Improving top-K recommendation with truster and trustee relationship in user trust network

Published in WWW 2016 (Short Paper)
Extended in Information Sciences (2016)

Chanyoung Park
Data mining Lab.

Goal of Recommender System

items

users

	A	B	C	D
A	5	?	?	3
B	4	?	?	2
C	?	1	3	1

Probabilistic Matrix Factorization (PMF)

- Ratings can be approximated from probabilistic methods.

1. Modeling rating variables

$$p(R|U,V,\sigma^2)=\prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|U_i^TV_j,\sigma^2)\right]^{I_{ij}}$$

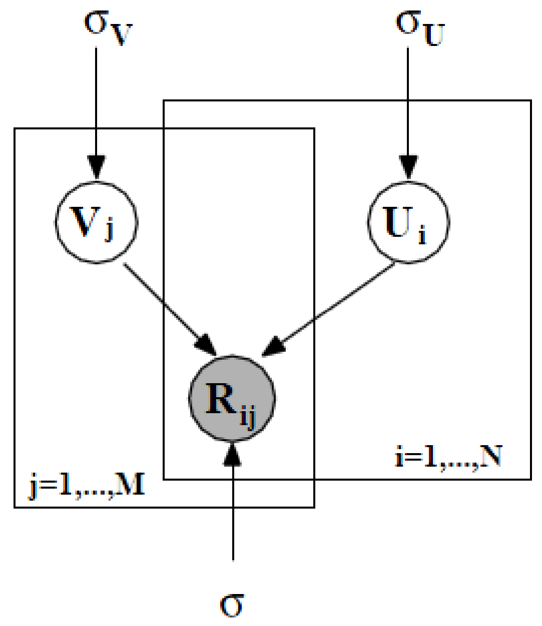
2. Modeling user and item variables

$$p(U|\sigma_U^2)=\prod_{i=1}^{N}\mathcal{N}(U_i|0,\sigma_U^2\mathbf{I}), \qquad p(V|\sigma_V^2)=\prod_{j=1}^{M}\mathcal{N}(V_j|0,\sigma_V^2\mathbf{I}).$$

3. Posterior probability over user and item variables

$$p(U,V|R,\sigma,\sigma_U,\sigma_V)\propto p(R|U,V,\sigma)p(U\big|\sigma_U^2)p(V|\sigma_V^2)$$

$$\begin{aligned}\ln p(U,V|R,\sigma^2,\sigma_V^2,\sigma_U^2) = &-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}(R_{ij}-U_i^TV_j)^2-\frac{1}{2\sigma_U^2}\sum_{i=1}^{N}U_i^TU_i-\frac{1}{2\sigma_V^2}\sum_{j=1}^{M}V_j^TV_j \\ &-\frac{1}{2}\left(\left(\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\right)\ln\sigma^2+ND\ln\sigma_U^2+MD\ln\sigma_V^2\right)+C, \qquad (3)\end{aligned}$$



<The graphical model of PMF>

Limitation of PMF

- They suffer from data sparsity problem
- What is *“Data sparsity problem”*?
  - Recommendation is hardly accurate due to lack of observations (i.e., ratings)
- To tackle the data sparsity problem, incorporating auxiliary information becomes important
  - Time related information
  - Textual data
  - **Social network relationships among users (Focus of this work)**

Motivation

- Most existing works exploit social information to reduce the rating prediction error, e.g., RMSE
- However, users are interested in seeing a list of top-k items rather than predicted ratings

	A	B	C	D	E
User1	5	?	?	3	?

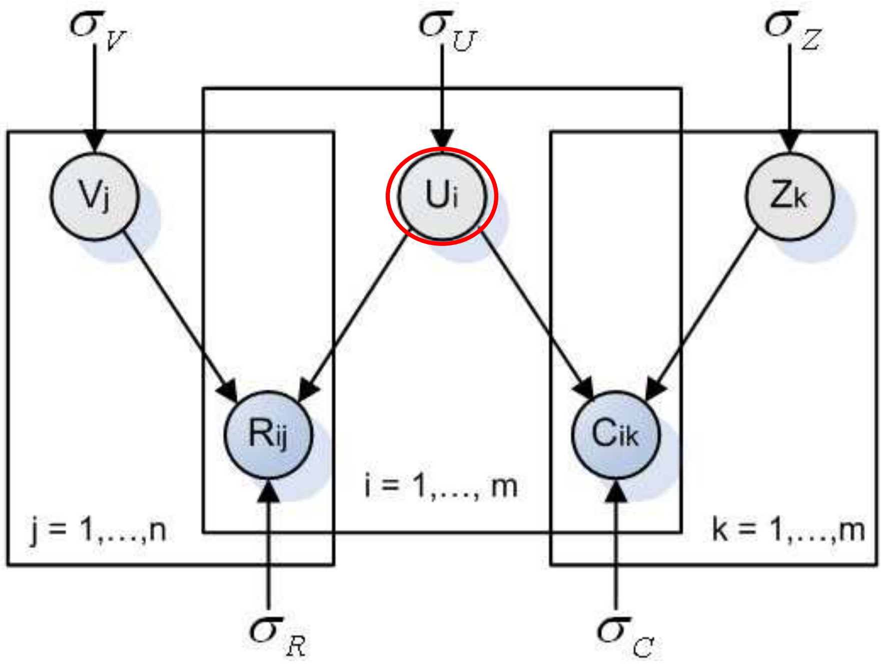
User1
Rank1: C
Rank2: E
Rank3: B

- Minimizing the rating prediction error does not always result in a better top-k list of items [Cremonesi et al. RecSys 2010]
- **Therefore, let’s focus on improving the result of top-k list of items using social network information.**

Related Work

Related Work 1 – Social Recommender System

1. SoRec [Ma et al., CIKM 09]
   - Matrix Co-Factorization technique

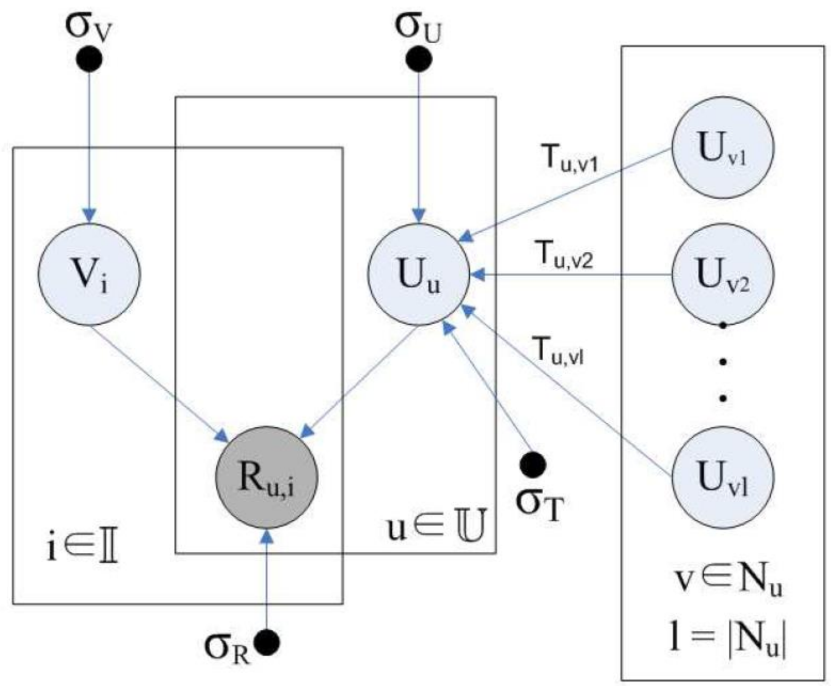


- Loss function

$$\mathcal{L}(R,C,U,V,Z) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(r_{ij} - g(U_i^T V_j))^2 + \frac{\lambda_C}{2}\sum_{i=1}^{m}\sum_{k=1}^{m} I_{ik}^{C}(c_{ik}^{*} - g(U_i^T Z_k))^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 + \frac{\lambda_Z}{2}\|Z\|_F^2, \quad (9)$$

Related Work 1 – Social Recommender System

2. SocialMF [Jamali et al., RecSys 10]
    - Models trust propagation



- Loss function

$$\mathcal{L}(R,T,U,V) = \frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} I^{R}_{u,i}(R_{u,i} - g(U_u^T V_i))^2 + \frac{\lambda_U}{2}\sum_{u=1}^{N} U_u^T U_u + \frac{\lambda_V}{2}\sum_{i=1}^{M} V_i^T V_i + \frac{\lambda_T}{2}\sum_{u=1}^{N}\Big((U_u - \sum_{v\in N_u} T_{u,v}U_v)^T (U_u - \sum_{v\in N_u} T_{u,v}U_v)\Big)$$

Related Work 1 – Social Recommender System

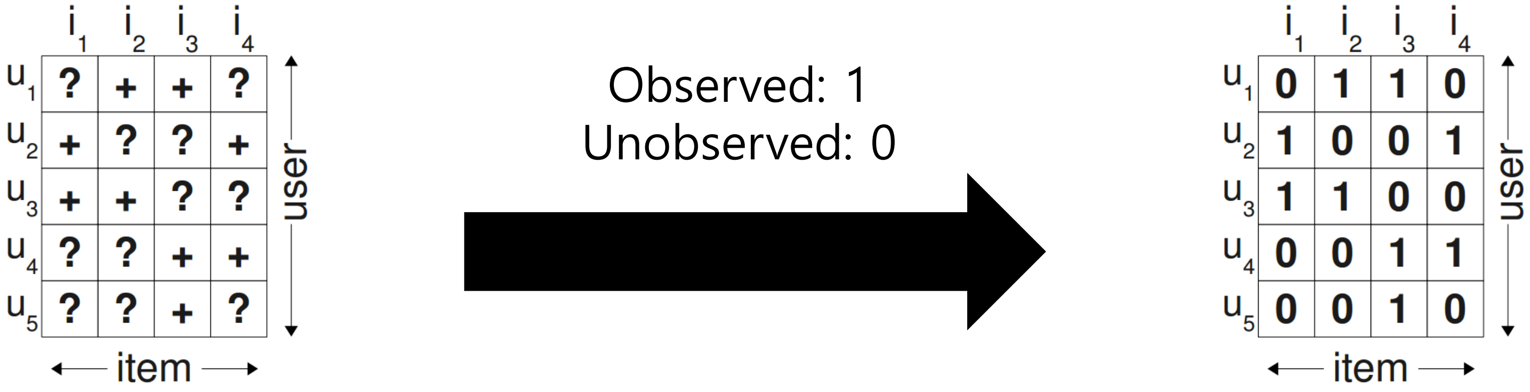
- Aforementioned social recommender systems mainly focus on minimizing the rating prediction error, e.g., MAE, RMSE
- Improving MAE and RMSE does not lead to improving top-k performance
- **Therefore, we focus on finding a better top-k list of items**

Related Work 2 – Top-k ranking RS

- Several approaches have been proposed for top-k recommendation
    - Can be cast as Learning-to-Rank (LTR) problem
- Learning-to-Rank
    - A Supervised ML method that directly builds a ranking list from training data
    - Pair-wise models
        - Learn users' relative preferences of each item pair
    - List-wise models
        - Directly predicts ranking list of items for each user based on the distance between the ground truth ranking list and the predicted list.

Related Work 2 – Top-k ranking RS

1. BPR [Rendle et al., UAI 09] (Pairwise method)

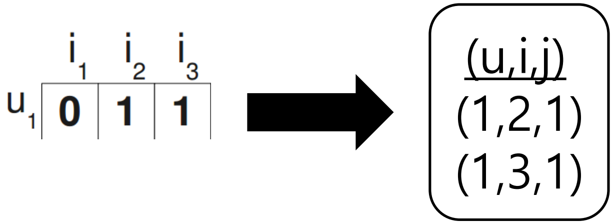


	i_1	i_2	i_3	i_4
u_1	?	+	+	?
u_2	+	?	?	+
u_3	+	+	?	?
u_4	?	?	+	+
u_5	?	?	+	?

Observed: 1
Unobserved: 0

	i_1	i_2	i_3	i_4
u_1	0	1	1	0
u_2	1	0	0	1
u_3	1	1	0	0
u_4	0	0	1	1
u_5	0	0	1	0

$$\mathrm{BPR}(\mathcal{D}_S) = \underset{\Theta}{\operatorname{argmax}} \sum_{(u,i,j)\in\mathcal{D}_S} \ln \sigma(\hat{s}_{u,i}(\Theta) - \hat{s}_{u,j}(\Theta)) - \lambda\|\Theta\|^2$$



	i_1	i_2	i_3
u_1	0	1	1

(u,i,j)
(1,2,1)
(1,3,1)

i: positive item
j: negative item

- ▶ $\mathcal{D}_S$ contains all pairs of positive and negative items for each user,
- ▶ $\hat{s}_{u,i}(\Theta)$ is the predicted score for user u and item i

Related Work 2 – Top-k ranking RS

2. ListRank [Shi et al., RecSys 10] (Listwise method)

- List-wise learning-to-rank algorithm + Matrix Factorization
- Top-one probability $P_{l_i}(R_{ij}) = \frac{\varphi(R_{ij})}{\sum_{k=1}^{K} \varphi(R_{ik})}$

Computes the probability of an item scored R_{ij} being ranked in top-1 position

- Loss function

$$L(U,V) = \sum_{i=1}^{M} \left\{ -\sum_{j=1}^{N} \overbrace{P_{l_i}(R_{ij}) \log P_{l_i}\left(g\left(U_i^T V_j\right)\right)}^{\text{Cross entropy}} \right\} + \frac{\lambda}{2}\left(\|U\|_F^2 + \|V\|_F^2\right)$$

$$= \sum_{i=1}^{M} \left\{ -\sum_{j=1}^{N} I_{ij} \frac{\exp(R_{ij})}{\sum_{k=1}^{N} I_{ik} \exp(R_{ik})} \log \frac{\exp\left(g\left(U_i^T V_j\right)\right)}{\sum_{k=1}^{N} I_{ik} \exp\left(g\left(U_i^T V_j\right)\right)} \right\} + \frac{\lambda}{2}\left(\|U\|_F^2 + \|V\|_F^2\right)$$

Preliminary: Defining permutation probability

- Probability of a permutation is defined with Plackett–Luce model

$$P(\pi \mid f) = \prod_{j=1}^{m} \frac{\varphi(f(x_{\pi(j)}))}{\sum_{k=j}^{m} \varphi(f(x_{\pi(k)}))}$$

$$P_{l_i}\left(R_{ij}\right) = \frac{\varphi\left(R_{ij}\right)}{\sum_{k=1}^{K} \varphi\left(R_{ik}\right)}$$

  - Example

$$P(\mathrm{ABC} \mid f) = \boxed{\frac{\varphi(f(\mathrm{A}))}{\varphi(f(\mathrm{A})) + \varphi(f(\mathrm{B})) + \varphi(f(\mathrm{A}))}} \cdot \boxed{\frac{\varphi(f(\mathrm{B}))}{\varphi(f(\mathrm{B})) + \varphi(f(\mathrm{C}))}} \cdot \boxed{\frac{\varphi(f(\mathrm{C}))}{\varphi(f(\mathrm{C}))}}$$

P(A ranked No.1)

P(B ranked No.2 | A ranked No.1)
= P(B ranked No.1)/(1- P(A ranked No.1))

P(C ranked No.3 | A ranked No.1, B ranked No.2)

Related Work 2 – Top-k ranking RS

- Although pair-wise models have shown substantial improvements in terms of top-k recommendation, they have issues with high computational complexity
- In this work, we adopt the **list-wise approach**

Related Work 3 –Top-k ranking **Social** RS

1. Sorank: Incorporating social information into learning to rank models for recommendation [Yao et al., WWW 2014]
   - Linearly combine a user's taste and her direct friends' taste

2. SBPR [Zhao et al., CIKM 14]
   - Social network integrated version of BPR [Rendle et al., UAI 09]
   - Optimize the top-k recommendation from relative ordering that can be extracted from purchase history or browsing history

Related Work 3 –Top-k ranking **Social** RS

- SoRank [Yao et al., WWW 2014] does not utilize other important information hidden in social network such as the structural information or follower-followee relationship

- SBPR [Zhao et al., CIKM 14] cannot handle numerical ratings directly

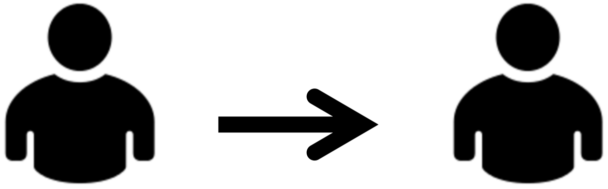
Proposed Method

Problem Definition

- $U = \{u_1, u_2, \dots, u_N\}$: Set of users
- $V = \{v_1, v_2, \dots, v_M\}$: Set of items
- $R = \left[r_{ij}\right]_{N \times M}$: Rating of u_i on v_j
- $S = [s_{ik}]_{N \times N}$: $s_{ik} = 1$, if u_i follows u_k $(s_{ik} \neq s_{ki})$
- <u>Problem</u>
  - Given: The observed rating matrix R and the trust matrix S
  - Goal: Recommend each user a list of unobserved items considering their personal preferences

Method: Modeling Rating

- Due to the asymmetry property ($s_{ik} \neq s_{ki}$), we map each user into two different latent vectors – **Follower** and **Followee**
- Assumption
  - When "$user\ A$" is given several choices of items, he asks the people he follows for their opinions about the items **(Follower role)**
  - The decision made by the "$user\ A$" will influence the people that follow "$user\ A$" **(Followee role)**



POSTECH

Method: Modeling Rating

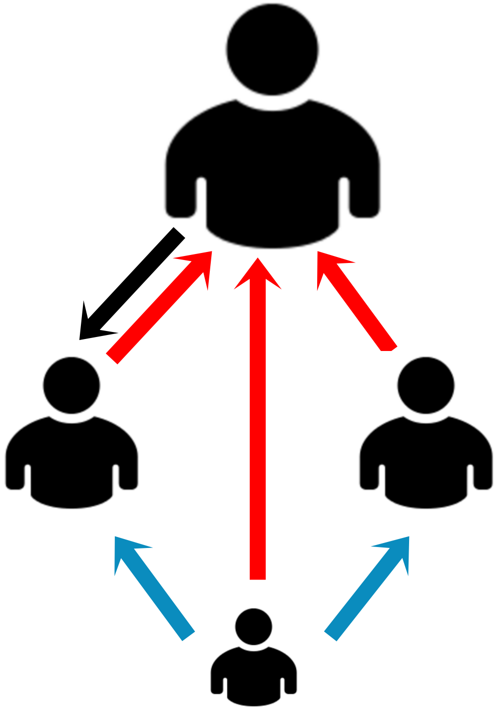
- Rating prediction

Follower + **Followee**

$$\hat{r}_{ij} = g(\mu + b_{u_i} + b_{v_j} + {q_j}^T(\alpha \underbrace{p_i + (1-\alpha) w_i}_{\textbf{User}} + |I_i|^{-\frac{1}{2}} \sum_{t \in I_i} y_t + |T_i|^{-\frac{1}{2}} \sum_{v \in T_i} x_v))$$

- b_{u_i}: User bias
- b_{v_j}: Item bias
- p_i: Follower latent vector
- w_i: Followee latent vector
- y_t: implicit influence of items rated by u_i
- x_v: implicit influence of users followed by u_i
- I_i: Set of items rated by u_i
- T_i: Set of users trusted by u_i

Method: Modeling Trust



- To reflect the structural information of trust network…
  - Adjust s_{ik} based on the degrees of nodes such that
    - Give lower weights to those who *follow* many users
    - Give higher weights to those who *are followed* by many users

$$s_{ik}^{*} = \sqrt{\frac{Indegree(v_k)}{Outdegree(v_i) + Indegree(v_k)}} \times s_{ik}$$

v_i: Node for u_i

Method: Modeling Trust

- Trust prediction

$$\hat{s}_{ik} = g(b_{p_i} + b_{w_k} + w_k^T p_i)$$

- ➢ b_{p_i}: Follower bias
- ➢ b_{w_k}: Followee bias
- ➢ p_i: Follower latent vector
- ➢ w_k: Followee latent vector

Method: Unified Model

- Final Loss function

$$\mathcal{L} = \overset{\text{Rating}}{\boxed{-\sum_{i=1}^{N}\sum_{j\in I_i} P_{l_i}(r_{ij}) \log P_{l_i}(\hat{r}_{ij})}} - \overset{\text{Trust}}{\boxed{\lambda_t \sum_{i=1}^{N}\sum_{k\in T_i} P_{l_i}(s^*_{ik}) \log P_{l_i}(\hat{s}_{ik})}}$$

$$+ \frac{\lambda_p}{2}||p_i||_F^2 + \frac{\lambda_w}{2}||w_i||_F^2 + \frac{\lambda_v}{2}||q_j||_F^2 + \frac{\lambda_c}{2}(\sum_{t\in I_i}||y_t||_F^2 + \sum_{v\in T_i}||x_v||_F^2)$$

$$+ \frac{\lambda_b}{2}(|||b_{u_i}||_F^2 + ||b_{v_j}||_F^2 + |b_{p_i}||_F^2 + ||b_{w_i}||_F^2)$$

Experiments

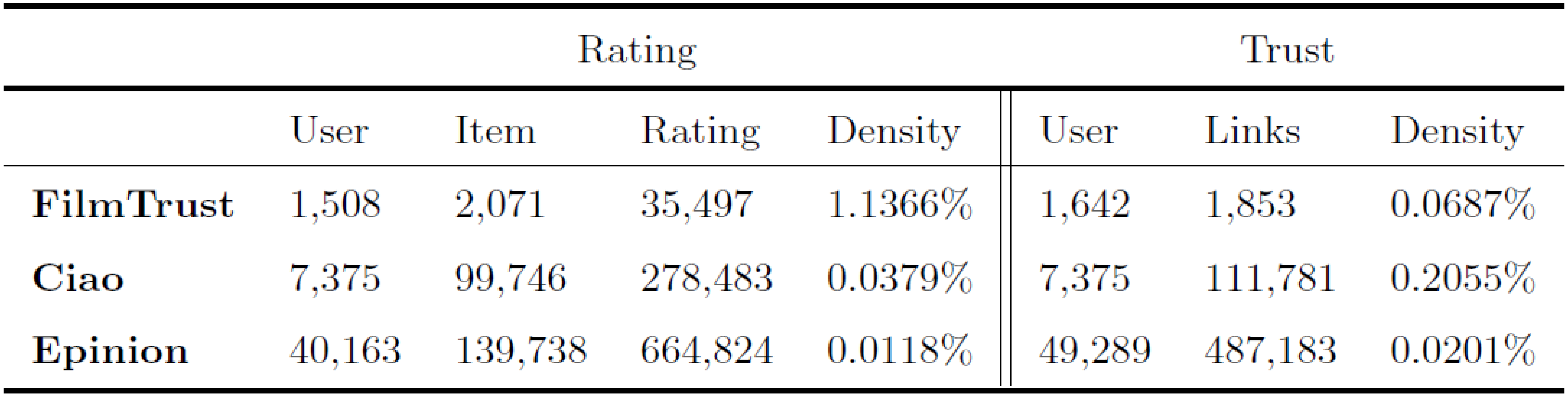
Experiments

- Questions to answer

1. How does TRecSo perform compared with other related competitors?
2. Does considering the social network structure enhance the performance of TRecSo?
3. How does the trade-off parameter of TRecSo affect the quality of top-k recommendation?

Experiments

- Data statistics



	Rating				Trust		
	User	Item	Rating	Density	User	Links	Density
FilmTrust	1,508	2,071	35,497	1.1366%	1,642	1,853	0.0687%
Ciao	7,375	99,746	278,483	0.0379%	7,375	111,781	0.2055%
Epinion	40,163	139,738	664,824	0.0118%	49,289	487,183	0.0201%

Experiments

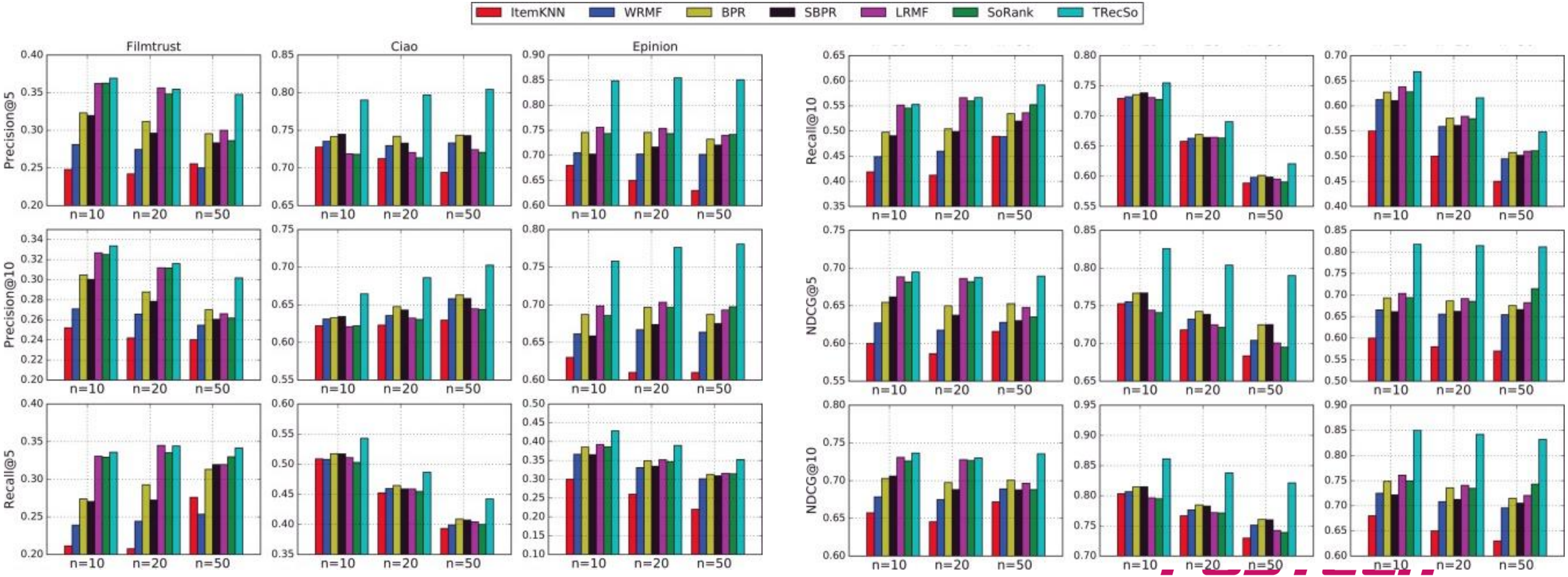
- Experiment protocol: Weak generalization
  - A widely used protocol for evaluating the performance of top-k recommender system
  - Evaluated by predicting the rank of unrated items for users known at training time
- We randomly select N=10, 20, 50 observed ratings for each user for training
- The model performance is tested on the remaining observed ratings

Experiments

- Competitors
  - Traditional CF method
    - ***ItemKNN:*** A traditional recommendation method based on similarity of items
  - Ratings-only-based LTR methods
    - ***WRMF***: A weighted matrix factorization algorithm with implicit feedback data
    - ***BPR***: An item recommendation algorithm based on pair-wise Learning-to-Rank strategy combined with matrix factorization.
    - ***ListRank***: A list-wise Learning-to-Rank method combined with matrix factorization
  - Social network-based LTR methods
    - ***SBPR****: An extended version of BPRMF by including social network information*
    - ***SoRank:*** A social network based list-wise Learning-to-Rank algorithm that linearly combines a users taste and her direct friends tastes in optimizing the top-k recommendation

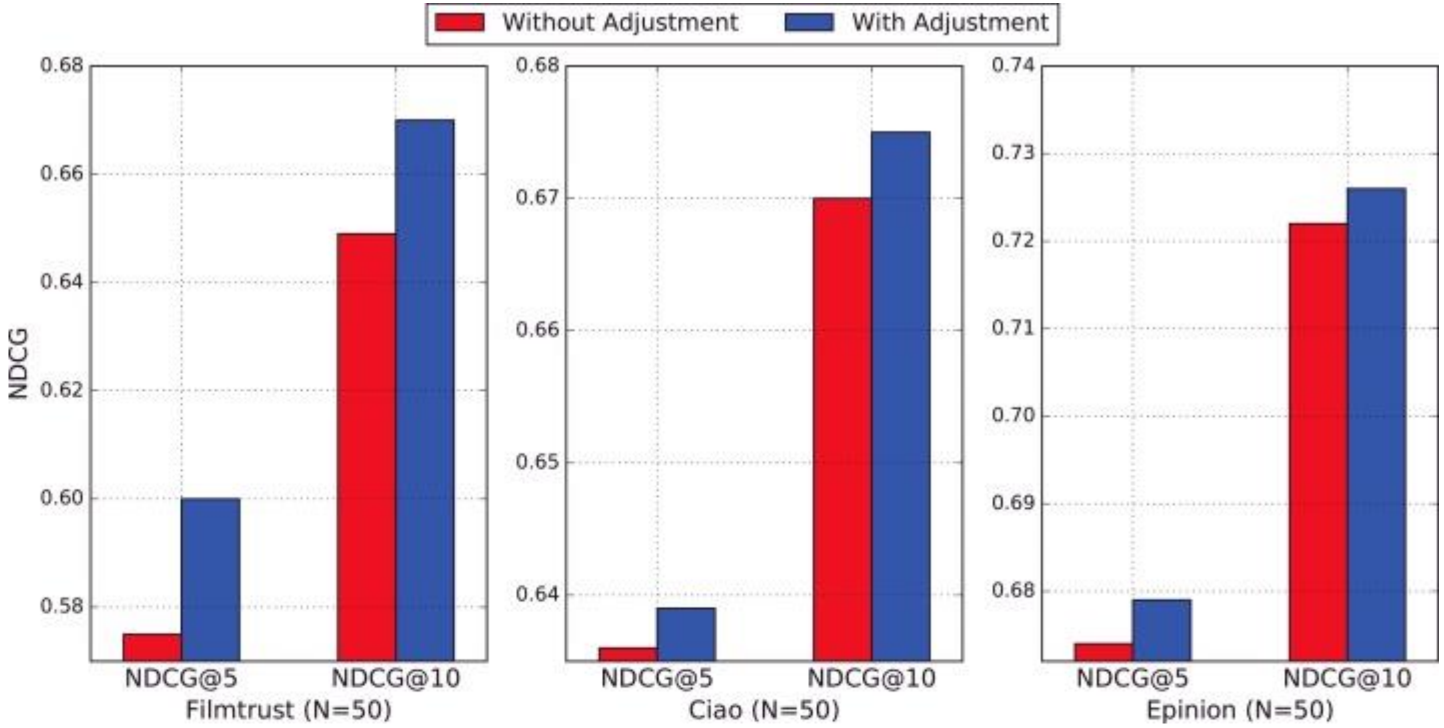
Experiments: Question #1

- Performance comparison



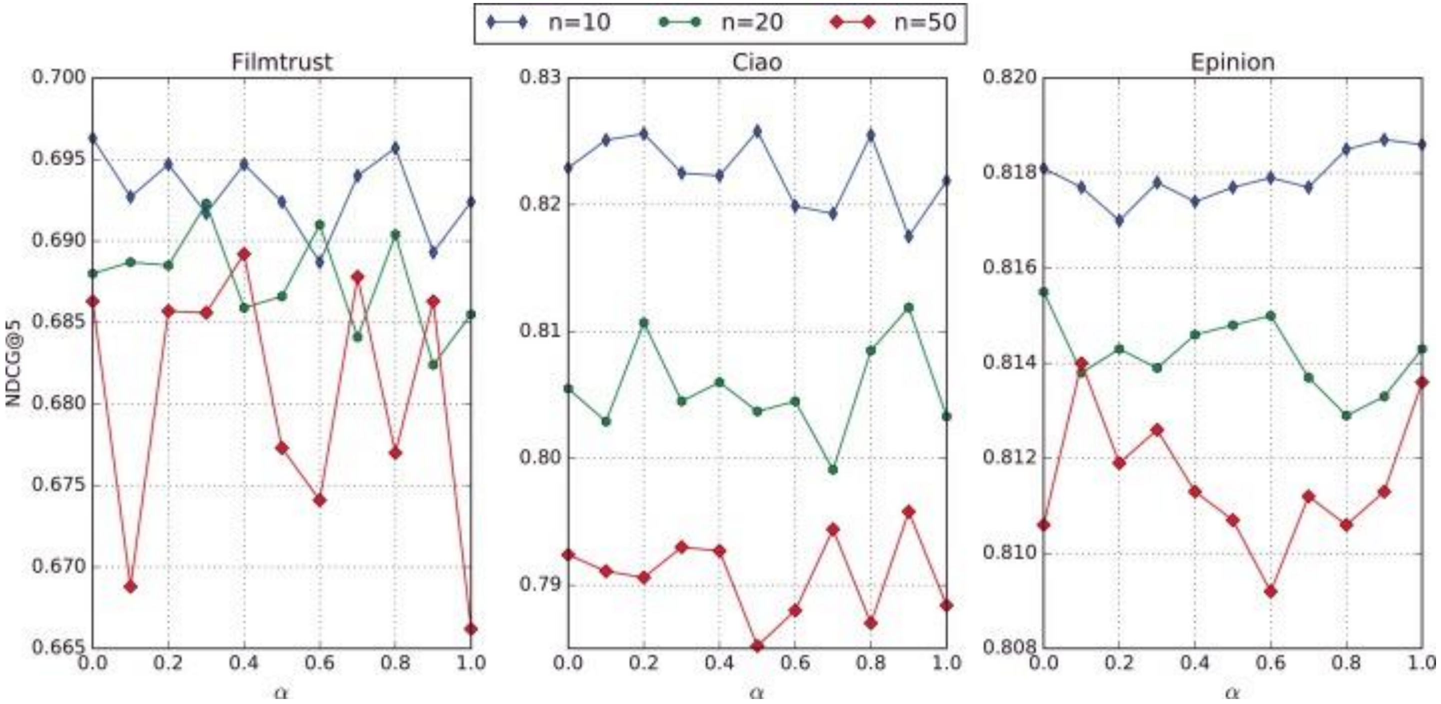
Experiments: Question #2

- Impact of considering graph structural information



Experiments: Question #3

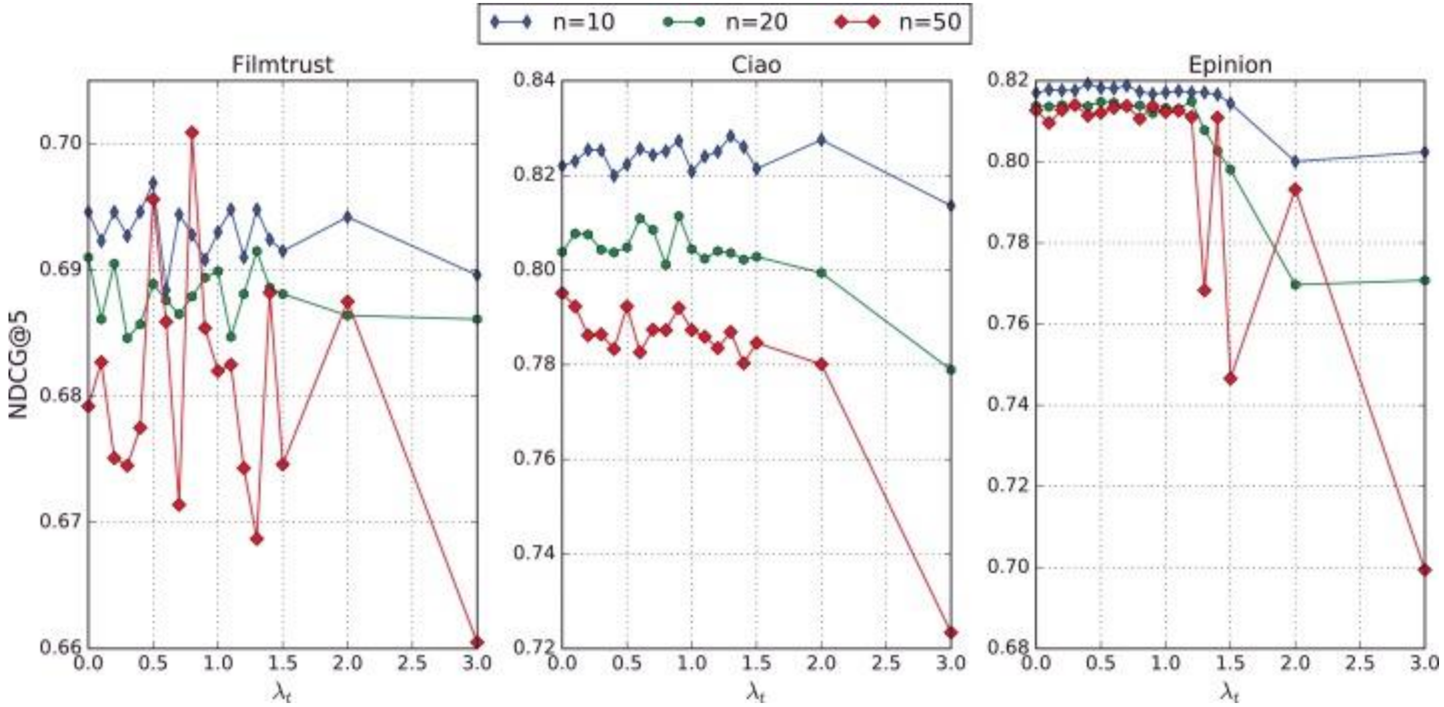
- Impact of trade-off parameters α
  - α: The parameter for balancing the relative importance of influence of follower and followee



- A proper value of α improves the recommendation quality

Experiments: Question #3

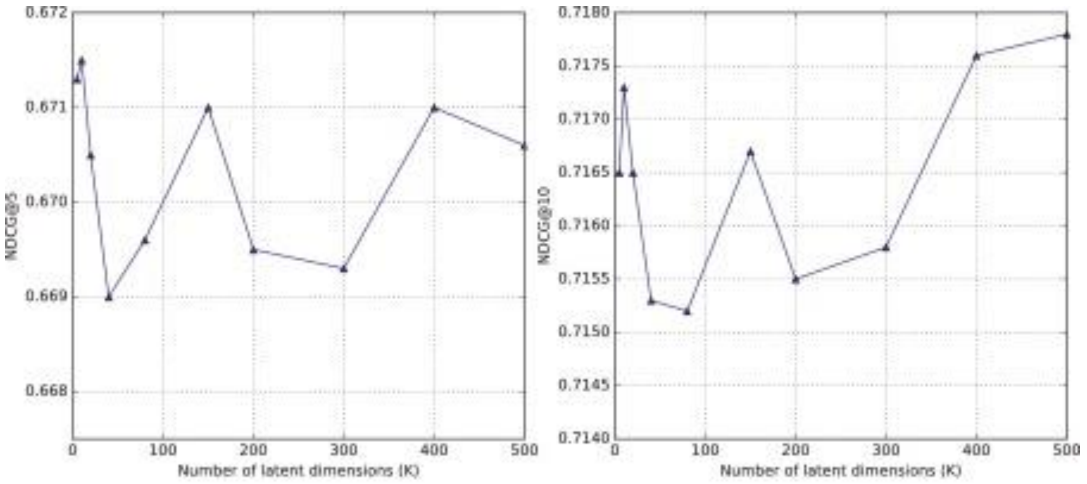
- Impact of trade-off parameters λ_t
  - λ_t: The parameter that controls the importance of trust regularization



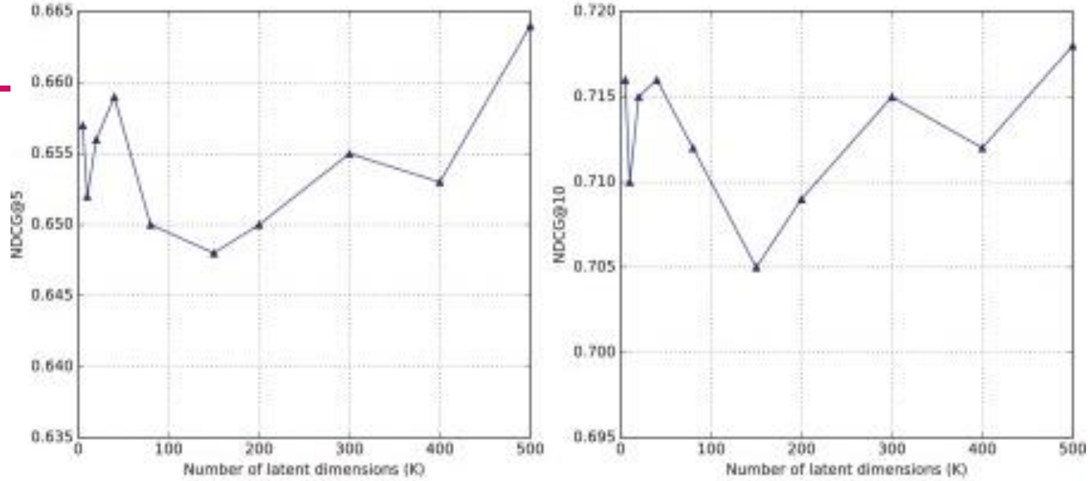
- Incorporating trust information ($\lambda_t > 0$) improves the recommendation

Experiments: Question #4

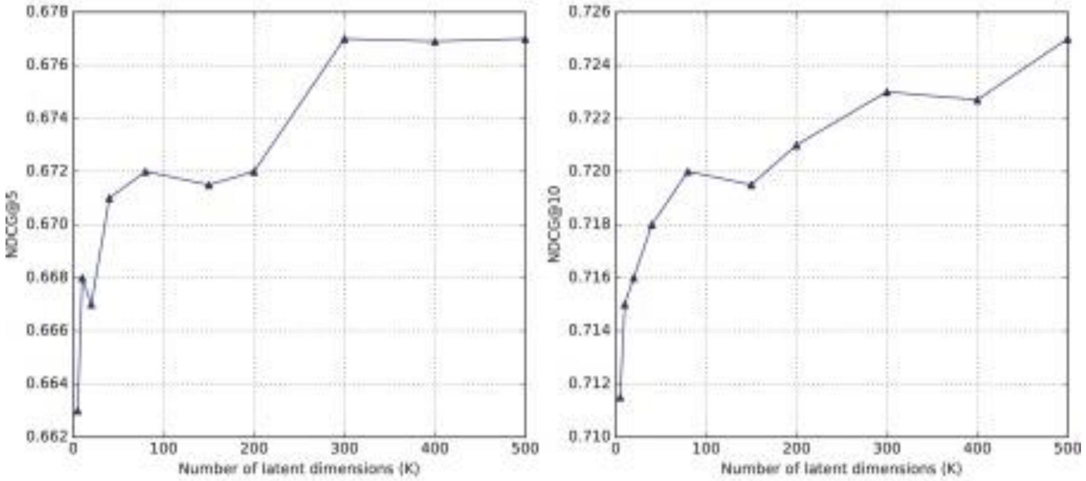
- Dimensionality analysis



(a) Filmtrust dataset (n=20)



(b) Ciao dataset (n=20)



(c) Epinion dataset (n=20)

Experiments: Question #4

- Dimensionality analysis
  - Generally, it is known that the performance of recommendation improves as the number of latent dimensions increases
  - Filmtrust / Ciao → No trend
  - Epinion → Desired trend
  - Reason
    - Each latent dimension represents the profile of user's interest and item's features
    - However, for datasets like Epinion (large number of users and items), the performance of recommendation improves as the number of latent dimensionality increases.
- Trade-off between performance and complexity
  - If the number of dimensions is too large, the complexity will significantly increase
  - Find a proper number of latent dimensions!

Conclusion

- This work proposes a novel MF based recommendation method that optimizes the top-k ranking prediction accuracy
  - Considered two roles of users as follower and followee
  - Considered the trust network information
- TRecSo significantly outperforms the state-of-the-art algorithms in the top-k ranking accuracy of recommendation

Reference

- [Cremonesi et al. RecSys 2010] Performance of recommender algorithms on top-n recommendation tasks
- [Jamali et al., RecSys 10]: A matrix factorization technique with trust propagation for recommendation in social networks
- [Ma et al., CIKM 09]: Sorec: social recommendation using probabilistic matrix factorization
- [Rendle et al., UAI 09]: Bpr: Bayesian personalized ranking from implicit feedback
- [Shi et al., RecSys 10]: List-wise learning to rank with matrix factorization for collaborative filtering
- [Tang et al., IJCAI 13]: Exploiting Local and Global Social Context for Recommendation
- [Zhao et al., CIKM 14]: Leveraging social connections to improve personalized ranking for collaborative filtering
- [Yao et al., WWW 2014] Sorank: incorporating social information into learning to rank models for recommendation,