_{ik} - U_i^T W_k)^2$$

$$+ \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 + \frac{\lambda_W}{2}\|W\|_F^2.$$

STE (SIGIR'09)

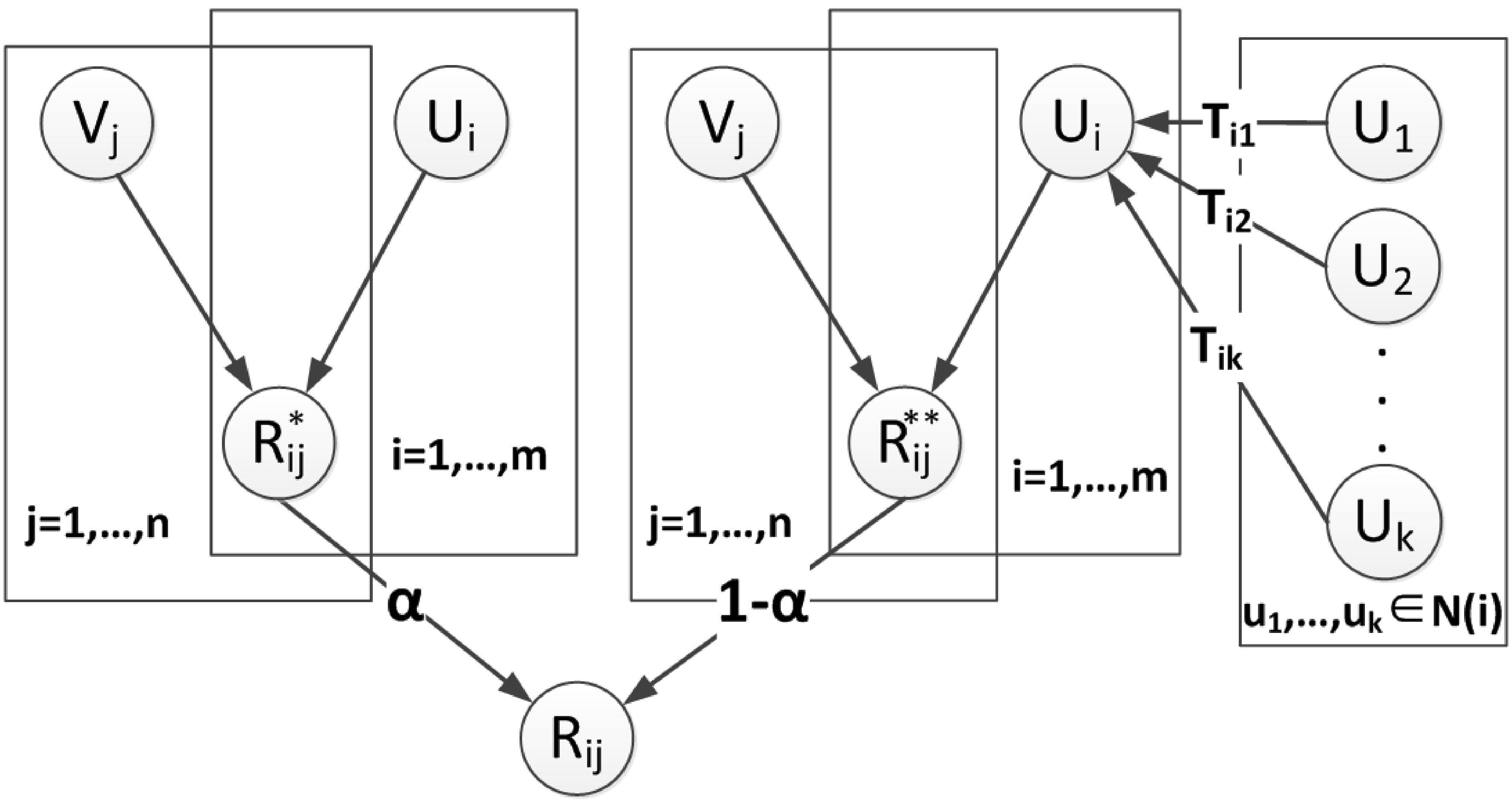
- **Social Trust Ensemble**: every user has his/her own taste and at the same time, every user may be influenced by his/her friends.

$$\alpha U_i^T V_j + (1-\alpha) \sum_{u_k \in N(i)} T_{ik} U_k^T V_j$$

- The user's favors and the trusted friends' favors are smoothed by α. T is the trust matrix.

➢ Hao Ma, Irwin King, and Michael R. Lyu. Learning to recommend with social trust ensemble. In SIGIR, pages 203–210, 2009.

STE (SIGIR'09)



V_j
U_i
R^*_{ij}
i=1,…,m
j=1,…,n
V_j
U_i
R^{**}_{ij}
i=1,…,m
j=1,…,n
T_{i1}
T_{i2}
T_{ik}
U_1
U_2
U_k
$u_1,…,u_k \in N(i)$
α
1-α
R_{ij}

STE (SIGIR'09)

- Loss Function

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}\Big(R_{ij} - \big(\alpha U_i^T V_j + (1-\alpha)\sum_{u_k \in N(i)} T_{ik} U_k^T V_j\big)\Big)^2$$

$$+ \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2,$$

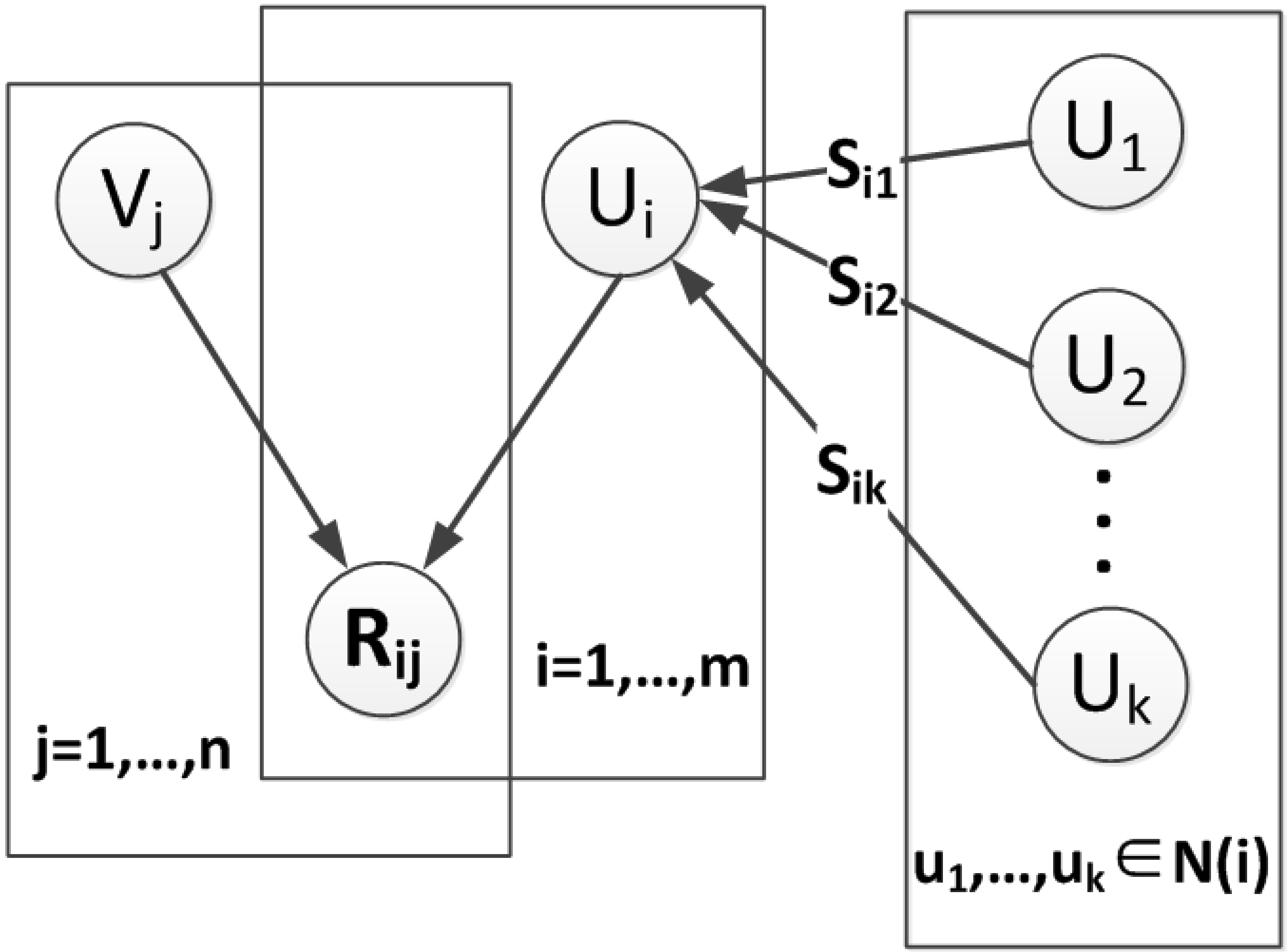
SocialMF (RecSys'10) & SR (WSDM'11)

- **Social Regularization**

  Impose constraints between one user and his friends to minimize the difference between friends' feature vectors.

- Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In RecSys, pages 135–142, 2010.
- Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In WSDM, pages 287–296, 2011.

SocialMF (RecSys'10) & SR (WSDM'11)



SocialMF (RecSys'10) & SR (WSDM'11)

- Loss Function

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2$$
$$+ \frac{\lambda_T}{2}\sum_{i=1}^{m}\|U_u - \sum_{u_w \in N_{u_i}} T_{u_i,u_w} U_w\|_F^2.$$

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2$$
$$+ \frac{\lambda_Z}{2}\sum_{i=1}^{m}\sum_{h=1}^{l} I_{ih}^{Z} Z_{ih} \sum_{u_w \in M_{ih}^{U}} S_{iw}\| U_i - U_w \|_F^2.$$

Two Extensions

- **CircleCon** (KDD'12, Extension of SocialMF)

  Divide direct friends into different circles and each circle corresponds to one category. When making prediction, only one circle is considered according to item category.

  - Xiswang Yang, Harald Steck, and Yong Liu. Circle-based recommendation in online social networks. In KDD, pages 1267–1275, 2012.

- **SR**$^{+}$ (SIGIR'13, Extension of SR)

  Use most similar users instead of friends.

  - Hao Ma. An experimental study on implicit social recommendation. In SIGIR, pages 73–82, 2013.

Outline

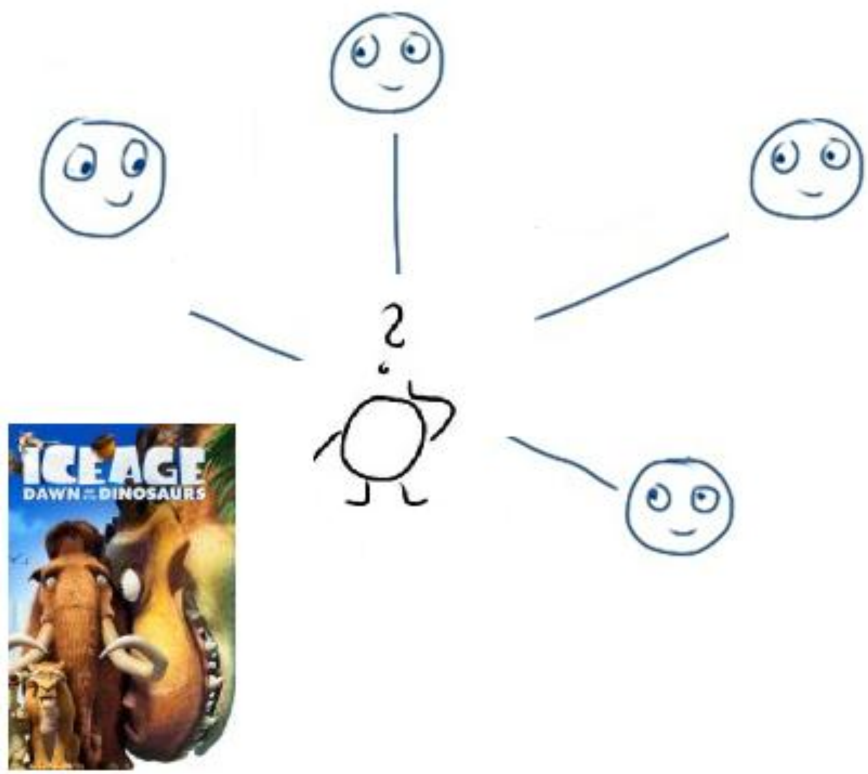
- Rating Based Social Recommender Systems in Retrospect
- Community Based Recommender Systems MFC & MFC^+
- Future Work

Motivation

- One goal of Social Recommender Systems is to alleviate the **cold-start** problem.

Motivation

- One goal of Social Recommender Systems is to alleviate the **cold-start** problem.



Motivation

- One goal of Social Recommender Systems is to alleviate the **cold-start** problem.

Social Cold Start

Community-based RecSys

- A **community** (also called module or cluster) is typically defined as a group of users with more interaction among its members than between its members and the rest of the network.

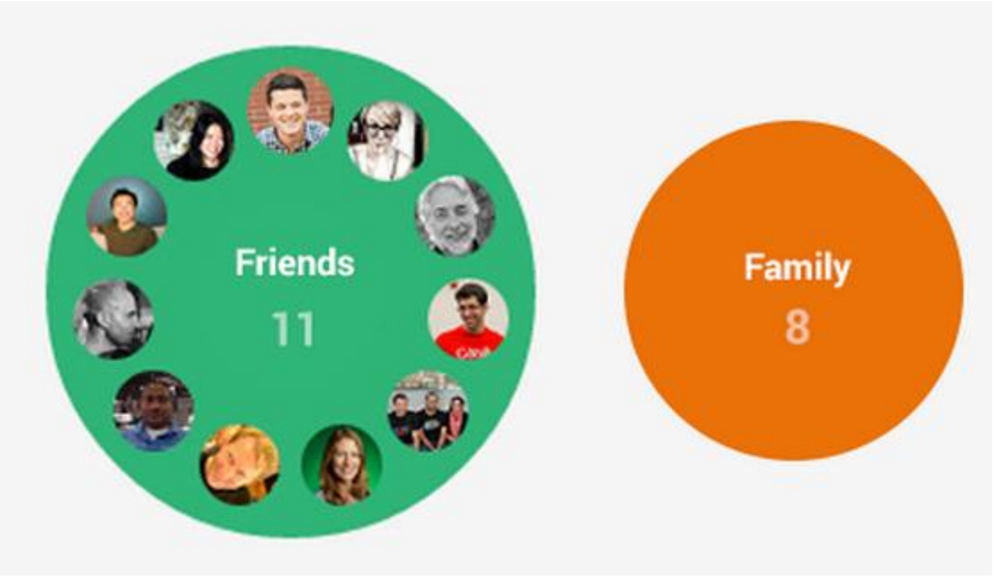
Use information from community members to make the data denser.

Community-based RecSys

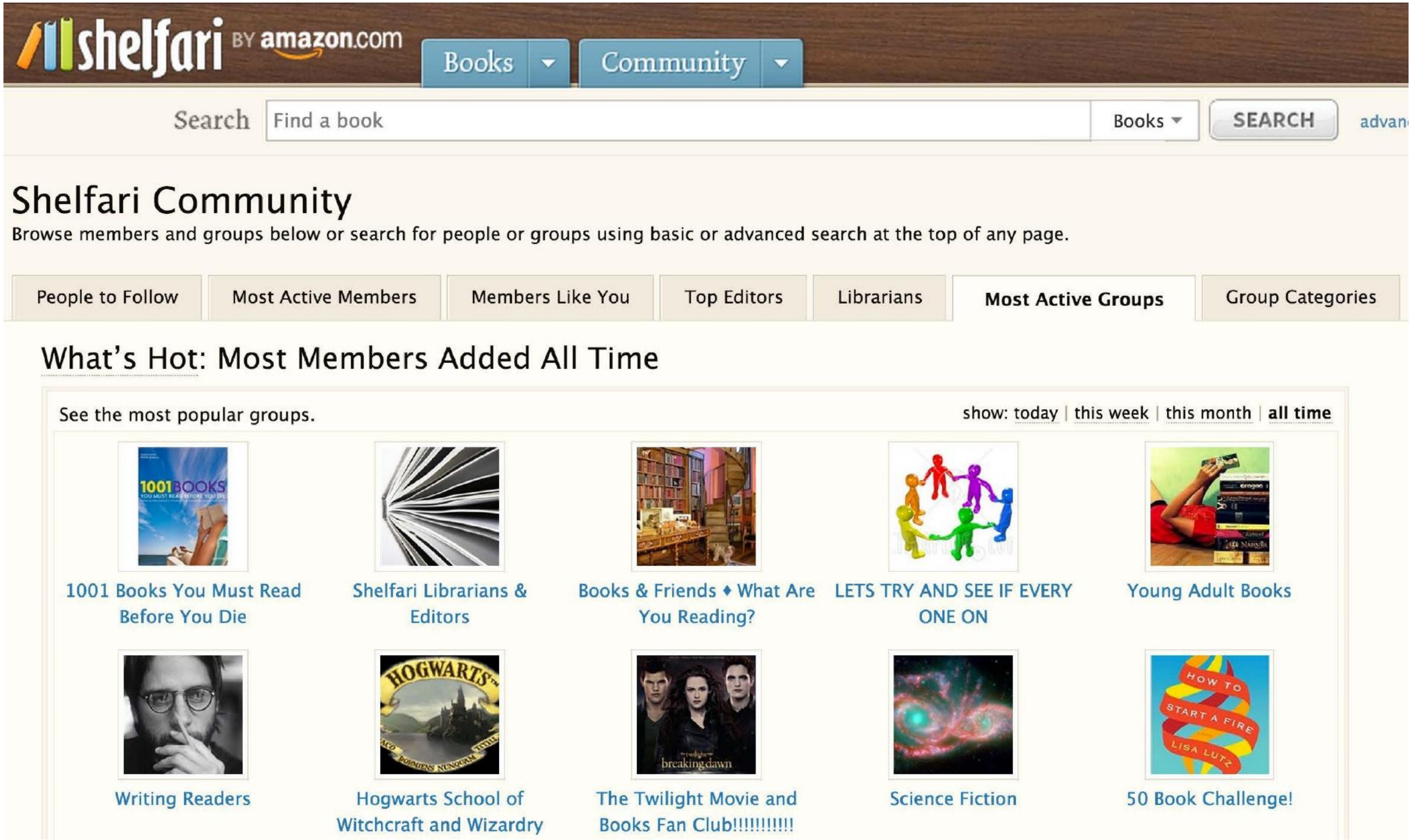
- **Movie rating system (e.g., Douban)**

  Groups topics: Comedy, Romance, etc.

- **Google Plus**



Community-based RecSys



Overlapping Community

- A user may belong to multiple communities (for example, a different reading groups).

- **CPM (Nature 05):** structure of social network

- **BIGCLAM (WSDM'13):** structure of social network

- **CESNA (ICDM'13):** structure of social network + node attribute

Overlapping Community

- S: user-user similarity (e.g., Pearson correlation)

- Z: similarity between user and community

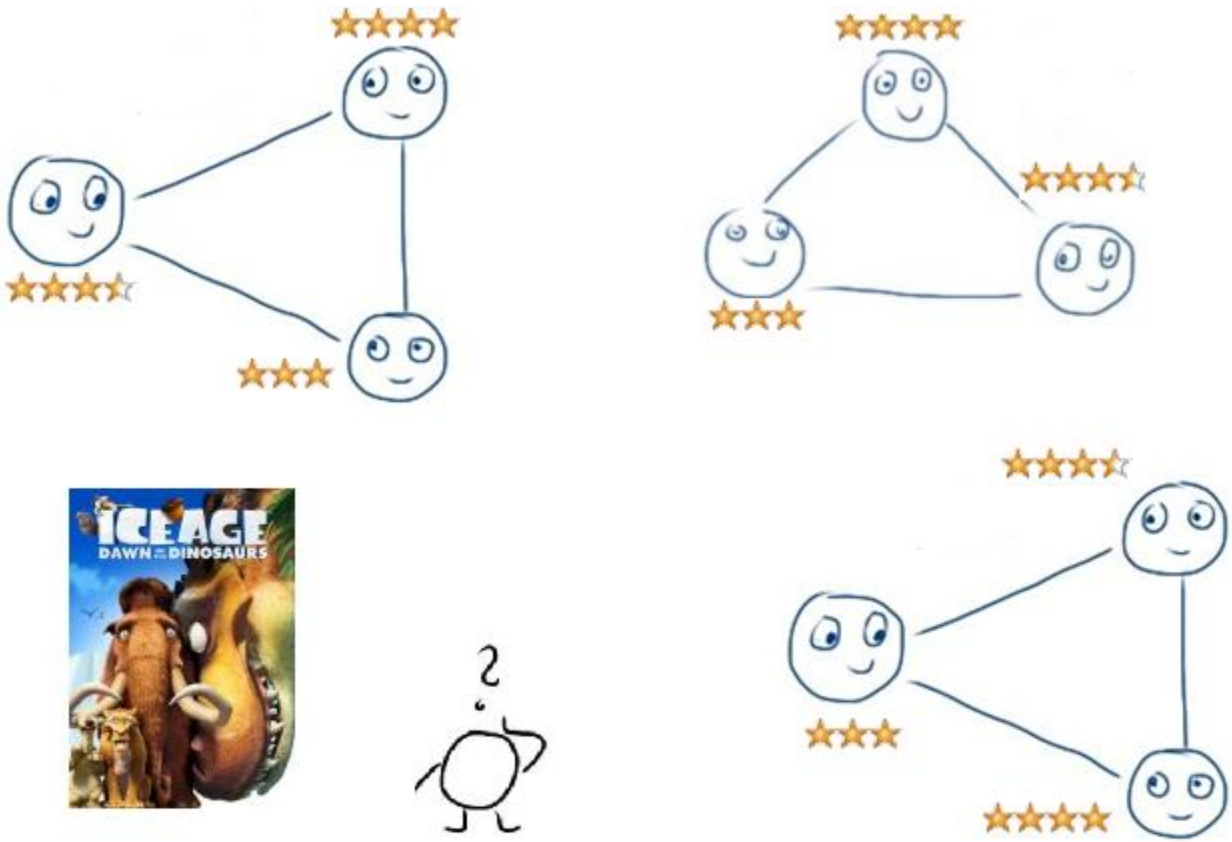
- **User vector**

  user's rating vector

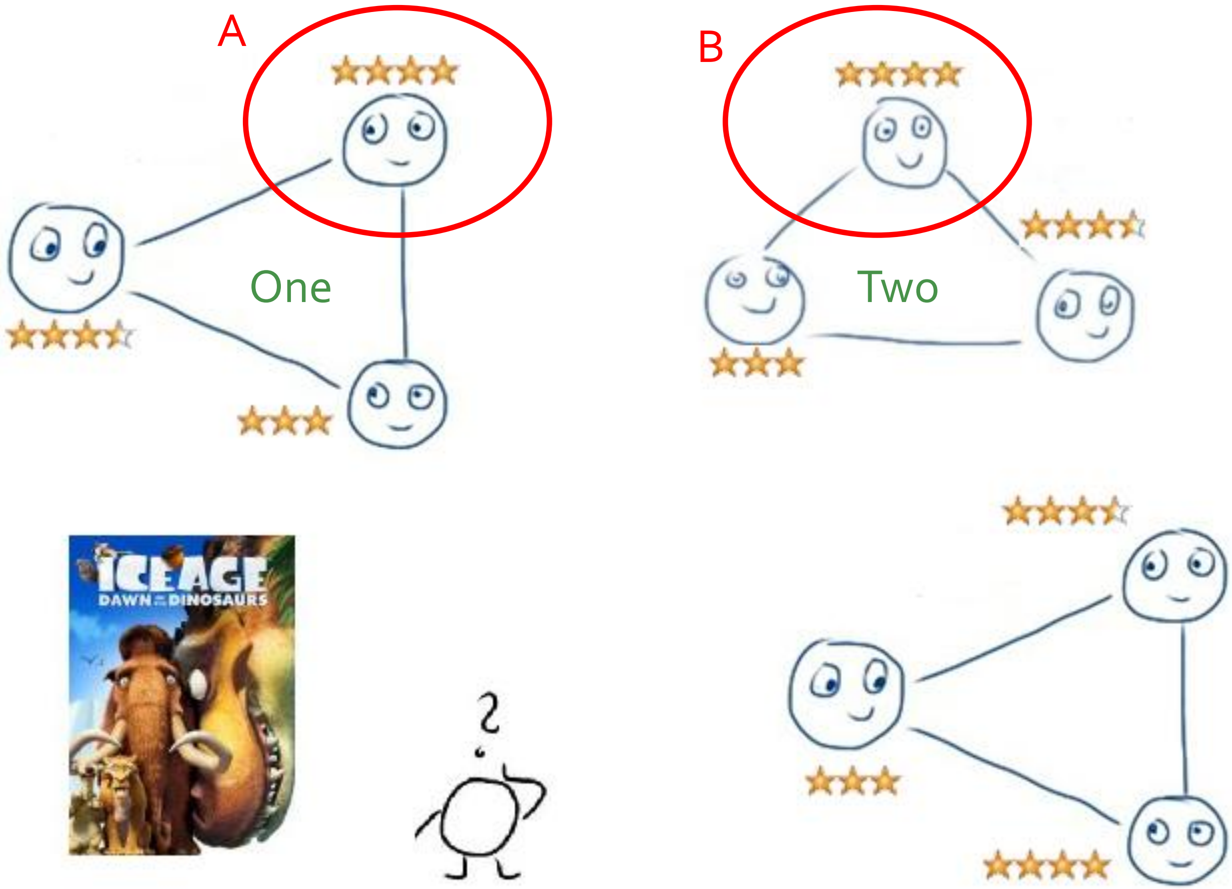
- **Community vector**

  center vector of community members' vectors

MFC



MFC



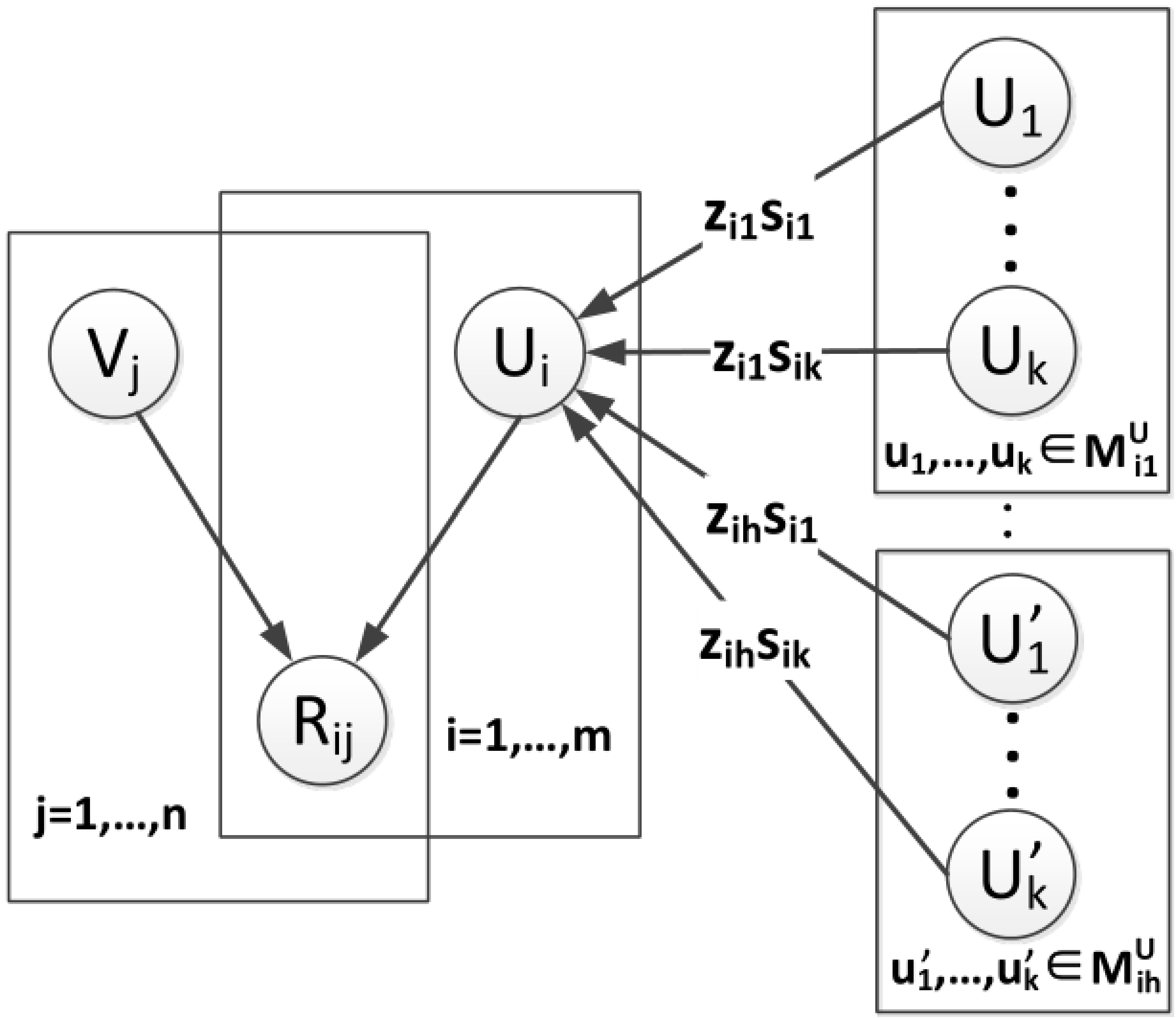
MFC

- Users with the same similarity to the target user may belong to different communities and should be treated differently, while the SR model considers these users equally.

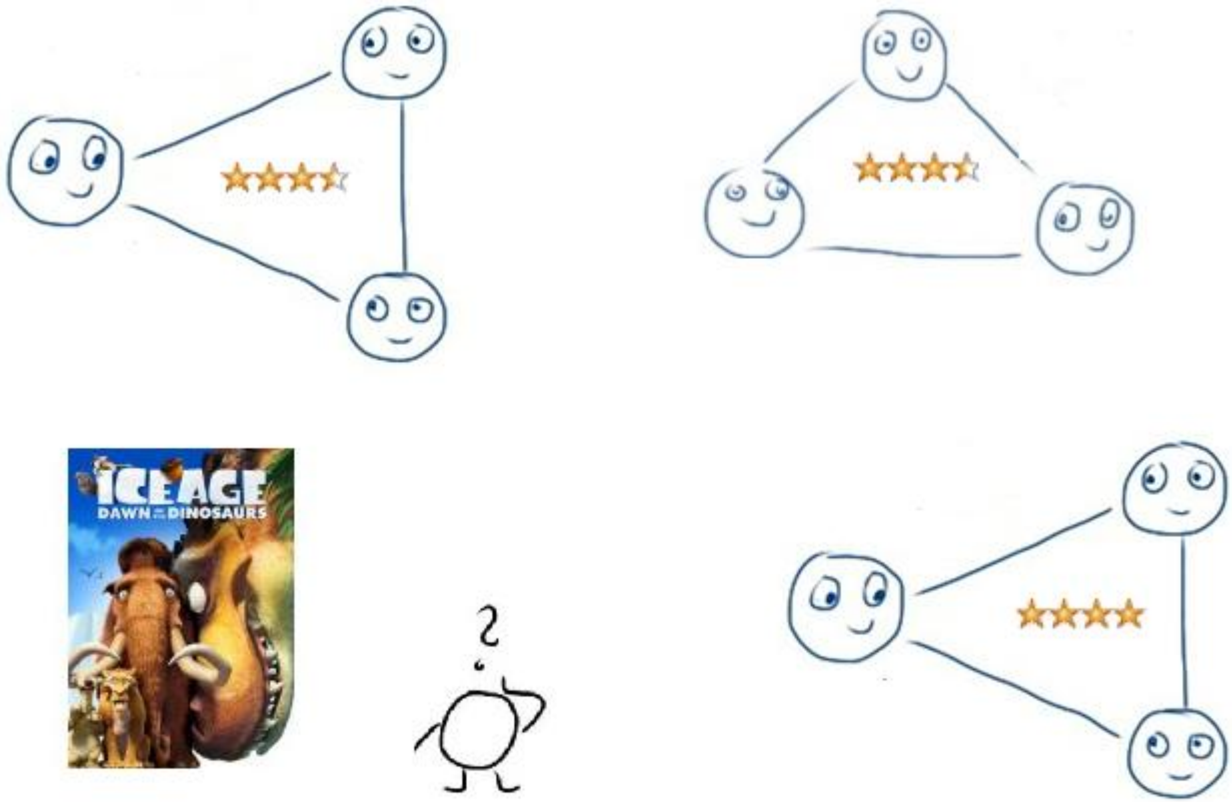
$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V|$$

$$+ \frac{\lambda_Z}{2}\sum_{i=1}^{m}\sum_{h=1}^{l} I_{ih}^{Z} Z_{ih} \sum_{u_w \in M_{ih}^{U}} S_{iw}\| U_i - U_w \|_F^2 .$$

MFC


U_1
$z_{i1}s_{i1}$
V_j
U_i
$z_{i1}s_{ik}$
U_k
$u_1,\ldots,u_k \in M_{i1}^U$
$z_{ih}s_{i1}$
$z_{ih}s_{ik}$
U_1'
R_{ij}
i=1,…,m
j=1,…,n
U_k'
$u_1',\ldots,u_k' \in M_{ih}^U$

MFC$^+$



Community Profile

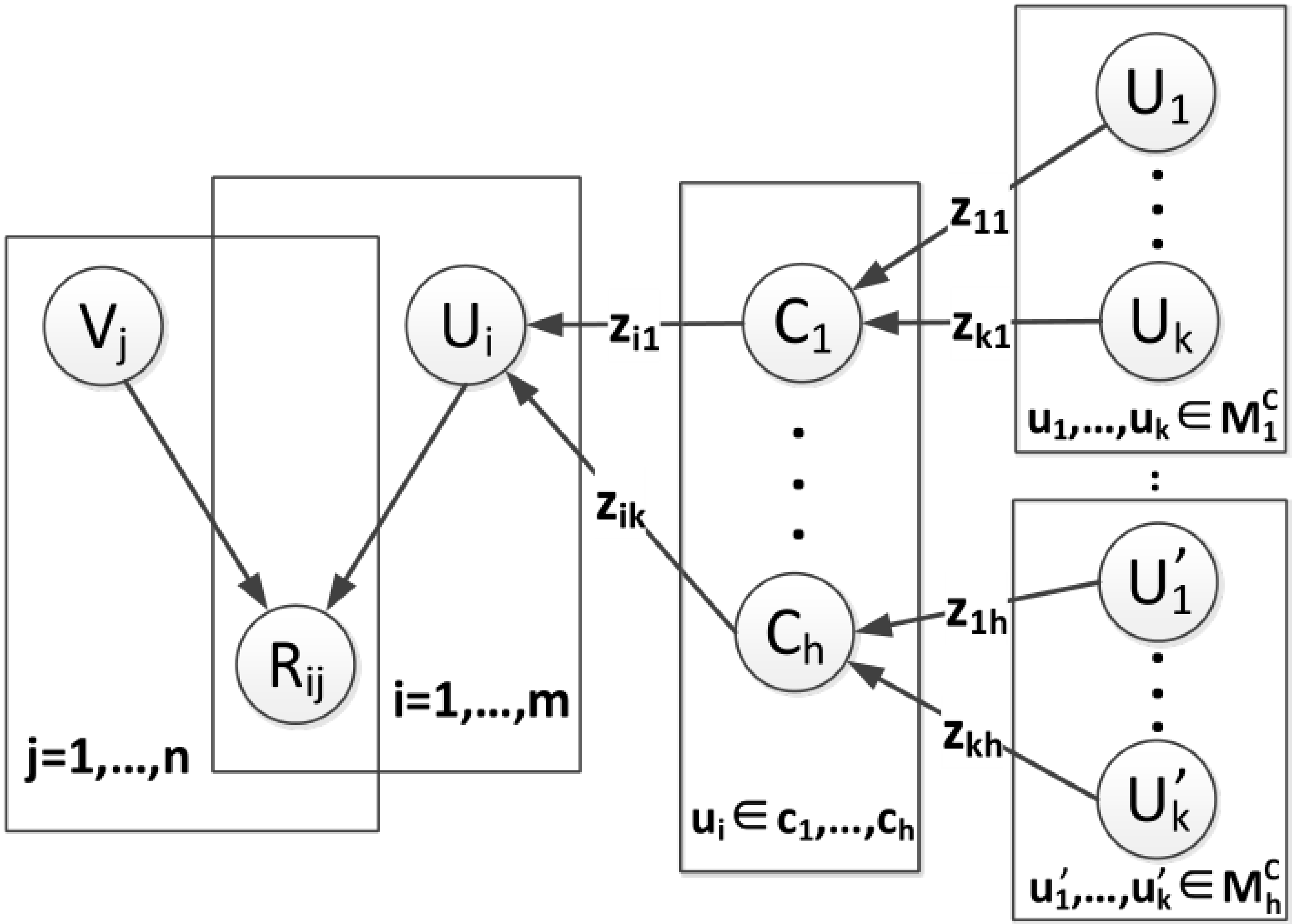
MFC+

- Use the aggregate of all its members' latent factor vectors as the community profile and impose constrains between community profile and the target user

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(R_{ij}-U_i^T V_j)^2+\frac{\lambda_U}{2}\|U\|_F^2+\frac{\lambda_V}{2}\|V\|_F^2$$

$$+\frac{\lambda_Z}{2}\sum_{i=1}^{m}\sum_{h=1}^{l} I_{ih}^{Z} Z_{ih}\left\|U_i-\frac{\sum_{u_w\in M_h^C} Z_{wh}U_w}{\sum_{u_w\in M_h^C} Z_{wh}}\right\|_F^2.$$

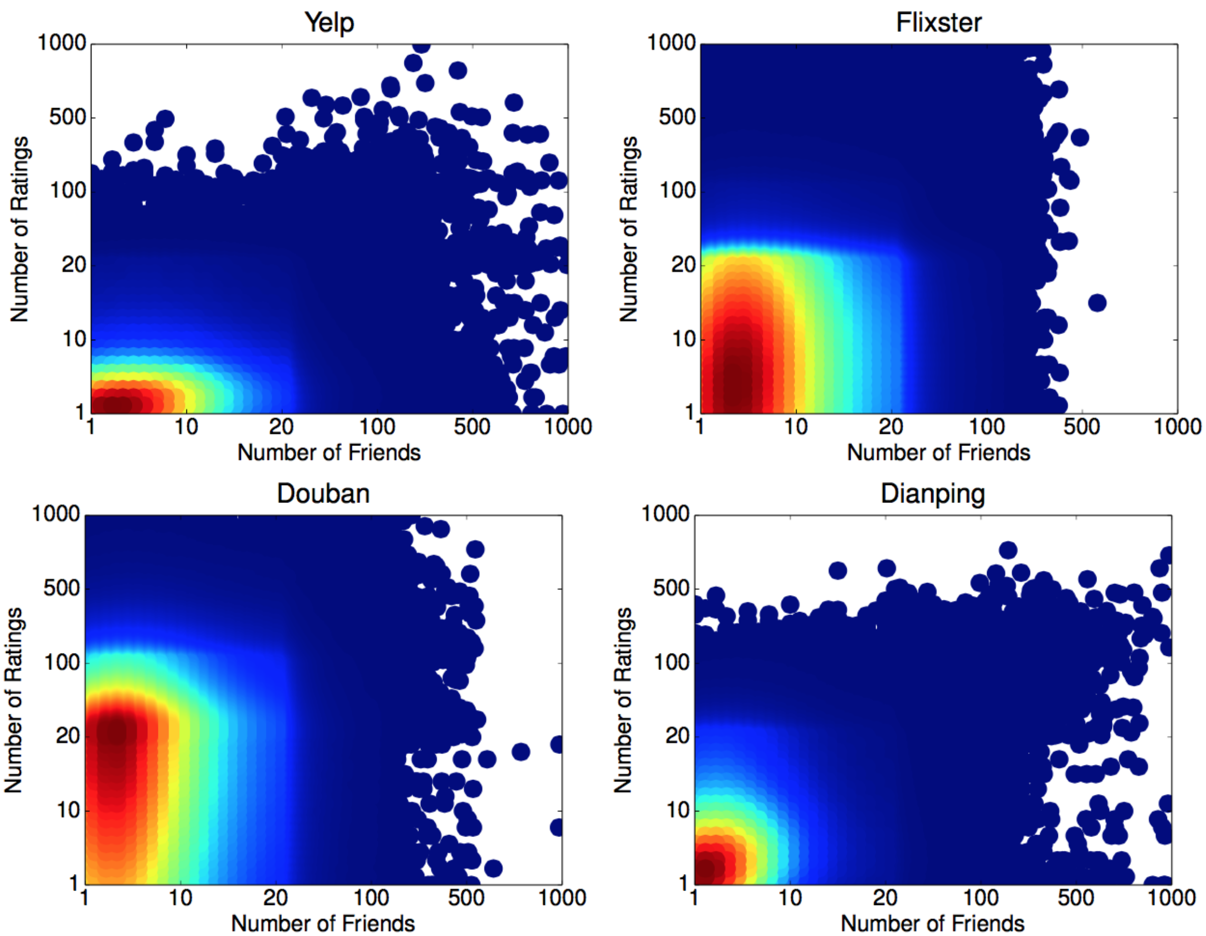
MFC+



U_1
z_{11}
V_j
U_i
z_{i1}
C_1
z_{k1}
U_k
$u_1,\ldots,u_k \in M_1^c$
z_{ik}
U_1'
z_{1h}
C_h
R_{ij}
z_{kh}
$i=1,\ldots,m$
$j=1,\ldots,n$
U_k'
$u_i \in c_1,\ldots,c_h$
$u_1',\ldots,u_k' \in M_h^c$

Experiment

- **Datasets:** Yelp, Flixster, Douban, Dianping



Experiment

- RMSE Performance on All Users

Table 4.2: Performance comparison

Dataset	SCF	BaseMF	SR	SR^+	MFC_p	MFC_p^+	MFC_b	MFC_b^+	MFC_c	MFC_c^+
Yelp	1.4730 24.35%	1.2498 10.84%	1.2216 8.78%	1.2032 7.39%	1.1618	1.1617	1.1543	1.1602	1.1324	**1.1143**
Flixster	1.1761 13.83%	1.1853 14.50%	1.1041 8.21%	1.0823 6.37%	1.0427	1.0436	1.0325	**1.0134**	None	None
Douban	1.2788 31.37%	1.0478 16.24%	0.9441 7.04%	0.9322 5.86%	0.8952	0.8961	**0.8776**	0.8823	None	None
Dianping	1.3642 41.86%	1.0598 25.16%	0.9449 16.05%	0.9012 11.98%	0.8678	0.8748	0.8812	0.8721	**0.7932**	0.8211

p: CPM b: BLGCLAM c: CESNA

Experiment

- **CircleCon**

  When making prediction, only one circle is considered. CircleCon has the same limitation as does SR that only direct friends are considered.

- **Results on the category of restaurants**

Table 4.3: Performance comparison with CircleCon

Dataset	CircleCon	MFC_p	MFC_p^+	MFC_b	MFC_b^+	MFC_c	MFC_c^+
Yelp	1.1907 7.71%	1.1332	1.1345	1.1287	1.1333	1.1001	**1.0989**
Dianping	0.8234 14.84%	0.7754	0.7801	0.7562	0.7612	0.7122	**0.7012**

Experiment

- Performance on rating-cold-start users:

Table 4.4: Performance on *rating-cold-start* users

Dataset	SCF	BaseMF	SR	SR^+	MFC_p	MFC_p^+	MFC_b	MFC_b^+
Yelp	1.7832 28.9%	1.4270 11.16%	1.3769 7.92%	1.3865 8.56%	1.3082	1.3079	**1.2678**	1.2867
Flixster	1.6086 27.42%	1.3573 13.98%	1.2589 7.25%	1.2321 5.23%	1.1747	1.1769	**1.1676**	1.1832
Douban	1.2194 23.90%	1.1198 17.13%	1.0373 10.54%	0.9823 5.53%	**0.9280**	0.9287	0.9331	0.9423
Dianping	1.6098 31.71%	1.3344 17.62%	1.1868 7.37%	1.2012 8.48%	1.1331	1.1197	1.1023	**1.0993**

Experiment

- Performance on social-cold-start users:

Table 4.5: Performance on *social-cold-start* users

Dataset	SCF	BaseMF	SR	SR^+	MFC_p	MFC_p^+	MFC_b	MFC_b^+
Yelp	1.6472 25.2%	1.3902 11.44%	1.3521 8.94%	1.3234 6.97%	1.2938	1.2935	1.2421	**1.2312**
Flixster	1.4080 24.34%	1.2160 12.39%	1.1911 10.56%	1.1342 6.07%	1.0953	1.0956	1.0732	**1.0653**
Douban	1.4320 36.29%	1.1432 20.20%	1.0119 9.84%	1.0012 8.88%	0.9432	0.9234	**0.9123**	0.9321
Dianping	1.3724 35.00%	1.1027 19.10%	0.9847 9.40%	0.9321 4.29%	0.9109	0.9246	**0.8921**	0.9012

Comparison between two models

- Root Mean Square Distance (RMSD):

$$\bar{S}_h = \frac{2\sum_{u_i,u_j \in c_h} S_{ij}}{|C_h|\left(|C_h| - 1\right)},$$

$$RMSD = \sqrt{\frac{2\sum_{u_i,u_j \in c_h} (S_{ij} - \bar{S}_h)^2}{|C_h|\left(|C_h| - 1\right)}},$$

- Small RMSD means community members have consistent tastes.

Comparison between two models

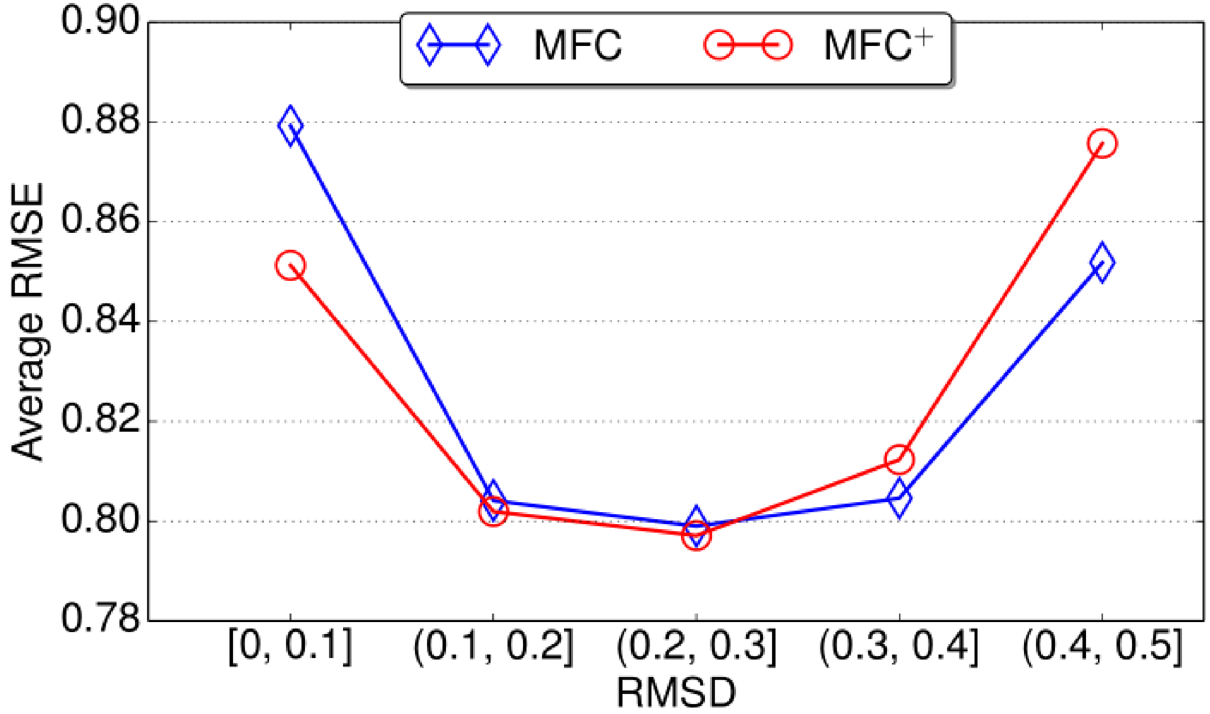


Figure 4.5: Performance of MFC and MFC$^+$ over different kinds of communities in Dianping when CPM is employed.

Outline

- Rating Based Social Recommender Systems in Retrospect
- Community Based Recommender Systems MFC & MFC^+
- Future Work

Future Work

- **Overlapping vs. Non-overlapping Community**

  We found that using overlapping community gives a better results.

- **Impact of different community detection approaches on MFC and MFC+**

  We found some prevalent community detection approaches give a very poor result on social recommender systems.

Future Work

- **Detected Community vs. Real Community**.

  Does detected community have a real meaning?

  We are working on testing our models on Douban dataset, which contains manually formed communities.

- **Get rid of community detection approaches.**

- **Apply social recommendation models in traditional recommender systems (e.g., implicit relationships like similar tastes and frequent interactions).**