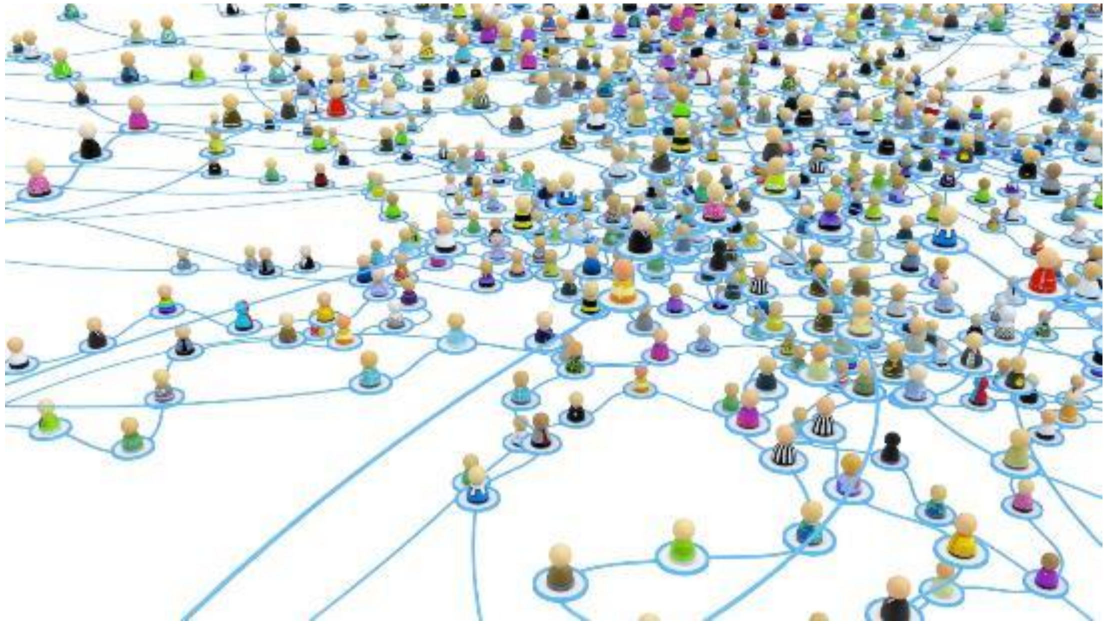


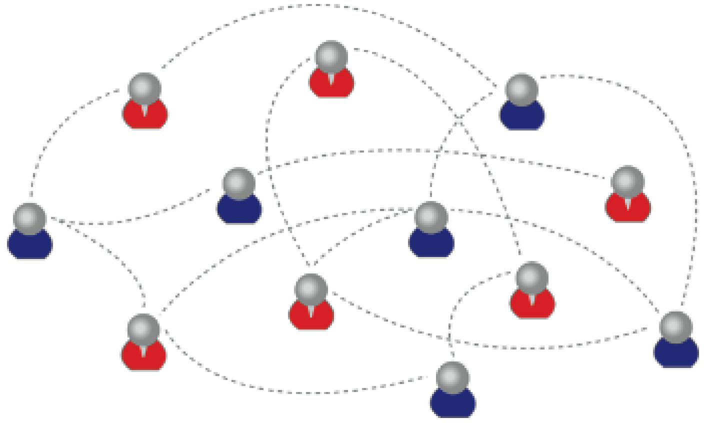
Social Regularization

SEMINAR RECOMMENDER SYSTEMS - 10.01.17

Maurício Camargo / Guilherme Holdack / Zafar Mahmood

Agenda

- Introduction to Social Recommendation
- Social Recommendation Methods
- SoRec: Social Recommendation Using Probabilistic Matrix Factorization
- SocialMF: A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks
- RSR: Recommended Systems With Social Regularization
- Conclusions and Comparison



1. Intro to Social Recommendation

Definition

- Traditional recommender systems assume that users are independent and identically distributed (i.i.d. assumption);
- However, online users are inherently connected via various types of relations such as friendships and trust relations;
- Users in social recommender systems are connected, providing social information.

1. Intro to Social Recommendation

Narrow x Broad Definition

- **Narrow Definition:** any recommendation with online social relations as an additional input, i.e., augmenting an existing recommendation engine with additional social signals.

- **Broad Definition:** recommender systems recommending any objects in social media domains such as items (the focus under the narrow definition), tags , people, and communities.

The **narrow definition** is used in the context of this presentation.

1. Intro to Social Recommendation

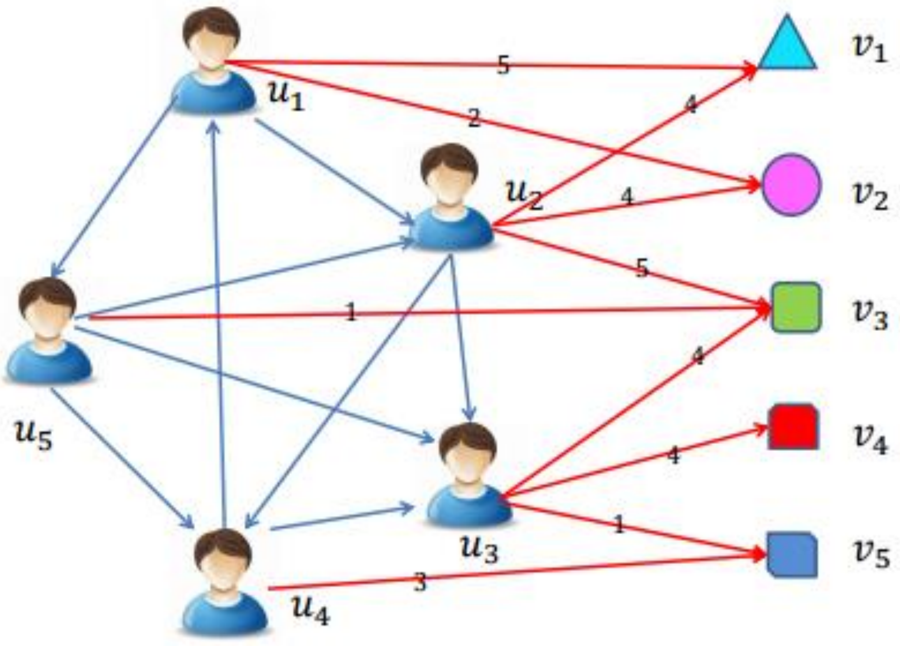
Reasons to use

- Connected users are more likely to share similar interests in topics than two randomly selected users;
- In the physical world, we usually ask suggestions from our friends (tend to be similar and also know our tastes);
- Provides an independent source of information about online users (specially useful on Cold Start);
- Exploiting social relations can potentially improve recommendation performance.

1. Intro to Social Recommendation

Representation

- In addition to the rating matrix in traditional recommender systems, there is also a second matrix to map the relations:



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

	u_1	u_2	u_3	u_4	u_5
u_1	0	1	0	0	1
u_2	0	0	1	1	0
u_3	0	0	0	0	0
u_4	1	0	1	0	0
u_5	0	1	1	1	0

2. Social Recommendation Methods

Overview of the Methods

- **Memory based**: for social recommendation, it takes both the rating information and social information to find similar users (ex: TidalTrust, MoleTrust, TrustWalker).

- **Model based**: uses matrix-factorization methods which also take into account the social relations. A unified framework can be stated as:

$$\min_{\mathbf{U},\mathbf{V},\Omega} \|\mathbf{W} \odot (\mathbf{R} - \mathbf{U}^\top \mathbf{V})\|_F^2 + \alpha \ \ Social(\mathbf{T}, \mathbf{S}, \Omega) \ \ + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 + \|\Omega\|_F^2)$$

2. Social Recommendation Methods

Model Based Methods

- **Co-factorization methods**: performs a co-factorization in the useritem matrix and the user-user social relation matrix by sharing the same user preference latent factor (Ex: SoRec and LOCABAL).

- **Ensemble methods**: a missing rating for a given user is predicted as a linear combination of ratings from the user and the social network (Ex: STE, mTrust).

- **Regularization methods**: For a given user, regularization methods force his preference to be closer to that of users in his social network. (Ex: SocialMF and Social Regularization).

SoRec: Social Recommendation Using Probabilistic Matrix Factorization

Maurício Camargo

SoRec

Motivation

Problems with current recommender systems:

- Ignores the social interactions or connections among users;
- Bad results on users who have made very few ratings or even none at all;
- Some existing approaches fail to handle very large datasets;

In reality, we always turn to friends we trust for movie, music or book recommendations;

Current Scenario

Collaborative Filtering

- Memory Based:

  - user-based and item-based approaches;

  - trust-based recommender systems – also use trust to calculate similarity (does not scale well).

- Model-based:

  - clustering model, aspect models and the latent factor model.

  - considers users independent and identically distributed

No model-based approach to deal with social relations.

Proposed Solution

SoRec (Social Recommendation):

- predict the missing values of the user-item by employing two different data sources.
- factorize the social network graph and user-item matrix simultaneously using $U^T Z$ and $U^T V$

U – low-dimensional user latent feature space

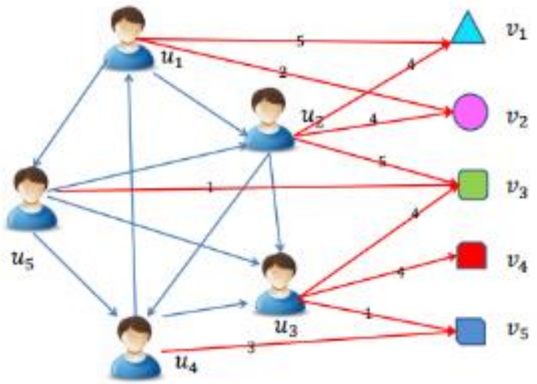
Z – factor matrix in the social network graph

V – low-dimensional item latent feature space

SoRec

How it works

1 - By analysing both the social relations and the ratings, we get two diferent tables:



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

	u_1	u_2	u_3	u_4	u_5
u_1	0	1	0	0	1
u_2	0	0	1	1	0
u_3	0	0	0	0	0
u_4	1	0	1	0	0
u_5	0	1	1	1	0

SoRec

How it works

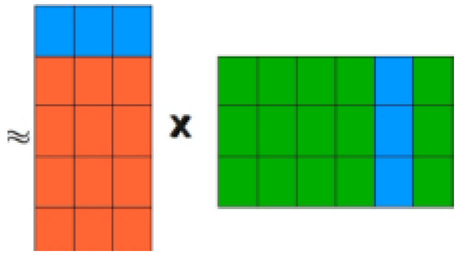
2 – Both resulting tables can be factorized into its latent features:

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

$U_1^T V$

	u_1	u_2	u_3	u_4	u_5
u_1	0	1	0	0	1
u_2	0	0	1	1	0
u_3	0	0	0	0	0
u_4	1	0	1	0	0
u_5	0	1	1	1	0

$U_2^T Z$



U_1 and ***U_2*** – low-dimensional user latent feature space

Z – factor matrix in the social network graph

V – low-dimensional item latent feature space

SoRec

How it works

3 – The trick is to force both factorizations to share the same U:

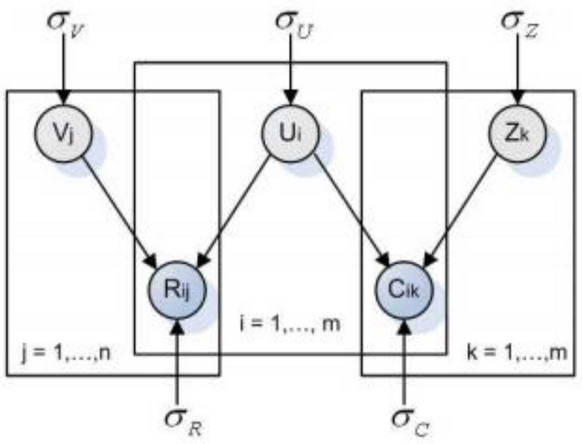
	v_1	v_2	v_3	v_4	v_5
u_1	5	?	2	?	?
u_2	4	4	5	?	?
u_3	?	?	4	4	1
u_4	?	?	?	?	3
u_5	?	?	1	?	?

U^TV

Same matrix

U^TZ

	u_1	u_2	u_3	u_4	u_5
u_1	0	1	0	0	1
u_2	0	0	1	1	0
u_3	0	0	0	0	0
u_4	1	0	1	0	0
u_5	0	1	1	1	0



$$p(C|U,Z,\sigma_C^2) = \prod_{i=1}^{m}\prod_{k=1}^{m} \mathcal{N}\left[\left(c_{ik}|g(U_i^T Z_k),\sigma_C^2\right)\right]^{I_{ik}^C}$$

$$p(R|U,V,\sigma_R^2) = \prod_{u=1}^{N}\prod_{i=1}^{M}\left[\mathcal{N}\left(R_{u,i}|g(U_u^T V_i),\sigma_r^2\right)\right]^{I_{u,i}^R}$$

U – low-dimensional user latent feature space

Z – factor matrix in the social network graph

V – low-dimensional item latent feature space

U will be influenced by the ***user x item*** ratings AND its ***social network.***

SoRec

How it works

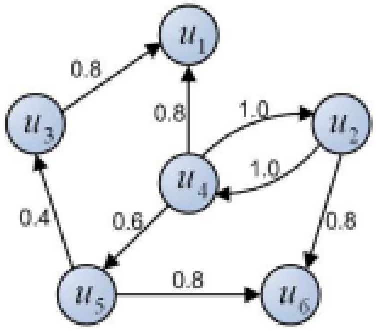
To improve the model:

- trust value should decrease if user i trusts lots of users;
- trust value should be increase if user k is trusted by lots of users.

$$c_{ik}^{*} = \sqrt{\frac{d^{-}(v_k)}{d^{+}(v_i) + d^{-}(v_k)}} \times c_{ik}$$

$d^{+}(v_i)$ = outdegree of node vi

$d^{-}(v_k)$ = indegree of node vk



The original equation becomes:

$$p(C|U, Z, \sigma_C^2) = \prod_{i=1}^{m} \prod_{j=1}^{n} \mathcal{N}\left[\left(c_{ik}^{*} | g(U_i^T Z_k), \sigma_C^2\right)\right]^{I_{ik}^C}$$

SoRec

How it works

For SoRec, the general equation:

$$\min_{\mathbf{U},\mathbf{V},\Omega} \|\mathbf{W} \odot (\mathbf{R} - \mathbf{U}^\top \mathbf{V})\|_F^2 + \alpha \;\; Social(\mathbf{T}, \mathbf{S}, \Omega) \;\; + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 + \|\Omega\|_F^2)$$

$\|\cdot\|_F^2$ - denotes the Frobenius norm

Becomes:

$$\min_{\mathbf{U},\mathbf{V},\mathbf{Z}} \|\mathbf{W} \odot (\mathbf{R} - \mathbf{U}^\top \mathbf{V})\|_F^2 + \alpha \sum_{i=1}^{n} \sum_{u_k \in \mathcal{N}_i} (\mathbf{S}_{ik} - \mathbf{u}_i^\top \mathbf{z}_k)^2 + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 + \|\mathbf{Z}\|_F^2)$$

$$Social(\mathbf{T}, \mathbf{S}, \Omega) = \min \sum_{i=1}^{n} \sum_{u_k \in \mathcal{N}_i} (\mathbf{S}_{ik} - \mathbf{u}_i^\top \mathbf{z}_k)^2$$

SoRec

How it works

Other notation:

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

In order to reduce the model complexity: λU = λV = λZ

The minimum can be found through gradient descent:

$$\frac{\partial \mathcal{L}}{\partial U_i} = \sum_{j=1}^{n} I_{ij}^{R} g'(U_i^T V_j)(g(U_i^T V_j) - r_{ij})V_j + \lambda_C \sum_{j=1}^{m} I_{ik}^{C} g'(U_i^T Z_k)(g(U_i^T Z_k) - c_{ik}^{*})Z_k + \lambda_U U_i$$

$$\frac{\partial \mathcal{L}}{\partial V_j} = \sum_{i=1}^{m} I_{ij}^{R} g'(U_i^T V_j)(g(U_i^T V_j) - r_{ij})U_i + \lambda_V V_j,$$

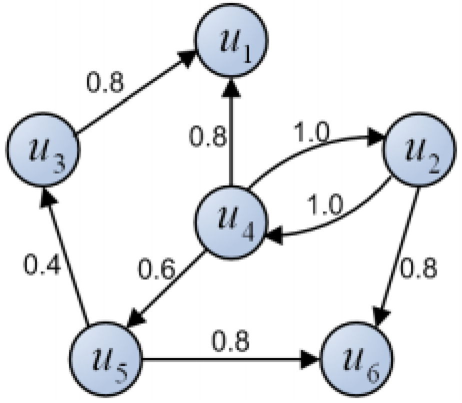
$$\frac{\partial \mathcal{L}}{\partial Z_k} = \lambda_C \sum_{i=1}^{m} I_{ik}^{C} g'(U_i^T Z_k)(g(U_i^T Z_k) - c_{ik}^{*})U_i + \lambda_Z Z_k$$

$$g'(x) = \exp(x)/(1+\exp(x))^2$$

derivative of the logistic function

SoRec

How it works



(a) Social Network Graph

	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

(b) User-Item 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

(c) Predicted User-Item Matrix

Even though user 4 does not rate any items, the approach still can predict reasonable ratings.

SoRec

How it works – pseudocode

Input: The rating information r, the social information c, the number of latent factors k, λ_C and λ (regularization parameters)

Output: The user preference matrix U and the item characteristic matrix V

1: Initialize U, V and Z randomly (with k factors)

2: while Not convergent do

3: Calculate ∂J ∂U , ∂J ∂V and ∂J ∂Z

4: Update U ← U − γu ∂J ∂U

$$\frac{\partial \mathcal{L}}{\partial U_i} = \sum_{j=1}^{n} I_{ij}^{R} g'(U_i^T V_j)(g(U_i^T V_j) - r_{ij})V_j + \lambda_C \sum_{j=1}^{m} I_{ik}^{C} g'(U_i^T Z_k)(g(U_i^T Z_k) - c_{ik}^{*})Z_k + \lambda_U U_i$$

5: Update V ← V − γv ∂J ∂V

$$\frac{\partial \mathcal{L}}{\partial V_j} = \sum_{i=1}^{m} I_{ij}^{R} g'(U_i^T V_j)(g(U_i^T V_j) - r_{ij})U_i + \lambda_V V_j,$$

6: Update Z ← Z − γz ∂J ∂Z

$$\frac{\partial \mathcal{L}}{\partial Z_k} = \lambda_C \sum_{i=1}^{m} I_{ik}^{C} g'(U_i^T Z_k)(g(U_i^T Z_k) - c_{ik}^{*})U_i + \lambda_Z Z_k$$

7: Evaluate LossFunction

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

8: end while

Complexity Analysis

$$\mathcal{L} = O(\rho_R l + \rho_C l)$$

Where ρ_R and ρ_C are the numbers of nonzero entries in matrices R and C.

$$\frac{\partial \mathcal{L}}{\partial U} = O(\rho_R l + \rho_C l)$$

$$\frac{\partial \mathcal{L}}{\partial V} = O(\rho_R l)$$

$$\frac{\partial \mathcal{L}}{\partial Z} = O(\rho_C l)$$

Total computational complexity in one iteration is: $O(\rho_R l + \rho_C l)$

Computational time of the method is linear with respect to the number of observations in the two sparse matrices. Thus, the approach can scale on large datasets.

Experimental Analysis

Epinions was selected as the data source

Epinions

- well known knowledge sharing site and review site.
- Users submit their opinions on topics such as products, companies, movies, or reviews issued by other users.
- Users can also assign products or reviews integer ratings from 1 to 5.
- Members maintain a “trust” and a “block (distrust)” list
- 40,163 users who have rated at least one of a total of 139,529 different items. The total number of reviews is 664,824

Density = $\frac{664824}{40163 \times 139529} = 0.01186\%.$ (very sparce)

Experimental Analysis

Epinions was select as the data source

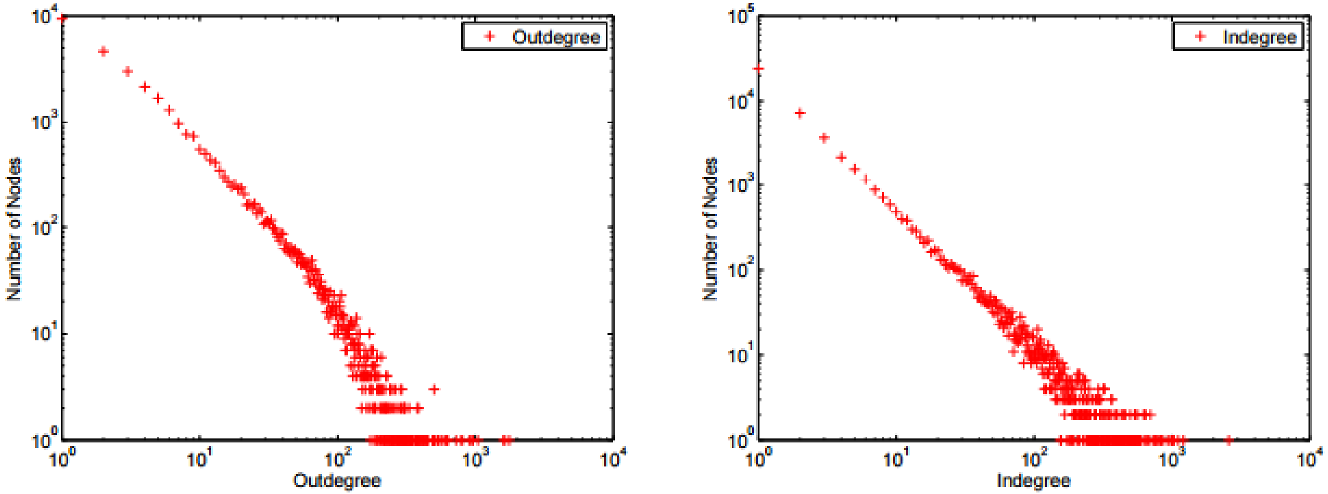


Figure 3: Degree Distribution of User Social Network

Statistics of User-Item Rating Matrix of Epinions

Statistics	User	Item
Min. Num. of Rated	1	1
Max. Num. of Rated	1022	2018
Avg. Num. of Rated	16.55	4.76

SoRec

Experimental Analysis

Comparison to other methods

Table 2: MAE comparison with other approaches (A smaller MAE value means a better performance)

Training Data	Dimensionality = 5				Dimensionality = 10			
	MMMF	PMF	CPMF	SoRec	MMMF	PMF	CPMF	SoRec
99%	1.0008	0.9971	0.9842	**0.9018**	0.9916	0.9885	0.9746	**0.8932**
80%	1.0371	1.0277	0.9998	**0.9321**	1.0275	1.0182	0.9923	**0.9240**
50%	1.1147	1.0972	1.0747	**0.9838**	1.1012	1.0857	1.0632	**0.9751**
20%	1.2532	1.2397	1.1981	**1.1069**	1.2413	1.2276	1.1864	**1.0944**

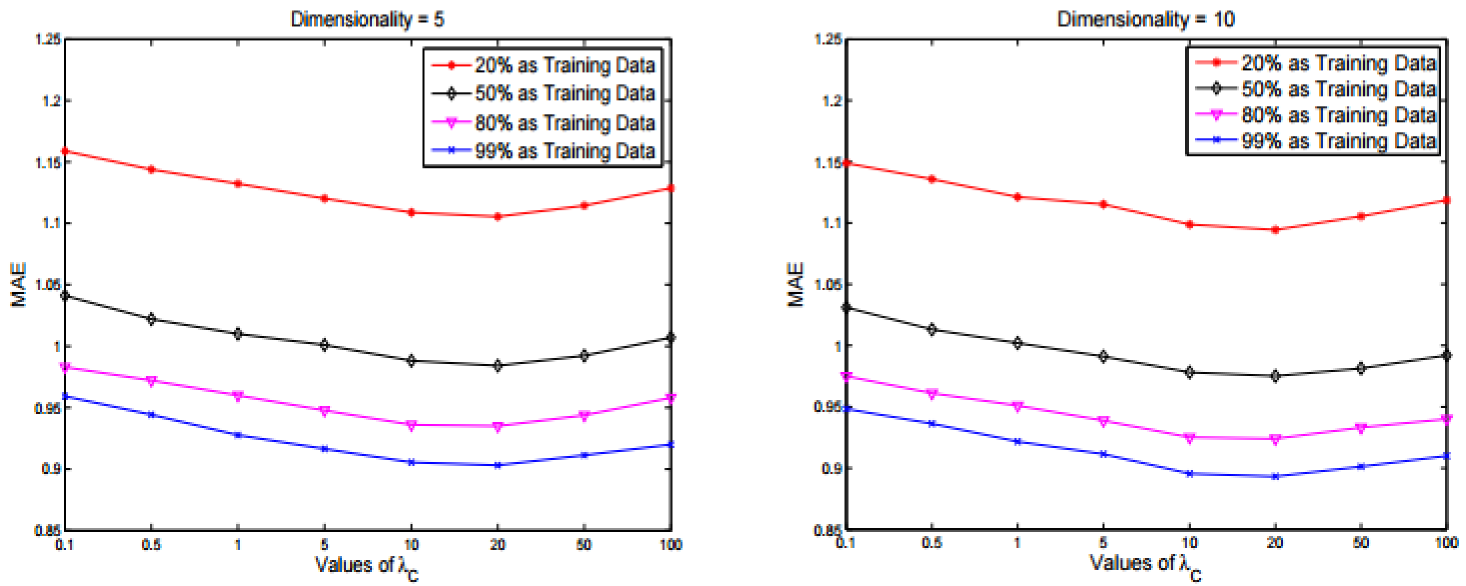
MMF - Maximum Margin Matrix Factorization
PMF - Probabilistic Matrix Factorization
CPMF - Constrained Probabilistic Matrix Factorization

$$MAE = \frac{\sum_{i,j} |r_{i,j} - \widehat{r}_{i,j}|}{N},$$

On average, the approach improves the accuracy by 11.01%, 9.98%, and 7.82% relative to MMMF, PMF and CPMF, respectively.

Experimental Analysis

Impact of λ_C

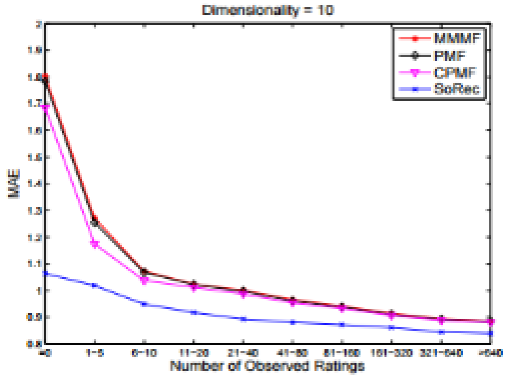


(a) Dimensionality=5 (b) Dimensionality=10

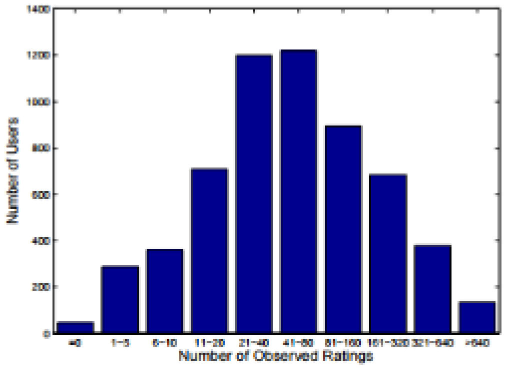
Figure 4: Impact of Parameter λ_C

Experimental Analysis

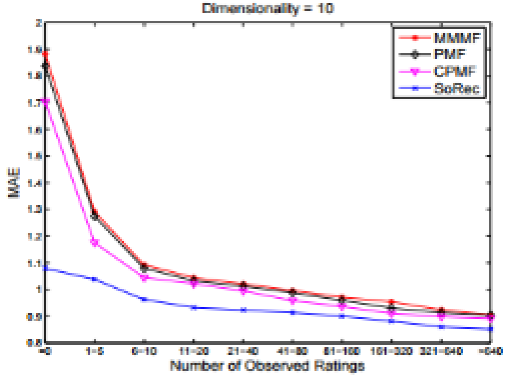
Performance on Different Users

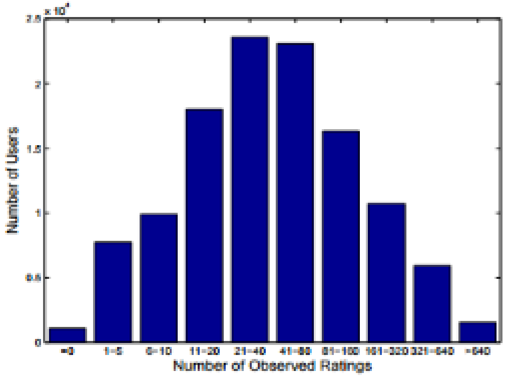


(a) Performance Comparison on Different User Rating Scales (99% as Training Data)



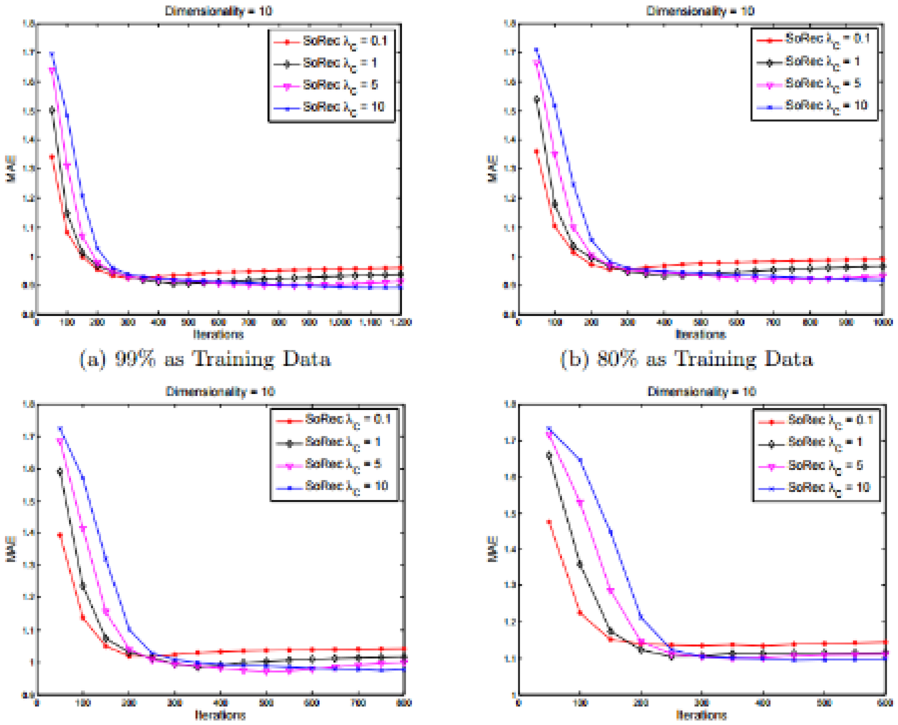
(b) Distribution of Testing Data (99% as Training Data)





Experimental Analysis

Efficiency Analysis



(a) 99% as Training Data

(b) 80% as Training Data

(c) 50% as Training Data

(d) 20% as Training Data

Figure 6: Efficiency Analysis

Conclusion and Future Work

Conclusion

- Experimental results: the approach outperforms the other state-of-the-art collaborative filtering algorithms.
- Complexity analysis: it is scalable to very large datasets.
- Can also be used to predict connections on social network.

Future Work:

- Investigate whether the distrust information is useful to increase the prediction quality, and how to incorporate it.
- Consider the diffusion process between users.

References

Ma, Hao, et al. "Sorec: social recommendation using probabilistic matrix factorization." Proceedings of the 17th ACM conference on Information and knowledge management. ACM, 2008.

Social Recommendation: A Review Jiliang Tang · Xia Hu · Huan Liu
https://pdfs.semanticscholar.org/fff4/4f028044dd6ee79b7c9c26a90a23dc8d4438.pdf

MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS
https://datajobs.com/data-science-repo/Recommender-Systems-%5BNetflix%5D.pdf

P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. In Proceedings of CoopIS/DOA/ODBASE, pages 492–508, 2004.

A MATRIX FACTORIZATION TECHNIQUE WITH TRUST PROPAGATION FOR RECOMMENDATION IN SOCIAL NETWORKS.

Social MF

INCORPORATING TRUST PROPAGATION

OUTLINE

- DEFINITION
- RELATED WORK
- PSEUDOCODE
- DATASETS
- EXPERIMENTS
- CONCLUSION + FUTURE WORK

OUTLINE

what is it?

DEFINITION

RELATED WORK

PSEUDOCODE

DATASETS

EXPERIMENTS

CONCLUSION + FUTURE WORK

OUTLINE

DEFINITION

RELATED WORK

who are the competitors ?

PSEUDOCODE

DATASETS

EXPERIMENTS

CONCLUSION + FUTURE WORK

OUTLINE

DEFINITION

RELATED WORK

PSEUDOCODE

DATASETS

EXPERIMENTS

CONCLUSION + FUTURE WORK

Let's have a look on it!

OUTLINE

DEFINITION

RELATED WORK

PSEUDOCODE

DATASETS

EXPERIMENTS

CONCLUSION + FUTURE WORK

which data are we dealing with ?

OUTLINE

DEFINITION

RELATED WORK

PSEUDOCODE

DATASETS

EXPERIMENTS

CONCLUSION + FUTURE WORK

how does
it behave ?

OUTLINE

DEFINITION

RELATED WORK

PSEUDOCODE

DATASETS

EXPERIMENTS

CONCLUSION + FUTURE WORK

where to go from here?

quick reminder

MATRIX FACTORIZATION

MATRIX FACTORIZATION

ONE OF THE MOST COMMON TECHNIQUES FOR MODEL BASED RECOMMENDATION.

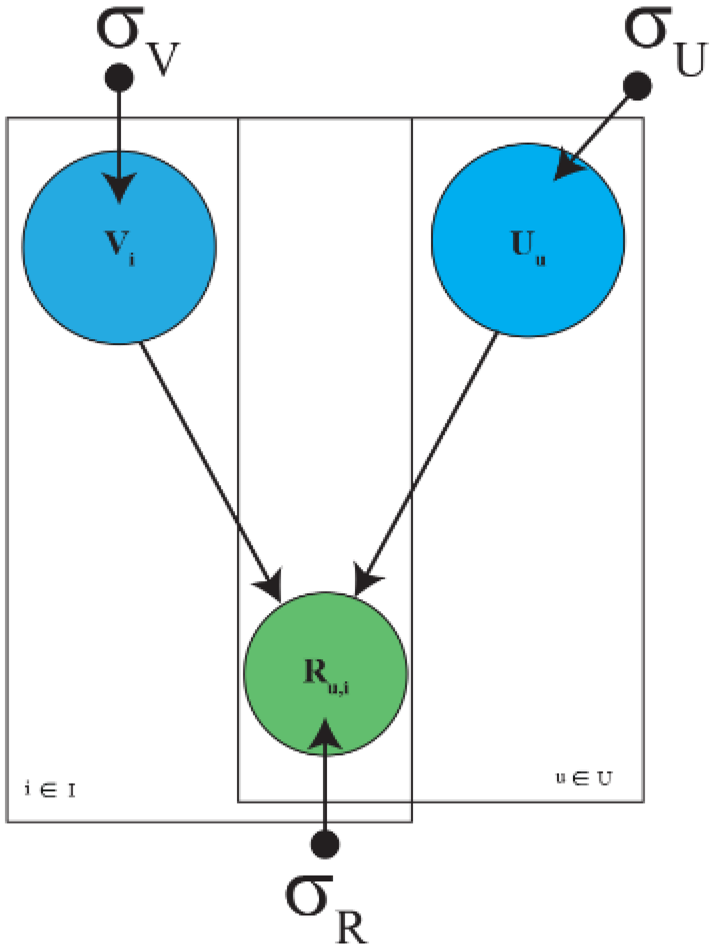
MATRIX FACTORIZATION

ONE OF THE MOST COMMON TECHNIQUES FOR MODEL BASED RECOMMENDATION.

LEARNS LATENT FEATURES FOR BOTH USERS AND ITEMS.

MATRIX FACTORIZATION

model and conditional probability



$$p(R|U,V,\sigma_R^2) = \prod_{u=1}^{N}\prod_{i=1}^{M}\left[\mathcal{N}\left(R_{u,i}|g(U_u^T V_i),\sigma_r^2\right)\right]^{I_{u,i}^R}$$

EACH RATING IS NORMALLY DISTRIBUTED AROUND THE PRODUCT OF BOTH, USERS AND ITEMS, FEATURE VECTORS - CONSIDERING STANDARD DEVIATION.

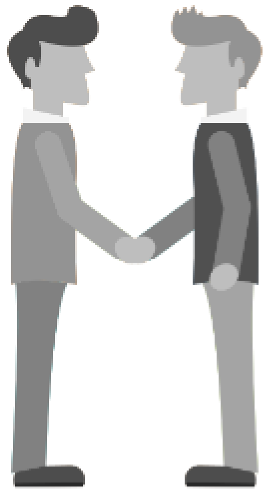

social relations
WHY PROPAGATING TRUST

WHAT IS IT ALL ABOUT?

direct neighbours

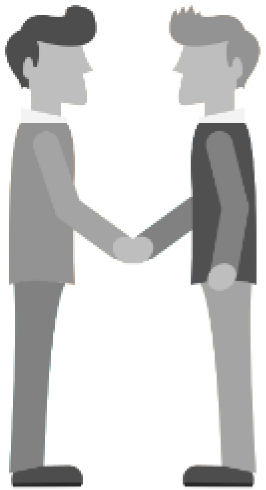
WHAT IS IT ALL ABOUT?

social influence



WHAT IS IT ALL ABOUT?

social relations



related work

PAPERS "COVERING" SAME TOPIC

RELATED WORK

- TIDAL TRUST
- MOLE TRUST
- ADVOGATO
- TRUSTWALKER
- SOCIAL TRUST ENSEMBLER - STE

RELATED WORK

- TIDAL TRUST
- MOLE TRUST
- ADVOGATO
- TRUSTWALKER
- SOCIAL TRUST ENSEMBLER - STE

FINDS ALL RATERS WITH THE SHORTEST PATH DISTANCE FROM THE SOURCE USER AND AGGREGATES THEIR RATINGS WEIGHTED BY THE TRUST BETWEEN THE SOURCE USER AND THESE RATERS.

RELATED WORK

- TIDAL TRUST
- MOLE TRUST
- ADVOGATO
- TRUSTWALKER
- SOCIAL TRUST ENSEMBLER - STE

SIMILAR TO TIDALTRUST, BUT RECEIVES A PARAMETER CALLED MAXIMUM-DEPTH. THIS WAY, ONLY RATERS CONNECTED UP TO A MAXIMUM DEGREE ARE CONSIDERED.

RELATED WORK

- TIDAL TRUST
- MOLE TRUST
- ADVOGATO
- TRUSTWALKER
- SOCIAL TRUST ENSEMBLER - STE

RECEIVES AN INTEGER INPUT, WHICH IS THE NUMBER OF MEMBERS TO TRUST. THIS NUMBER IS INDEPENDENT OF USERS OR ITEMS, SO IT IS NOT AN APPROPRIATE APPROACH FOR TRUST-BASED RECOMMENDATION.

RELATED WORK

TIDAL TRUST

MOLE TRUST

ADVOGATO

TRUSTWALKER

SOCIAL TRUST ENSEMBLER - STE

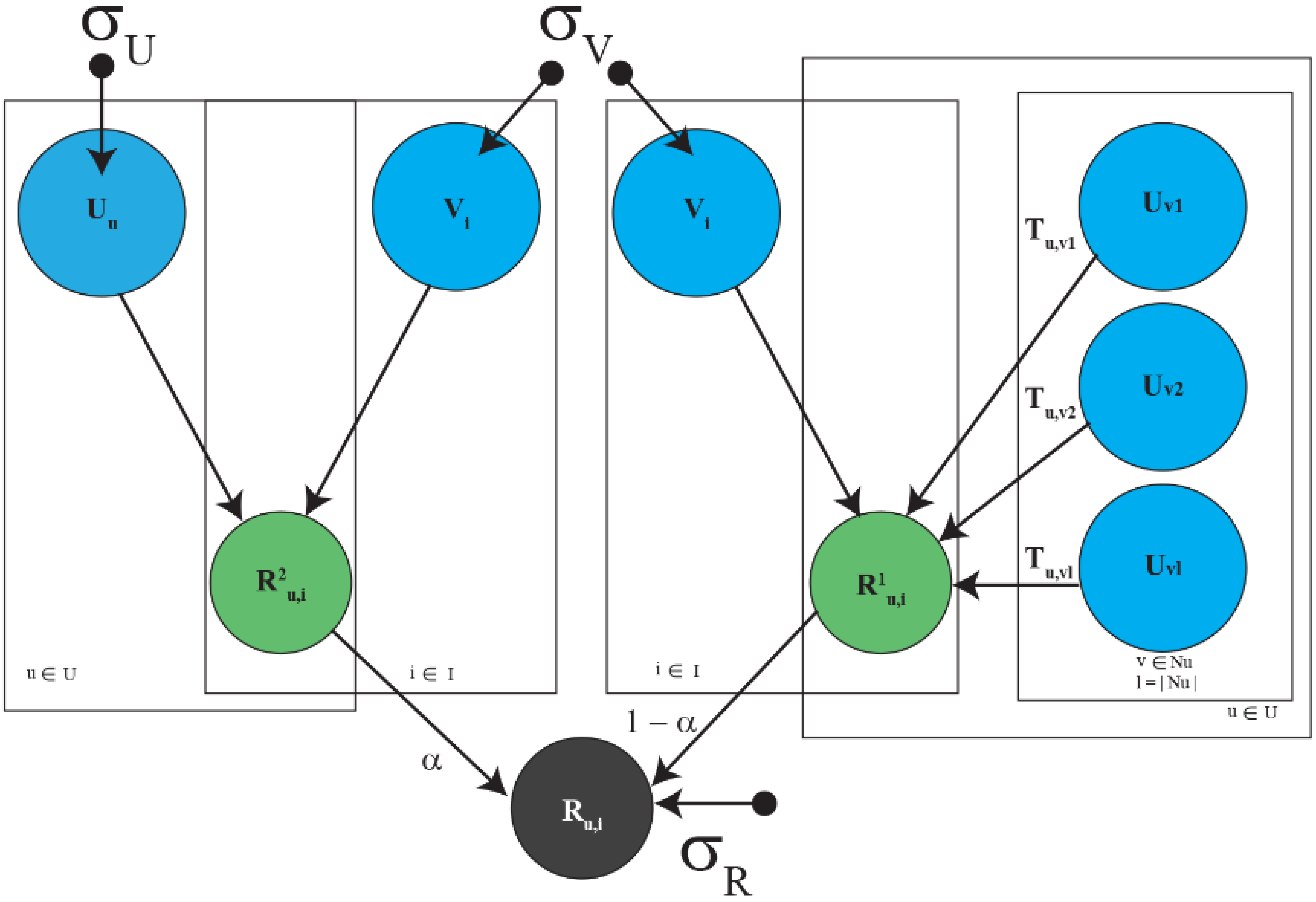
FOCUSES ON TRUST-BASED AND ITEM-BASED RECOMMENDATIONS. THERE IS A PROBABILITY OF CONSIDERING THE RATING OF A SIMILAR ITEM INSTEAD OF THE RATING FOR THE TARGET ITEM ITSELF, DEPENDING ON THE LENGTH OF THE WALK.

RELATED WORK

TIDAL TRUST
MOLE TRUST
ADVOGATO
TRUSTWALKER
SOCIAL TRUST ENSEMBLER - STE

PARTNER USED AS COMPARISON TO THIS PAPER. THIS MODEL AFFECTS THE RATINGS OF A GIVEN USER, MAKING USE OF THE FEATURE VECTORS OF THE DIRECT NEIGHBOURS; IT DOES NOT HANDLE TRUST PROPAGATION, AS IT DOES NOT AFFECT THE FEATURE VECTORS OF THE TARGET USER.

OR...



$$\hat{R}_{u,i} = g(\alpha U_u^T V_i \ + \ (1-\alpha) \sum_{v \in N_u} T_{u,v} U_v^T V_i)$$

Pseudocode

LET'S ANALYSE IT AND THEN... SEE SOME CODING..!

INCORPORATING TRUST

THE BEHAVIOUR OF A GIVEN USER IS AFFECTED BY THE INFLUENCE OF HIS DIRECT NEIGHBOURS.

INCORPORATING TRUST

THE BEHAVIOUR OF A GIVEN USER IS AFFECTED BY THE INFLUENCE OF HIS DIRECT NEIGHBOURS.

social influence !

INCORPORATING TRUST

THE BEHAVIOUR OF A GIVEN USER IS AFFECTED BY THE INFLUENCE OF HIS DIRECT NEIGHBOURS.

THE PROPOSED EQUATION:

$$\widehat{U}_u = \frac{\sum_{v \in N_u} T_{u,v} U_v}{\sum_{v \in N_u} T_{u,v}}$$

INCORPORATING TRUST

THE BEHAVIOUR OF A GIVEN USER IS AFFECTED BY THE INFLUENCE OF HIS DIRECT NEIGHBOURS.

THE PROPOSED EQUATION:

$$\widehat{U}_u = \frac{\sum_{v \in N_u} T_{u,v} U_v}{\sum_{v \in N_u} T_{u,v}}$$

all social networks analysed in this paper are binary, so: $T_{u,v}$ can only be zero or one!

AND, HAVING ALL ROWS OF THE TRUST MATRIX NORMALIZED:

THE SUM OF ALL "TRUSTS" IN A ROW WOULD BE: $\sum_{v=1}^{N} T_{u,v} = 1$

THIS WAY:

$$\widehat{U}_u = \frac{\sum_{v \in N_u} T_{u,v} U_v}{\sum_{v \in N_u} T_{u,v}}$$

THE SUM OF ALL "TRUSTS" IN A ROW WOULD BE: $\sum_{v=1}^{N} T_{u,v} = 1$

THIS WAY:

$$\hat{U}_u = \frac{\sum_{v \in N_u} T_{u,v} U_v}{\sum_{v \in N_u} T_{u,v}}$$

becomes

$$\hat{U}_u = \sum_{v \in N_u} T_{u,v} U_v$$

SO IT BECOMES EASY TO CONCLUDE, THAT THE ESTIMATE OF THE LATENT FEATURES VECTOR OF A GIVEN USER IS NOTHING MORE THAN THE WEIGHTED AVERAGE OF THE LATENT FEATURE VECTORS OF HIS DIRECT NEIGHBOURS!

THE ESTIMATED FEATURE VECTOR OF A GIVEN USER CAN BE INFERRED THIS WAY:

$$\begin{pmatrix} \widehat{U}_{u,1} \\ \widehat{U}_{u,2} \\ \ldots \\ \widehat{U}_{u,K} \end{pmatrix} = \begin{pmatrix} U_{1,1} & U_{2,1} & \ldots & U_{N,1} \\ U_{1,2} & U_{2,2} & \ldots & U_{N,2} \\ \ldots & .. & \ldots & \ldots \\ U_{1,K} & U_{2,K} & \ldots & U_{N,K} \end{pmatrix} \begin{pmatrix} T_{u,1} \\ T_{u,2} \\ \ldots \\ T_{u,N} \end{pmatrix}$$

THE ESTIMATED FEATURE VECTOR OF A GIVEN USER CAN BE INFERRED THIS WAY:

$$\begin{pmatrix} \widehat{U}_{u,1} \\ \widehat{U}_{u,2} \\ \dots \\ \widehat{U}_{u,K} \end{pmatrix} = \begin{pmatrix} U_{1,1} & U_{2,1} & \dots & U_{N,1} \\ U_{1,2} & U_{2,2} & \dots & U_{N,2} \\ \dots & .. & \dots & \dots \\ U_{1,K} & U_{2,K} & \dots & U_{N,K} \end{pmatrix} \begin{pmatrix} T_{u,1} \\ T_{u,2} \\ \dots \\ T_{u,N} \end{pmatrix}$$

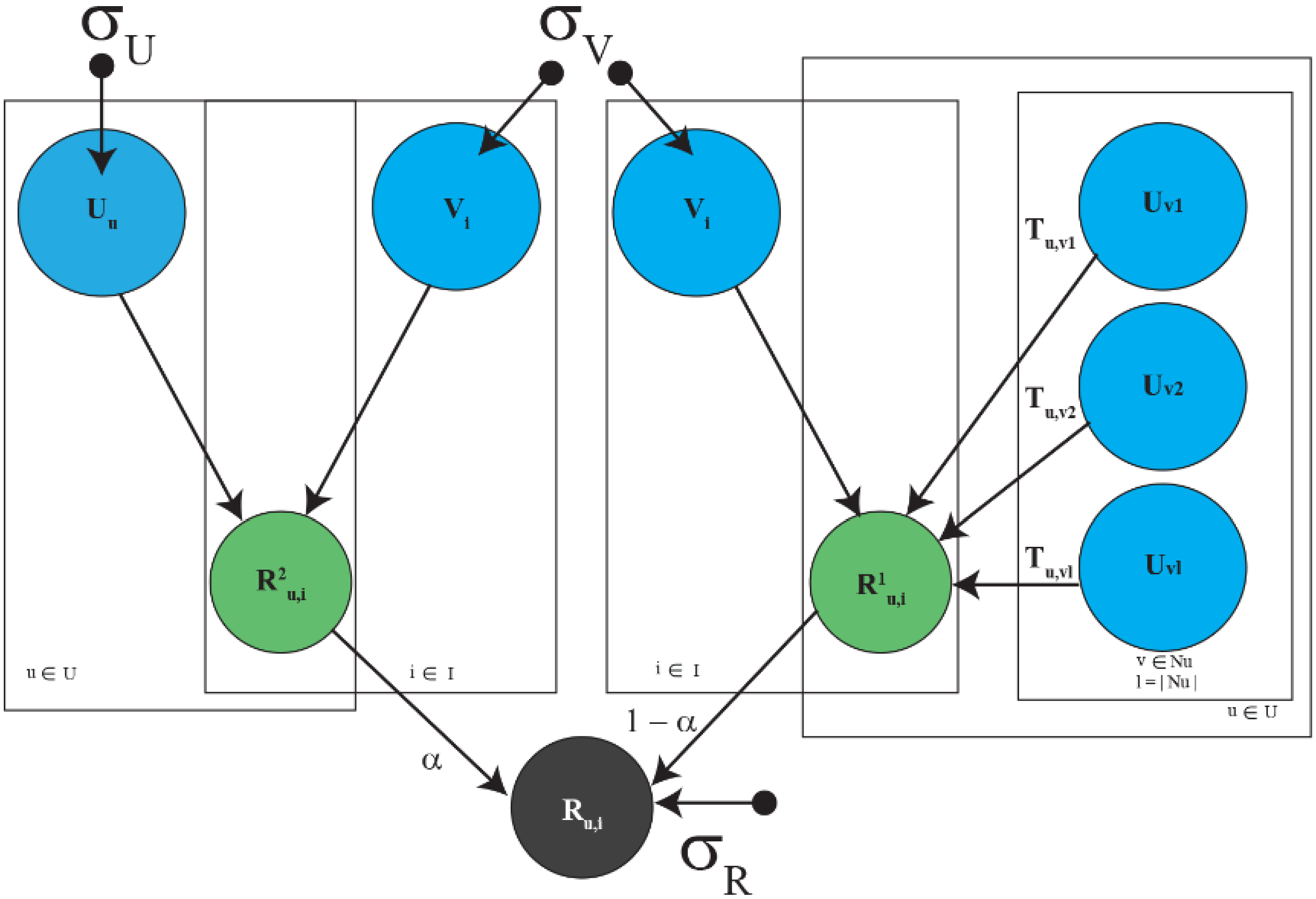
question to the audience:

hey, audience! what's the main difference between STE and SocialMF, by now?

AGAIN...

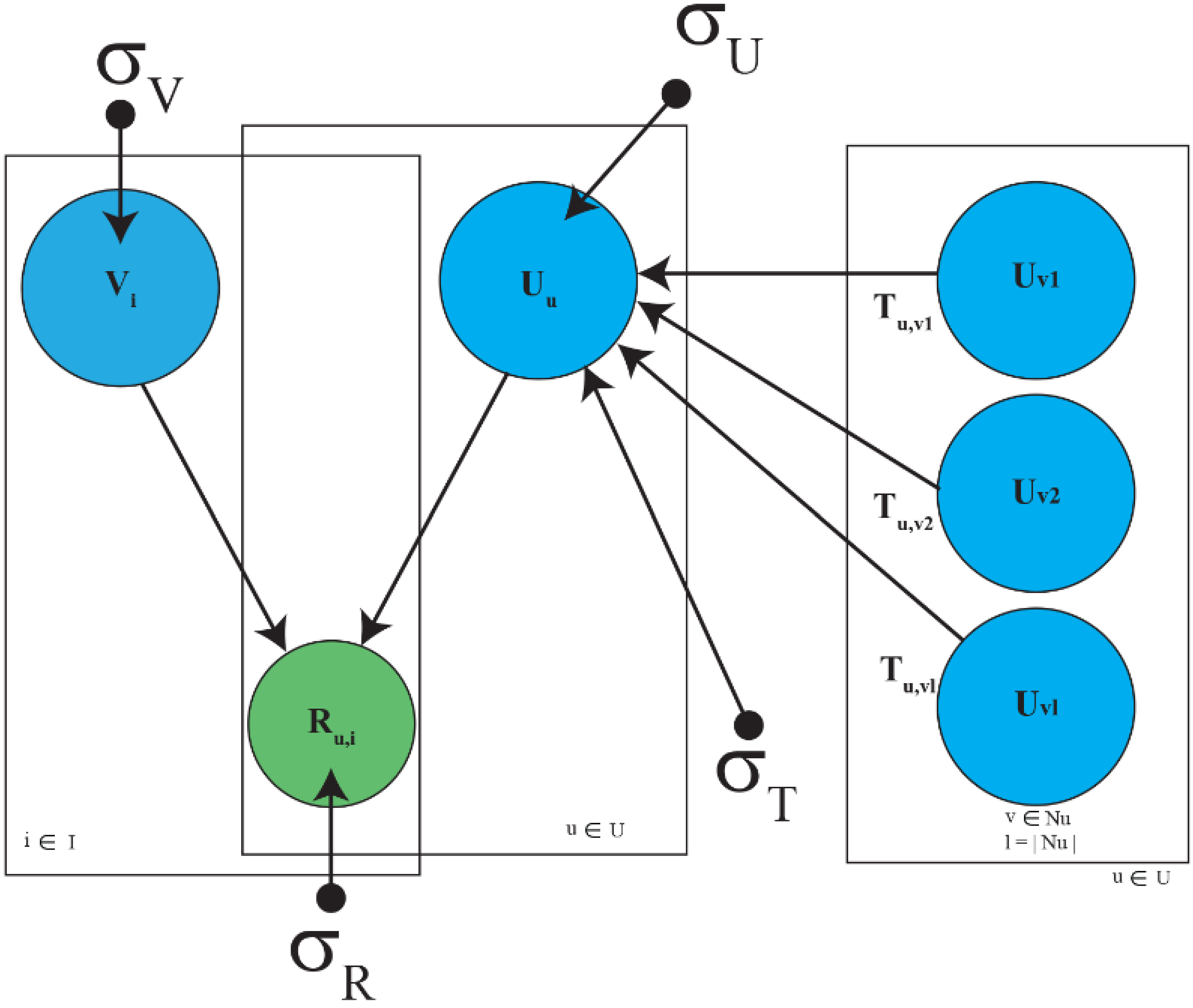
~~AGAIN...~~

nochmal



$$\hat{R}_{u,i} = g(\alpha U_u^T V_i \; + \; (1-\alpha) \sum_{v \in N_u} T_{u,v} U_v^T V_i)$$

AND SOCIAL MF?

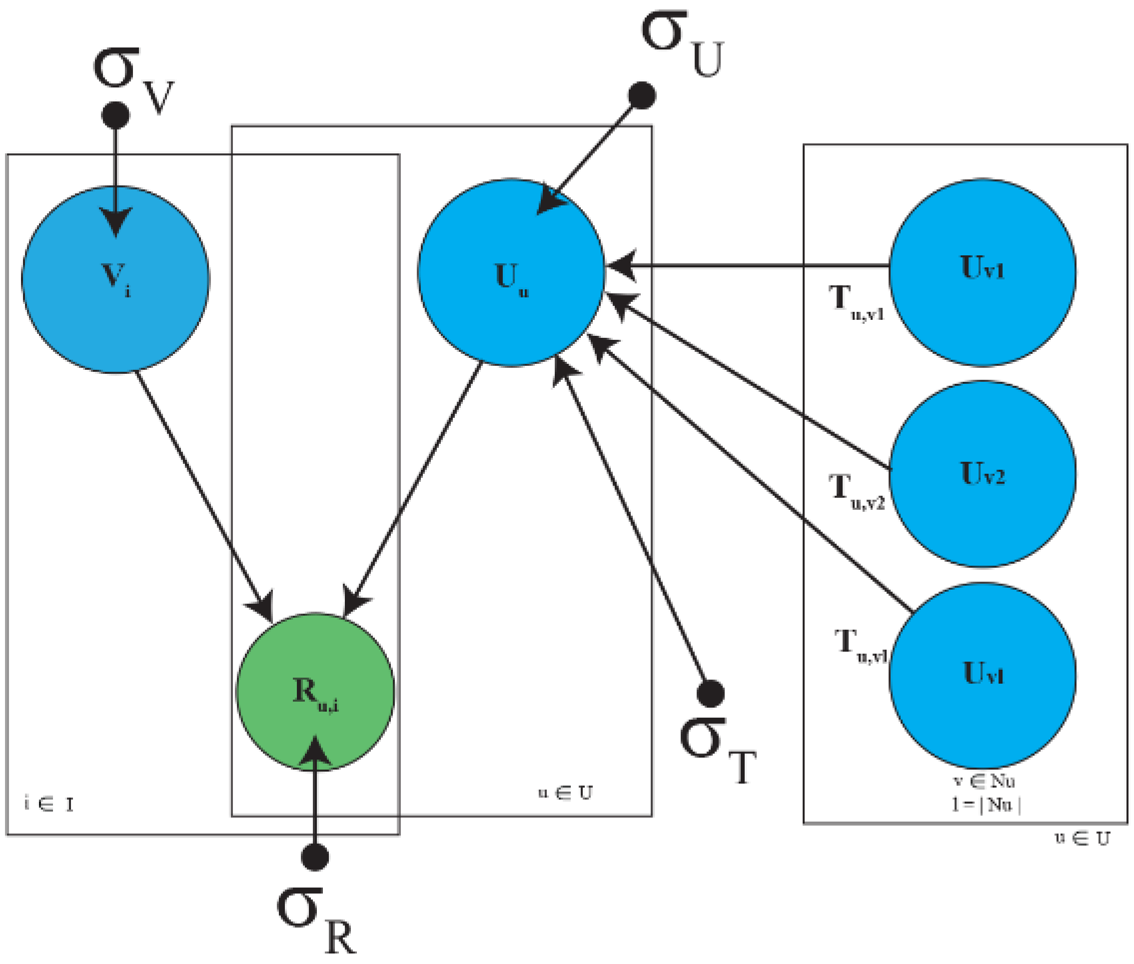

σV
σU
Vi
Uu
Uv1
Tu,v1
Uv2
Tu,v2
Tu,vl
Uvl
Ru,i
σT
i ∈ I
u ∈ U
v ∈ Nu
l = | Nu |
u ∈ U
σR

WAIT. NOT SO FAST!

~~WAIT. NOT SO FAST!~~

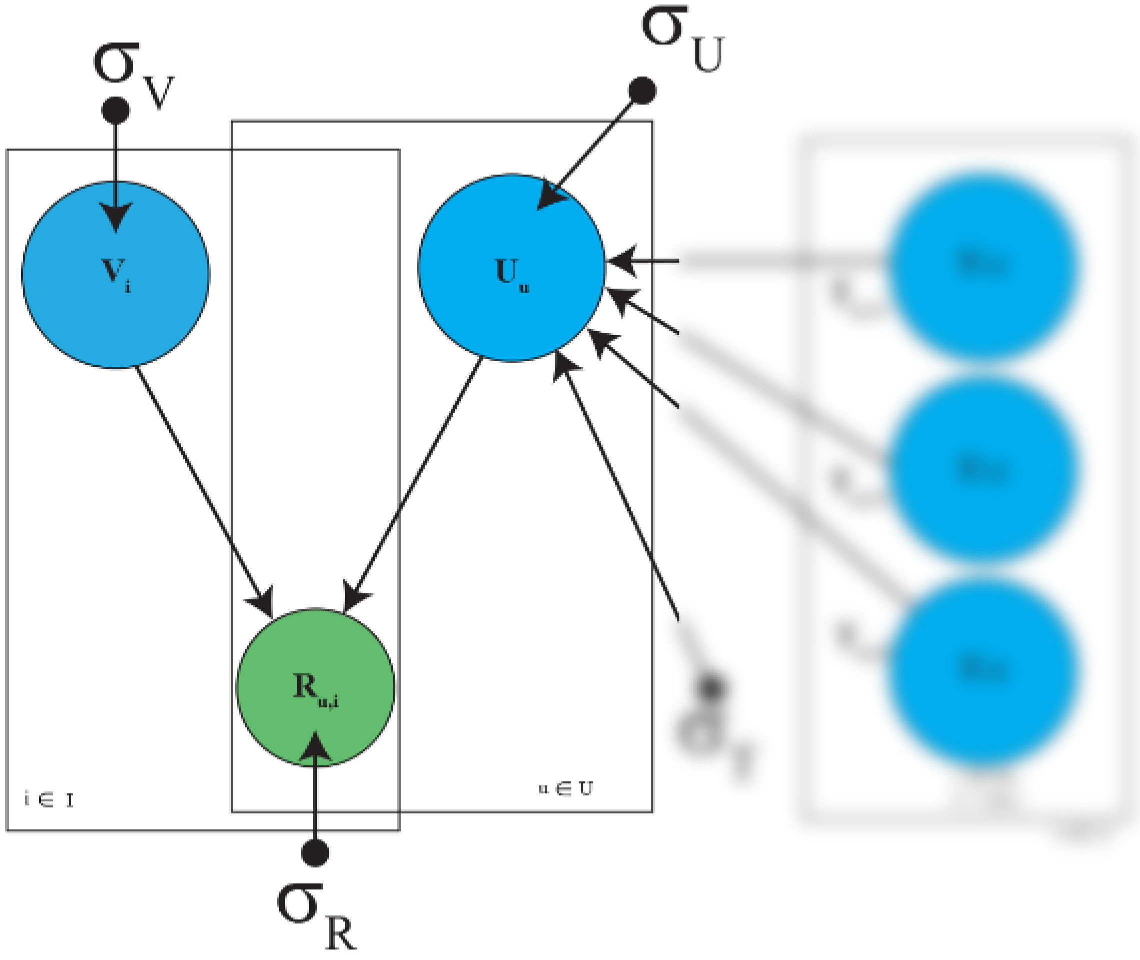
Let's get some maths done!

$$
\begin{aligned}
&p(U,V|R,T,\sigma_R^2,\sigma_T^2,\sigma_U^2,\sigma_V^2) \propto \\
&\qquad p(R|U,V,\sigma_R^2)p(U|T,\sigma_U^2,\sigma_T^2)p(V|\sigma_V^2) \\
&\qquad = \prod_{u=1}^{N}\prod_{i=1}^{M}\left[\mathcal{N}\left(R_{u,i}|g(U_u^T V_i),\sigma_r^2\right)\right]^{I_{u,i}^R} \\
&\qquad \times \prod_{u=1}^{N}\mathcal{N}\left(U_u|\sum_{v\in N_u} T_{u,v}U_v,\sigma_T^2\mathbf{I}\right) \\
&\qquad \times \prod_{u=1}^{N}\mathcal{N}\left(U_u|0,\sigma_U^2\mathbf{I}\right)\times\prod_{i=1}^{M}\mathcal{N}\left(V_i|0,\sigma_V^2\mathbf{I}\right)
\end{aligned}
$$

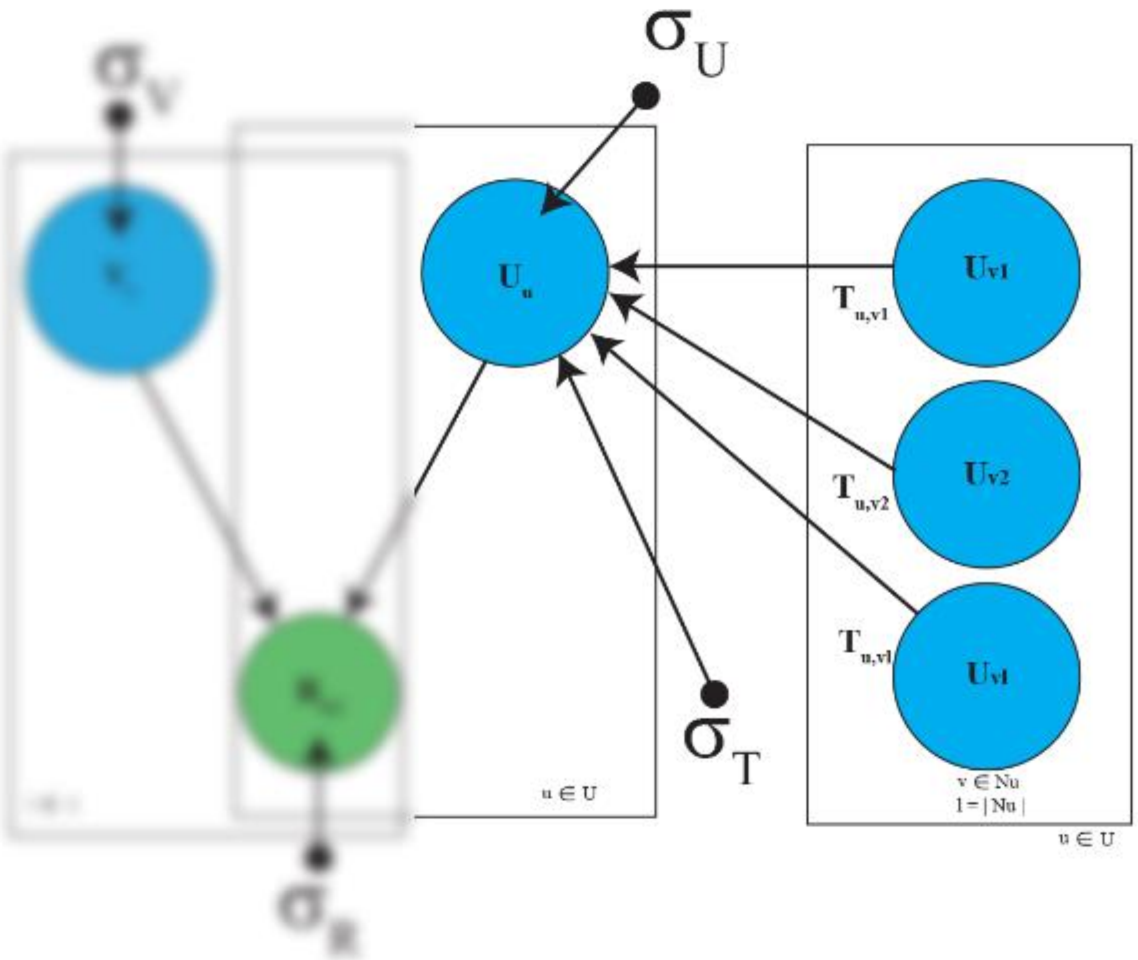


$$
\begin{aligned}
p(U, V | R, T, \sigma_R^2, \sigma_T^2, \sigma_U^2, \sigma_V^2) \propto \\
& p(R|U, V, \sigma_R^2) p(U|T, \sigma_U^2, \sigma_T^2) p(V|\sigma_V^2) \\
& = \prod_{u=1}^{N} \prod_{i=1}^{M} \left[\mathcal{N}\left(R_{u,i} | g(U_u^T V_i), \sigma_r^2 \right) \right]^{I_{u,i}^R}
\end{aligned}
$$

THE PRODUCT OF THE PROBABILITIES OF OBSERVED RATINGS ARE THE SAME AS IN MATRIX FACTORIZATION.

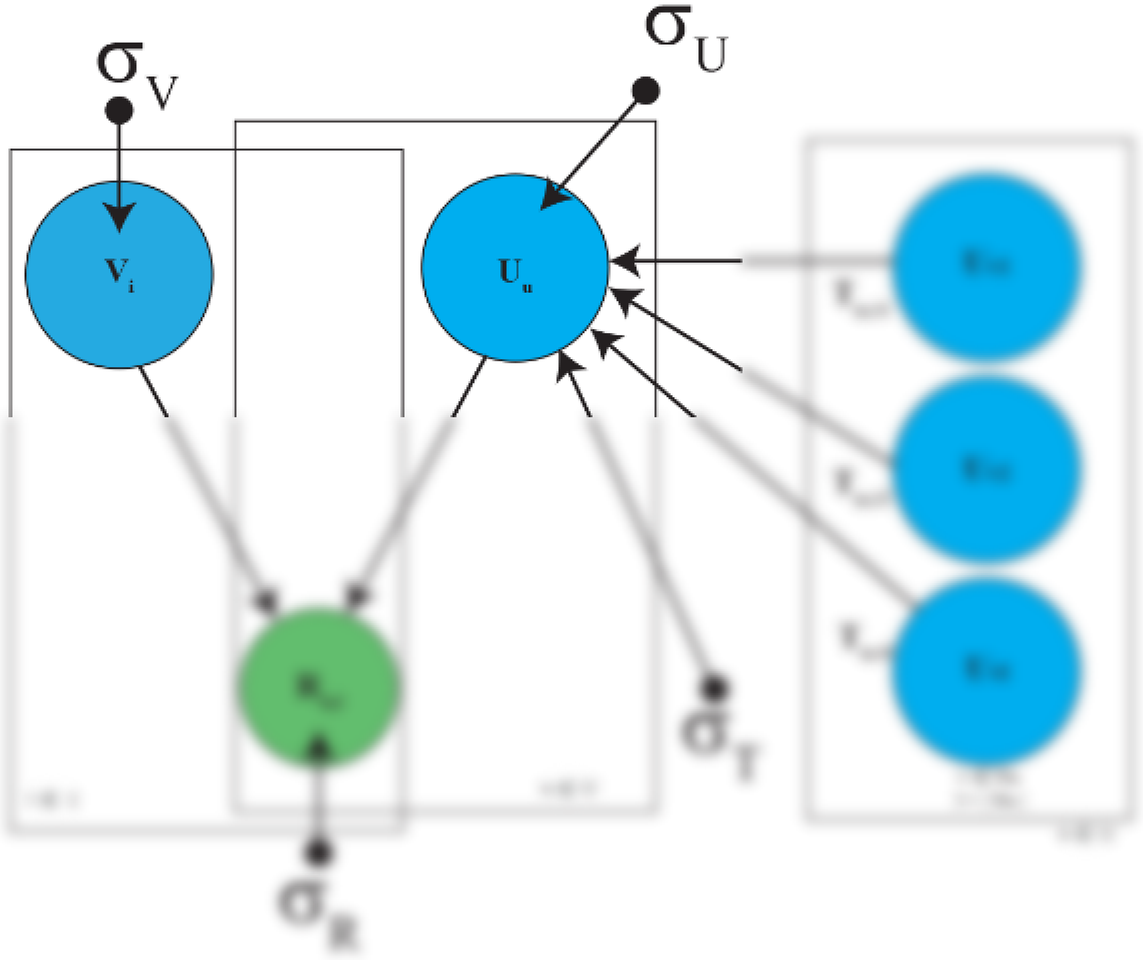


$$\times \prod_{u=1}^{N} \mathcal{N}\Big(U_u | \sum_{v \in N_u} T_{u,v} U_v, \sigma_T^2 \mathbf{I}\Big)$$



NORMAL PROBABILITY OVER THE LATENT USER FEATURES **INFLUENCED** BY THE LATENT FEATURES OF THE **DIRECT NEIGHBORS**.

$$\times \prod_{u=1}^{N} \mathcal{N}\left(U_u|0, \sigma_U^2 \mathbf{I}\right) \times \prod_{i=1}^{M} \mathcal{N}\left(V_i|0, \sigma_V^2 \mathbf{I}\right)$$



YES, RATING VALUES ARE DEPENDENT ON LATENT FEATURES OF USERS AND ITEMS. BUT, REMEMBER... LATENT FEATURES OF USERS ARE INFLUENCED BY DIRECT NEIGHBOURS!

AND ALSO...

BEING THE OBJECTIVE FUNCTION STATED BEFORE:

$$\hat{R}_{u,i} = g(\alpha U_u^T V_i \; + \; (1-\alpha) \sum_{v \in N_u} T_{u,v} U_v^T V_i)$$

A GRADIENT DESCENT CAN BE PERFORMED TO FIND A LOCAL MINIMUM FOR ALL USERS AND ITEMS:

$$\frac{\partial \mathcal{L}}{\partial U_u} = \sum_{i=1}^{M} I_{u,i}^{R} V_i g'(U_u^T V_i)(g(U_u^T V_i) - R_{u,i}) + \lambda_U U_u \quad + \lambda_T (U_u - \sum_{v \in N_u} T_{u,v} U_v)) - \lambda_T \sum_{\{v|u \in N_v\}} T_{v,u} \Big(U_v - \sum_{w \in N_v} T_{v,w} U_w \Big)$$

$$\frac{\partial \mathcal{L}}{\partial V_i} = \sum_{u=1}^{N} I_{u,i}^{R} U_v g'(U_u^T V_i)(g(U_u^T V_i) - R_{u,i}) + \lambda_V V_i$$

in the experiments:

LAMBDA V = LAMBDA U

$$\lambda_U = \sigma_R^2 / \sigma_U^2, \lambda_V = \sigma_R^2 / \sigma_V^2$$
$$\lambda_T = \sigma_R^2 / \sigma_T^2$$
$$\lambda_U = \lambda_V$$

$g'(x) = e^{-x} / (1 + e^{-x})^2$.

PSEUDOCODE

Inputs: observed ratings R, users U, items V and trust information T
Output: the latent feature vectors

U and V initialization - samples from normal noises with zero mean

while not converged do

update U: $\frac{\partial \mathcal{L}}{\partial U_u} = \sum_{i=1}^{M} I_{u,i}^{R} V_i g'(U_u^T V_i)(g(U_u^T V_i) - R_{u,i}) + \lambda_U U_u \ + \lambda_T (U_u - \sum_{v \in N_u} T_{u,v} U_v)) - \lambda_T \sum_{\{v|u \in N_v\}} T_{v,u} \Big(U_v - \sum_{w \in N_v} T_{v,w} U_w\Big)$

update V: $\frac{\partial \mathcal{L}}{\partial V_i} = \sum_{u=1}^{N} I_{u,i}^{R} U_v g'(U_u^T V_i)(g(U_u^T V_i) - R_{u,i}) + \lambda_V V_i$

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

datasets

WHICH DATA ARE WE DEALING WITH

DATASETS

Statistics	Flixster	Epinions
Users	1M	71K
Social Relations	26.7M	508K
Ratings	8.2M	575K
Items	49K	104K
Users with Rating	150K	47K
Users with Friend	980K	60K

COLD-START USERS REPRESENT MORE THAN 50% OF EACH DATASET.

for both datasets, all ratings have been normalized to a scale from zero to one.

experiments

RUNNING IT

EXPERIMENTS

COMPARING WITH:

- PLAIN MATRIX FACTORIZATION
- COLLABORATIVE FILTERING
- STE $\alpha = 0.4$

ERROR MEASURE: RMSE.

- 5-FOLD CROSS VALIDATION; 80% TRAIN, 20% TEST

$\lambda_U = \lambda_V = 0.1$

EXPERIMENTS

TAKING INTO ACCOUNT DIFFERENT SIZES OF K - 5 AND 10.

EPINIONS:

Method	K=5	K=10
CF	1.180	1.180
BaseMF	1.175	1.195
STE	1.145	1.150
SocialMF	1.075	1.085

LAMBDA T = 5

FLIXSTER:

Method	K=5	K=10
CF	0.911	0.911
BaseMF	0.878	0.863
STE	0.864	0.852
SocialMF	0.821	0.815

LAMBDA T = 1

EXPERIMENTS

OVERALL OBSERVATIONS:

EPINIONS:

FOR K = 5:

* 6.2% OF GAIN OVER STE

FOR K = 10:

* 5.7% OF GAIN OVER STE

FLIXSTER:

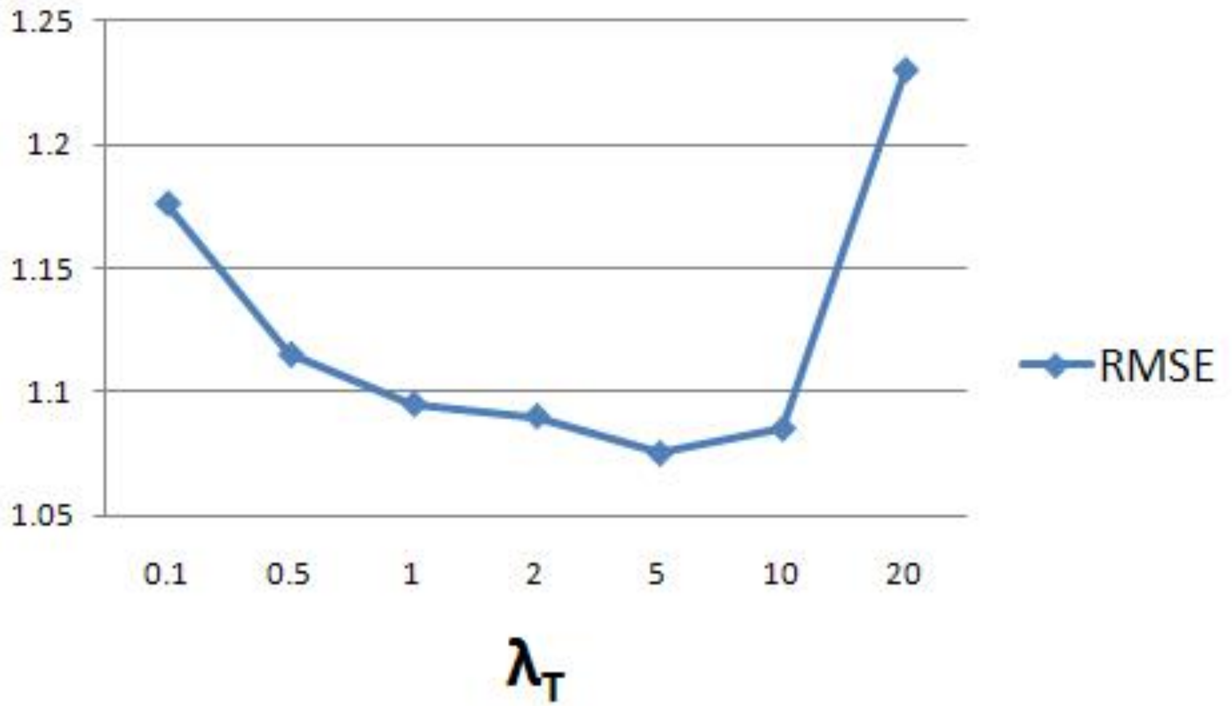
FOR K = 5 AND FOR K = 10:

* 5.0% OF GAIN OVER STE

EXPERIMENTS

THE DIFFERENT VALUES FOR LAMBDA T:

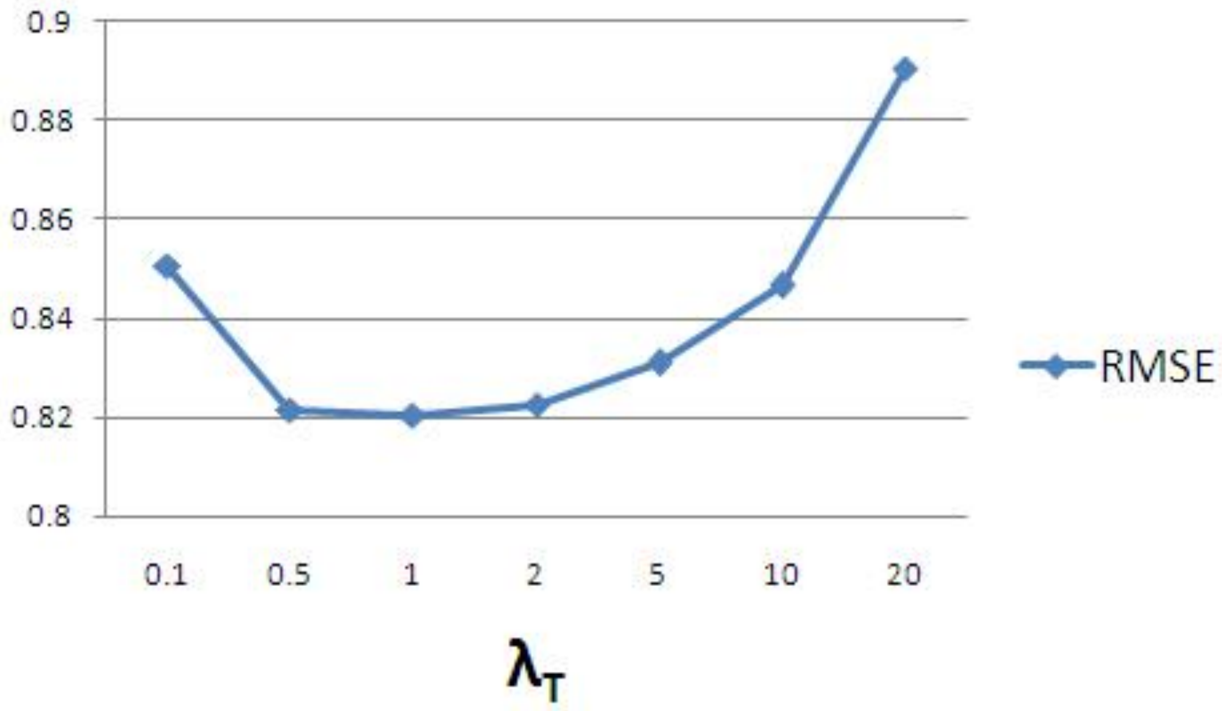
EPINIONS:



EXPERIMENTS

THE DIFFERENT VALUES FOR LAMBDA T:

FLIXSTER:



COLD-START USERS?

COLD-START USERS?

those with less than 5 ratings!

EXPERIMENTS

COLD-START USERS:

Method	Epinions	Flixster
CF	1.361	1.228
BaseMF	1.352	1.213
STE	1.295	1.152
SocialMF	1.159	1.057

K = 5 FOR BOTH DATASETS

EXPERIMENTS

COLD-START USERS:

11.5 % GAIN OVER STE

Method	Epinions	Flixster
CF	1.361	1.228
BaseMF	1.352	1.213
STE	1.295	1.152
SocialMF	1.159	1.057

K = 5 FOR BOTH DATASETS

EXPERIMENTS

COLD-START USERS:

8.5 % GAIN OVER STE

Method	Epinions	Flixster
CF	1.361	1.228
BaseMF	1.352	1.213
STE	1.295	1.152
SocialMF	1.159	1.057

K = 5 FOR BOTH DATASETS

conclusion + future

WHAT COMES NEXT

RELEVANT POINTS

OUTPERFORMS ALL OTHER METHODS COMPARED.

EVEN FOR COLD-START USERS!

RELEVANT POINTS

OUTPERFORMS ALL OTHER METHODS COMPARED.
EVEN FOR COLD-START USERS!

WHAT ABOUT NEGATIVE TRUST?
HOW COULD SOCIAL MF DEAL WITH IT?

references

GOOD SOURCES

Propagation of Trust and Distrust.
Guha, R. , Kumar R., Prabhakar, R., Tomkins, A.
https://pdfs.semanticscholar.org/3911/6b28f1a94e7d0aec082fb325ffdeae430012.pdf

Social-aware Matrix Factorization for Recommender Systems, 2013.
Weidele, D.
https://kops.uni-konstanz.de/bitstream/handle/123456789/29251/Weidele_0-259317.pdf

A Generative Bayesian Model for Item and User Recommendation in Social Rating Networks with Trust Relationships
C. Gianni, Manco G., Ortale R.
http://www.academia.edu/23622275/A_Generative_Bayesian_Model_for_Item_and_User_Recommendation_in_Social_Rating_Networks_with_Trust_Relationships

Recommended System with Social Regularization

Hao Ma,   Dengyoung Zhou,   Chao Liu,   Micheal R.Lyu,   Irwin King

Microsoft Research & Chinese University of Hong Kong 2011

Zafar Mahmood

Outline

Introduction and Motivation

- Widely studied for information retrieval
- For production Recommendation, used in Amazon, Itunes, Netflix etc
- We always ask friends for recommendation in different products
- We Used Trust aware Systems
- Previous methods ignores social relationship in process,

Trust Aware And Social Friends (1)

- Different Approaches
- “Trust aware” doesn’t have to know each other, … SoundCloud ,twitter etc
  - Based on the Assumption that user have similar taste
- “Social aware” to interact and connect with their friends in the real life, ... facebook etc
  - *Need to incorporate social information*

Traditional Systems

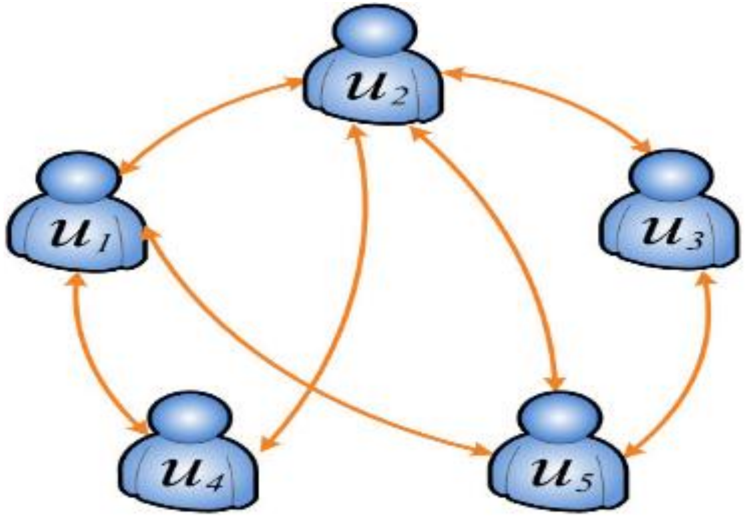
Collaborative Filtering

- Neighborhood Approaches (User or Items)
- Model Based approaches

Problem Definition

Predict the missing terms of user-item matrix by Incorporate the social network information

- Bidirectional social Connection (User – Item Matrix)
- Unidirectional trust Connection



	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

Low Rank Matrix Factorization

- We have User and Item Matrix, approx rating matrix by multiplying *l-rank factors*

$$R \approx U^T V$$

<u>*Extremely Sparse*</u> ………………………………………….. (1)

- Traditionally, we use Single Value Decomposition (SVD) for minimization of *R*

$$1/2 \left\| R - U^T V \right\|_F^2$$

…………………………………………………..(2)

- Due to sparsity we only need factorize the observed rating in matrix
- So, we use Indicator function for missing value's ----> *I* = {1,0}
    - when user rated the item = 1 , else = 0

$$\min_{U,V} 1/2 \sum_{i=1}^{m} \sum_{j=1}^{n} I_{i,j} (R_{i,j} - U_i^T V_j)^2$$

……..…...................(3)

Now to avoid overfitting, we add normalization

$$\min_{U,V} 1/2 \sum_{i=1}^{m} \sum_{j=1}^{n} I_{i,j}(R_{i,j} - U_i^T V_j)^2 - \lambda_1/2 \left\| U \right\|_F^2 + \lambda_2/2 \left\| V \right\|_F^2$$ …...........................(4)

$$\lambda_1, \lambda_2 > 0$$

- Now we can use Gradient Approach to Find the minimum

Social Regularization

Two models are used for social Regularization

- Average Based Model
- Individual Based Model

Average Based Model

We always ask our friend for recommendation using (…. 4) Matrix Factorization

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

……………………..(5)

$$\alpha > 0, \lambda_1, \lambda_2 > 0, F^+(i)\ldots.(i)$$
$$|F^+(i)| == |F^-(i)|\ldots.(ii)$$

In social Network, Facebook etc

- In (…...5) we have given the average taste users friends, which doesn't seems right, due to diverse taste nature …. changing it by introducing a similarity function

$$\min_{U,V} \mathcal{L}_1(R,U,V) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2 + \frac{\alpha}{2}\sum_{i=1}^{m} \| U_i - \frac{\sum_{f\in\mathcal{F}^+(i)} Sim(i,f)\times U_f}{\sum_{f\in\mathcal{F}^+(i)} Sim(i,f)} \|_F^2 + \frac{\lambda_1}{2}\|U\|_F^2 + \frac{\lambda_2}{2}\|V\|_F^2 \qquad \text{……………………(6)}$$

- As similarity is more accurate than our previous approach,
- Now to find the local minima, we just take the derivative

$$\frac{\partial \mathcal{L}_1}{\partial U_i} = \sum_{j=1}^{n} I_{ij}(U_i^T V_j - R_{ij})V_j + \lambda_1 U_i$$
$$+ \alpha(U_i - \frac{\sum_{f\in\mathcal{F}+(i)} Sim(i,f) \times U_f}{\sum_{f\in\mathcal{F}+(i)} Sim(i,f)})$$
$$+ \alpha \sum_{g\in\mathcal{F}^-(i)} \frac{-Sim(i,g)(U_g - \frac{\sum_{f\in\mathcal{F}+(g)} Sim(g,f)\times U_f}{\sum_{f\in\mathcal{F}+(g)} Sim(g,f)})}{\sum_{f\in\mathcal{F}+(g)} Sim(g,f)}$$

$R_{ij} = UserItem\ matrix$
$I_{ij} = Indicator\ Function$
$F^+ = Out\ link\ friends$
$F^- = In\ link\ friends$
$U_i = first\ person$
$U_f = first\ person\ friend$
$U_g = second\ person\ friend$

$$\frac{\partial \mathcal{L}_1}{\partial V_j} = \sum_{i=1}^{m} I_{ij}(U_i^T V_j - R_{ij})U_i + \lambda_2 V_j.$$

Individual-based Regularization

- Previously, we used similarity average of friends
- In reality users have diverse taste, so this could cause information loss so, add another regularization term,
- Constraint between user and their friends, individually

$$\frac{\beta}{2}\sum_{i=1}^{m}\sum_{f\in\mathcal{F}^{+}(i)} Sim(i,f)\|U_i - U_f\|_F^2, \qquad \text{……... (iii)}$$

Now putting in equation (……. 5)

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

………………………………. (6)

- Also deal with 2nd degree friends
- Like U(i) and U(g) are not friends but indirectly minimizing the distance between the feature vectors ….. (expanding.....(iii))

$$Sim(i,f)\|U_i - U_f\|_F^2 \text{ and } Sim(f,g)\|U_f - U_g\|_F^2.$$

- Now for local minima we again use the gradient descent (……… 6)

$$\begin{aligned}\frac{\partial \mathcal{L}_2}{\partial U_i} &= \sum_{j=1}^{n} I_{ij}(U_i^T V_j - R_{ij})V_j + \lambda_1 U_i \\ &+ \beta \sum_{f \in \mathcal{F}^+(i)} Sim(i,f)(U_i - U_f) \\ &+ \beta \sum_{g \in \mathcal{F}^-(i)} Sim(i,g)(U_i - U_g), \\ \frac{\partial \mathcal{L}_2}{\partial V_j} &= \sum_{i=1}^{m} I_{ij}(U_i^T V_j - R_{ij})U_i + \lambda_2 V_j\end{aligned}$$

$R_{ij} = UserItem\ matrix$

$I_{ij} = Indicator\ Function$

$F^+ = Out\ link\ friends$

$F^- = In\ link\ friends$

$U_i = first\ person$

$U_f = U_g = first\ person\ friend$

Similarity Function

We have User's rating, for similarity two methods are used.

- Two popular methods raging [0,1] Vector Space Similarity (VSS), ignore the individual rating behavior

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

- Pearson Correlation Coefficient (PCC) [-1,1], considers individual Rating behavior

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

To map in [0,1] we will do $f(x) = (x + 1)/ 2$

Datasets

two data-sets

- Douban
  - Rating and Recommendation about movies, books, music
  - Provides information about social friends
  - In Movie Group, Users = 129,490, Movies = 58,541,.. total rated cells in matrix = 16,830,839

Table 1: Statistics of User-Item Matrix of Douban

Statistics	User	Item
Min. Num. of Ratings	1	1
Max. Num. of Ratings	6,328	49,504
Avg. Num. of Ratings	129.98	287.51

Table 2: Statistics of Friend Network of Douban

Statistics	Friends per User
Max. Num.	986
Avg. Num.	13.07

Epinions

- Visitors read review of other users for and item selection
- Each user Maintain Trust list
- Users = 51,670; items = 83,509, ….. total rating cells in matrix = 631,064

Table 3: Statistics of User-Item Matrix of Epinions

Statistics	User	Item
Max. Num. of Ratings	1960	7082
Avg. Num. of Ratings	12.21	7.56

Table 4: Statistics of Trust Network of Epinions

Statistics	Trust per User	Be Trusted per User
Max. Num.	1763	2443
Avg. Num.	9.91	9.91

Comparison’s

Comparison with previous three other different methods

- NMF
  - For image analysis, also used in Collaborative Filtering
- Probabilistic Matrix Factorization (PMF)
  - User-item matrix for recommendation
- RECOMMENDATION WITH SOCIAL TRUST ENSEMBLE (RSTE) *
  - Trust aware recommendation user’s rating

Parameters

In Douban and Epinions , lemda (0.001)

Alpha = 0.001 …. on Douban

Beta = 0.01 …. on Epinions

Result By Doubian

- Results given by Different Previous Methods and Our Present Method SR_1 and SR_2,

Training	Metrics	UserMean	ItemMean	NMF	PMF	RSTE	$SR1_{vss}$	$SR1_{pcc}$	$SR2_{vss}$	$SR2_{pcc}$
80%	MAE Improve	0.6809 18.59%	0.6288 11.85%	0.5732 3.30%	0.5693 2.63%	0.5643 1.77%	0.5579	0.5576	0.5548	**0.5543**
	RMSE Improve	0.8480 17.59%	0.7898 11.52%	0.7225 3.28%	0.7200 2.94%	0.7144 2.18%	0.7026	0.7022	0.6992	**0.6988**
60%	MAE Improve	0.6823 18.02%	0.6300 11.22%	0.5768 3.03%	0.5737 2.51%	0.5698 1.84%	0.5627	0.5623	0.5597	**0.5593**
	RMSE Improve	0.8505 17.20%	0.7926 11.15%	0.7351 4.20%	0.7290 3.40%	0.7207 2.29%	0.7081	0.7078	0.7046	**0.7042**
40%	MAE Improve	0.6854 17.06%	0.6317 10.00%	0.5899 3.63%	0.5868 3.12%	0.5767 1.42%	0.5706	0.5702	0.5690	**0.5685**
	RMSE Improve	0.8567 16.83%	0.7971 10.61%	0.7482 4.77%	0.7411 3.86%	0.7295 2.33%	0.7172	0.7169	0.7129	**0.7125**

- SR_1 = Average Based Model
- SR_2 = Individual Based Model

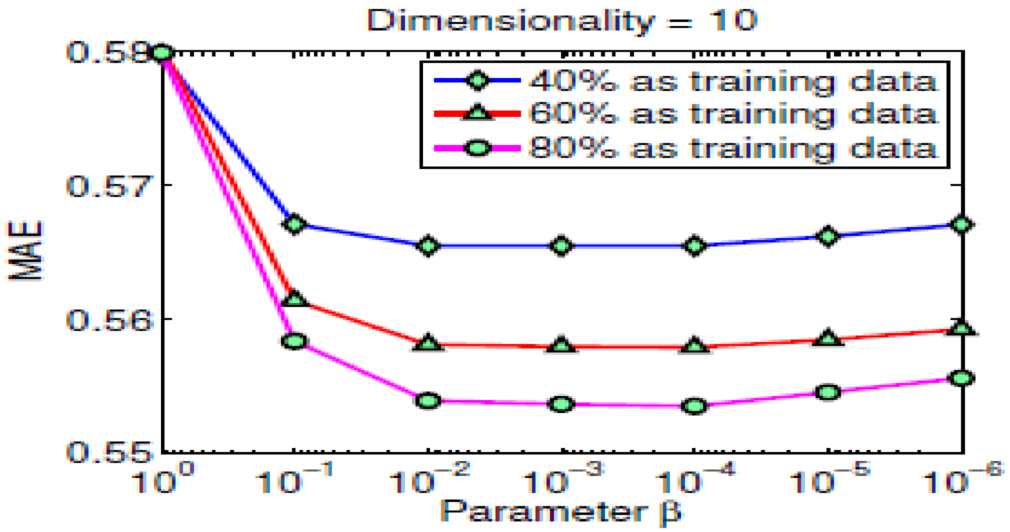
Results By Epinions

Training	Metrics	UserMean	ItemMean	NMF	PMF	RSTE	$SR1_{vss}$	$SR1_{pcc}$	$SR2_{vss}$	$SR2_{pcc}$
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%				

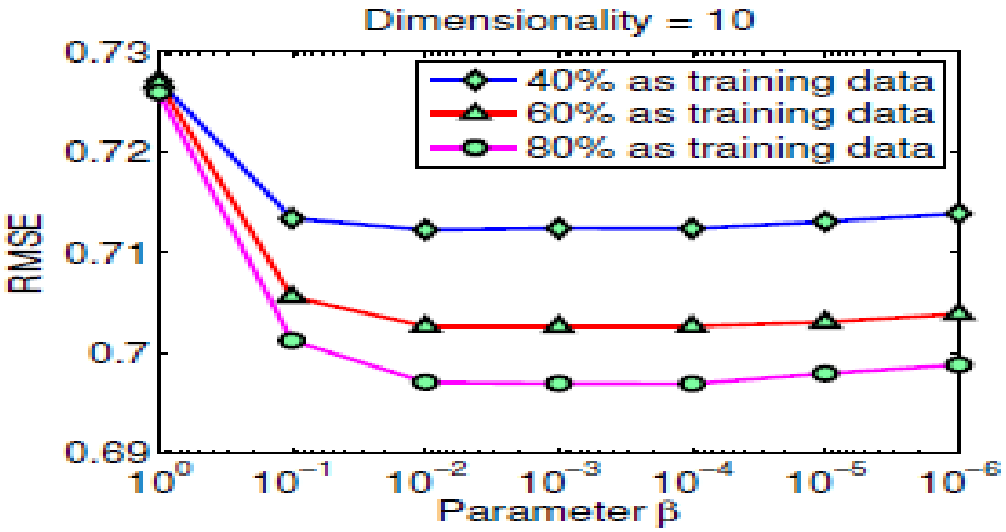
Impact Of Parameters

We keep the values of beta low

- Only uses second model
- Douban

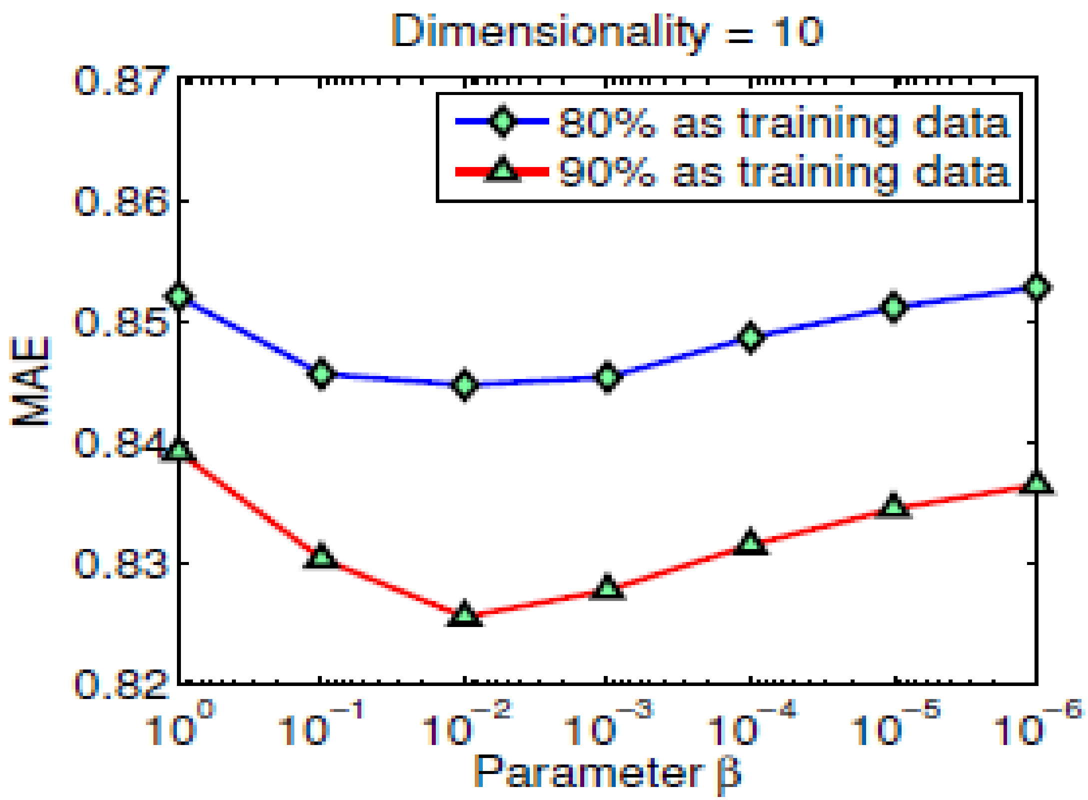


(a) Douban (MAE)

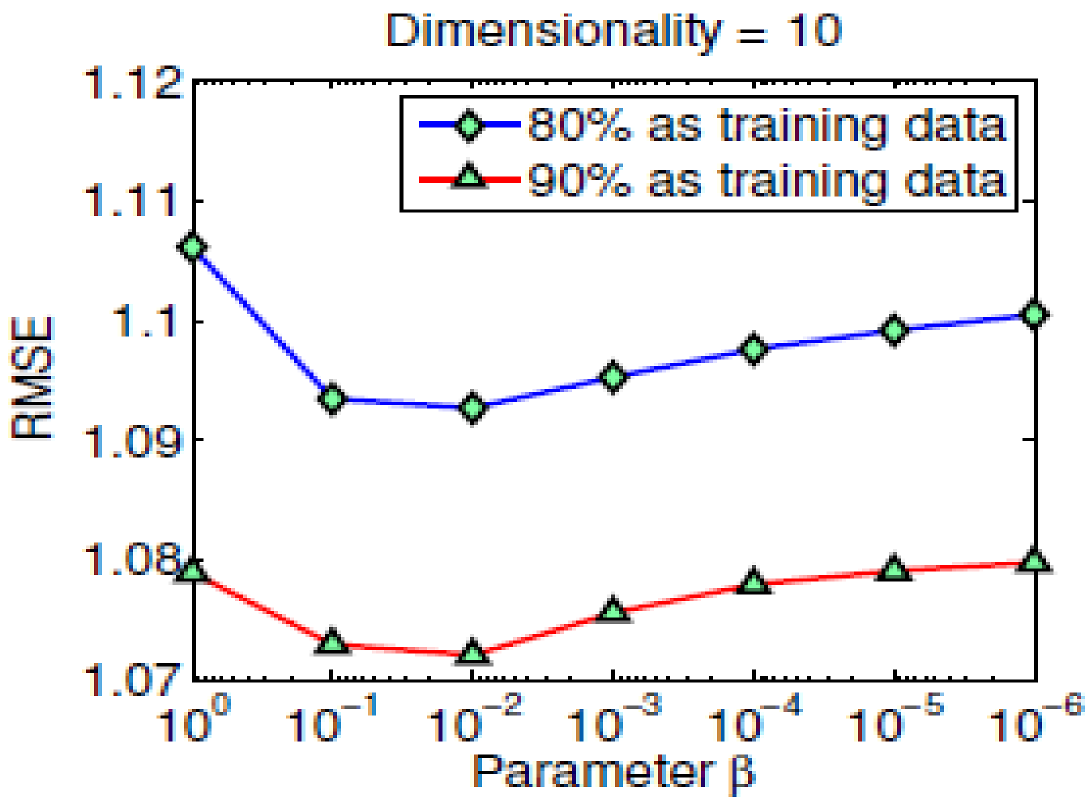


(b) Douban (RMSE)

Epinions



(c) Epinions (MAE)



(d) Epinions (RMSE)

Impact Of Similarity Functions

- Also test the similarity function by few alterations (random & set all to 1)
- As we used PCC and VSS for evaluation

Dataset	Training	Metrics	SR2 Sim=1	SR2 Sim=Ran	$SR2_{vss}$	$SR2_{pcc}$
Douban	80%	MAE	0.5579	0.5592	0.5548	0.5543
		RMSE	0.7034	0.7047	0.6992	0.6988
	60%	MAE	0.5631	0.5643	0.5597	0.5593
		RMSE	0.7083	0.7098	0.7046	0.7042
	40%	MAE	0.5724	0.5737	0.5690	0.5685
		RMSE	0.7195	0.7209	0.7129	0.7125
Epinions	90%	MAE	0.8324	0.8345	0.8258	0.8256
		RMSE	1.0794	1.0809	1.0744	1.0739
	80%	MAE	0.8511	0.8530	0.8447	0.8443
		RMSE	1.1002	1.1018	1.0958	1.0954

Pseudo Code : Averaging Method

Input

$$U = R^{lxm} \approx lxm$$
$$\mathrm{V} = R^{lxn} \approx lxn$$
$$\lambda_1 = \lambda_2 = 0.001$$
$$\alpha = 0.001$$

Algorithm

$$for\ i : m$$
$$for\ j : n$$
$$x = I_{i,j}\left(U_i^T V_j - R_{i,j} \right) + \lambda U_i$$
$$for\ f : f^+$$
$$b = b + (U_i - \frac{sim(i,f) * U_f}{sim\ (i,f)})$$
$$for\ g : f^-$$
$$c = c + \frac{\left(-sim(i,g)\ \frac{(U_g - sim(g,f)\ * U_f\ )}{sim(g,f)} \right)}{sim(g,f)}$$
$$V_j = I_{i,j}\left(U_i^T - R_{i,j}\right) * U_i + \lambda_2 V_j$$
$$U_i = x + \alpha * b + \alpha * c$$
$$return\ U, V$$

Pseudo Code : Individual Method

Input

$U = R^{lxm} \approx lxm$

$V = R^{lxn} \approx lxn$

$\lambda_1 = \lambda_2 = 0.001$

$\beta = 0.001$

Algorithm

$$
\begin{aligned}
& for\ i : m \\
& \quad for\ j : n \\
& \quad\quad x = I_{i,j}\left(U_i^T V_j - R_{i,j}\right) + \lambda U_i \\
& \quad\quad for\ f : f^+ \\
& \quad\quad\quad\quad b = b + sim(i,f) * (U_i - U_f) \\
& \quad\quad for\ g : f^- \\
& \quad\quad\quad\quad c = c + sim(i,g)(U_i - U_g) \\
& \quad V_j = I_{i,j}\left(U_i^T - R_{i,j}\right) * U_i + \lambda_2 V_j \\
& \quad U_i = x + \beta * b + \beta * c \\
& return\ U, V
\end{aligned}
$$

Conclusion and Future Work

- Two general algorithms are proposed that imposed social regularization using PCC and VSS
- **Quite generic** method also can be applied to trust aware recommendation problems
- **Comparison** shows it outperforms the state of the art **RSTE** method

- Make it more better if we have user's information about **Clicking behavior** and **Tagging Records**
- To make it more realistic we can use **categorical cluster** wise approach

References

- D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755):788–791, Oct. 1999.
- R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In Advances in Neural Information Processing Systems, volume 20, 2008.
- H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In Proc. Of SIGIR ’09, pages 203–210, Boston, MA, USA, 2009

Conclusions and Comparison

	SoRec	SocialMF	SRS
Model Based	**V**	**V**	**V**
Method	**Co-factorization**	**Regularization methods**	**Regularization methods**
$Social(\mathbf{T}, \mathbf{S}, \Omega)$	$\min \sum_{i=1}^{n} \sum_{u_k \in \mathcal{N}_i} (\mathbf{S}_{ik} - \mathbf{u}_i^\top \mathbf{z}_k)^2$	$\min \sum_{i=1}^{n} (\mathbf{u}_i - \sum_{u_k \in \mathcal{N}_i} \mathbf{S}_{ik} \mathbf{u}_k)^2$	$\min \sum_{i=1}^{n} \sum_{u_k \in \mathcal{N}_i} \mathbf{S}_{ik} (\mathbf{u}_i - \mathbf{u}_k)^2$
Dataset - Epinions03	**V**		**V**
Dataset - Epinions02		**V**	
Dataset - Douban			**V**
Dataset - Flixster		**V**	
Error Metric - RMSE		**V**	**V**
Error Metric - MAE	**V**		**V**

Conclusions and Comparison

	Best MAE	Best RMSE	Context
SoRec – Epinions	**0.8932**	---	Dimensionality = 10 99% Training Data
SocialMF – Epinions [2]	---	**1.075**	80% Training Data
SocialMF - Flixter	--	**0.815**	5-fold CV.
SRS – Epinions	**0.8256**	**1.0739**	PCC, Individual Method 90% Training Data
SRS - Douban	**0.5543**	**0.6988**	PCC , Individual Method 80% Training Data

Conclusions and Comparison

Issues on Social Recommendation

Social recommendation may also perform worse than tradicional recommender systems:

- social network composed of valuable friends, casual friends and event friends; users are not necessarily all that similar;
- social relations mixed with useful and noise connections;
- users with fewer ratings are likely to also have fewer connections.

QUESTIONS

Backup Slides - SRS

NMF

- Originally Used for image Analysis, But now widely used in Collaborative Filtering (For recommendation uses User Item matrix)
- algorithm for non-negative matrix factorization that is able to learn parts of faces and semantic features of text.
- *This is in contrast to other methods, such as principal components analysis and vector quantization*,
- that learn holistic, not parts-based, representations. Non-negative matrix factorization is distinguished from the other methods by its use of non-negativity constraints. These constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations.
- When non-negative matrix factorization is implemented as a neural network, parts-based representations emerge by virtue of two properties: the firing rates of neurons are never negative and synaptic strengths do not change sign.

PMF (Probabilistic Matrix Factorization)

- model which scales linearly with the number of observations and, more importantly, performs well on the large, sparse, and very imbalanced Netflix dataset
- users who have rated similar sets of movies are likely to have similar preferences
- When the predictions of multiple PMF models are linearly combined with the predictions of Restricted Boltzmann Machines models, we achieve an error rate of 0.8861, that is nearly 7% better than the score of Netflix's own system.

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

RSTE (RECOMMENDATION WITH SOCIAL TRUST ENSEMBLE)

- Aiming at modeling recommender systems more accurately and realistically, we propose a novel probabilistic factor analysis framework, which naturally fuses the users' tastes and their trusted friends' favors together.
- term Social Trust Ensemble (RSTE) to represent the formulation of the social trust restrictions on the recommender systems.

$$p(R|U,V,\sigma_R^2) = \prod_{i=1}^{m}\prod_{j=1}^{n}\left[\mathcal{N}\left(R_{ij}|g(U_i^T V_j),\sigma_R^2\right)\right]^{I_{ij}^R},$$

- Uses the epinion Dataset