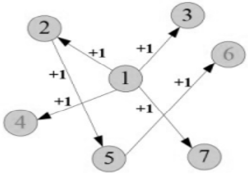
Recommendation in Signed Networks

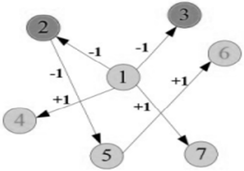
Presented by: Group 8

Eya Boumaiza
Khouloud Sallami
Turan Bilalov

Introduction

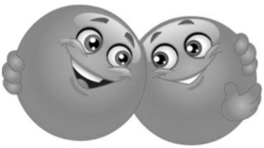
- The increasing popularity of social media allows users to participate in online activities in a pervasive way.
- The exploitation of social networks can potentially improve recommendation performance.
- Users in social networks are connected via various types of relations. Depending on these types of connections, we can distinct 2 types of social networks :
  - **Unsigned Social Networks**
  - **Signed Social Networks**

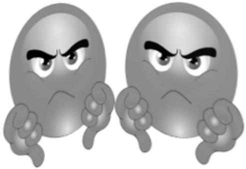




Introduction

- The vast majority of social recommender systems focus on unsigned social networks.
- little work exists for signed social networks (SSN).





Signed social networks will be exploited in the context of this presentation.

Papers

- Paper 1 : Recommendations in signed social networks

  Jiliang Tang, Charu Aggarwal, Huan Liu, April 2016

- Paper 2 : Recommending Positive Links in Signed Social Networks by Optimizing a Generalized AUC

  Dongjin Song, David A.Meyer, January 2015

- Paper 3 : Efficient latent link recommendation in signed networks

  Dongjin Song, David A.Meyer, Dacheng Tao, August 2015

Paper 1: Recommendations in Signed Social Networks

Jiliang Tang, Charu Aggarwal, Huan Liu

Eya Boumaiza

Outline

Outline

Motivation

- Negative links add a significant amount of knowledge than that embedded only in positive links [1]
- A small number of negative links can improve the performance of positive link prediction remarkably [2].

  → Potentially helpful in recommendations
- Negative links have different properties from positives links

  → Recommendation can not be successfully carried out by simply extending recommender systems with unsigned social networks. [3]

Hypothesis

- Introducting a novel recommendation framework, RecSSN, which mathematically exploits both positive and negative links from SSN.
- Evaluating the proposed framework to understand its effectiveness.

Outline

Problem statement

- In addition to the user-item matrix R $\in \mathbb{R}_{N*m}$, signed social networks among users are also available.
- A signed social network G can be decomposed into a positive component G_p and a negative component G_n.
- Let $A^p \in \mathbb{R}_{N*m}$ be the adjacency matrix of G_p where :

$$A^p_{ij} = \begin{cases} 1 & \text{if } u_i \text{ has a positive link to } u_j \\ 0 & \text{otherwise.} \end{cases}$$

- Let $A^n \in \mathbb{R}_{N*m}$ be the adjacency matrix of G_n where :

$$A^n_{ij} = \begin{cases} 1 & \text{if } u_i \text{ has a negative link to } u_j \\ 0 & \text{otherwise.} \end{cases}$$

Given observed values in R and a signed social network G with positive links A^p, and negative links A^n, the problem aims to infer missing values in R.

Outline

Proposed framework

- Two types of information from signed social networks can be exploited for recommendation : local information and global information [5].

- Local information reveals the correlations between the user and his friends or foes.
- Global information reveals the reputation of the user in the whole social network.

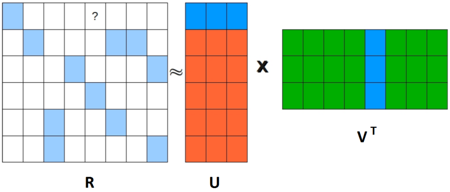
- Providing approaches to capture local and global information from signed social networks.
- Proposing a recommendation system, RecSSN, which exploits these model components from signed social networks.

How to first model this information ?

Matrix Factorization

- $U_i \in \mathbb{R}^K$: K-dimensional preference latent factor of u_i
- $V_j \in \mathbb{R}^K$: K-dimensional characteristic latent factor of item j.
- U $= \{U_1, U_2, ..., U_n\}$: user feature matrix
- V $= \{V_1, V_2, ..., V_m\}$: item feature matrix

$$R_{ij} = U_i^T V_j$$



Solving the following optimization problem [6] :

$$\min \sum_{i=1}^{N} \sum_{j=1}^{m} \mathbf{W}_{ij} \|\mathbf{R}_{ij} - \mathbf{U}_i^\top \mathbf{V}_j\|_2^2 + \alpha(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2)$$

where W_{ij} controls the confidence in rating R_{ij} and α is a hyperparameter to avoid overfitting.

Preference properties of users in SSN

To exploit global and local information in SSN, we need to understand users preference properties

- Unsigned social networks : a user's preference is similar to or influenced by their friends. [4]

→ Such assumptions are not applicable in signed social networks.

Investigating similar preference properties of users in signed social networks by :

- Performing data-driven analysis
- Making assumptions based on this analysis.
- Examining these assumptions by performing t-tests.

Datasets

- **Epinions :** a popular product review site
  - Users can create positive (trust) and negative (distrust) links to other users.
  - Users can rate various products with scores ranging from 1 to 5.
  - if u_i rates v_j, R_{ij} is the rating score, and R_{ij}=0 otherwise.
- **Slashdot :** a technology news platform
  - Users can create friend (positive) and foe (negative) links to other users
  - Users can specify tags associated with them.
  - if v_j is associated with u_i, R_{ij}=1, and R_{ij}=0 otherwise.

Statistics of the Epinions and Slashdot datasets

	Epinions	Slashdot
# of Users	18,210	11,868
# of Items	41,089	27,942
# of Positive Links	358,985	290,719
# of Negative Links	75,091	67,108
Density of User-item Matrix	8.42e-4	1.20e-3
# of Users with Negative Links	11,598	7,837

Preference properties of users in SSN

- Constructing 3 circles for each user u_i :
  - FR_i : randomly selected users who have positive links with u_i
  - FO_i : randomly selected users who have negative links with u_i
  - RA_i : randomly selected users who have no links with u_i.
- Computing the similarities s^p, s^n and s^r between users and their circles :

Average similarities between users and their circles

Epinions			
	CI	COSINE	CI-COSINE
"Friend" Circles ($\mathbf{s}^p$)	6.4520	0.0292	0.4954
"Foe" Circles ($\mathbf{s}^n$)	2.0808	0.0167	0.3811
Random Circles ($\mathbf{s}^r$)	1.2014	0.0092	0.2497
Slashdot			
	CI	COSINE	CI-COSINE
"Friend" Circles ($\mathbf{s}^p$)	8.5517	0.0456	0.5141
"Foe" Circles ($\mathbf{s}^n$)	2.5035	0.0206	0.4329
Random Circles ($\mathbf{s}^r$)	1.7151	0.0129	0.3226

- **Users are likely to be similar with their friends**
- **Users are likely to be more similar with their friends than their foes**

T-test

- Examining preceding assumptions by performing t-test on $\{s^p, s^n\}$ and $\{s^p, s^r\}$.

- $H_0 : s^p \leq s^n \quad H_1 : s^p \geq s^n$
- $H_0 : s^p \leq s^r \quad H_1 : s^p \geq s^r$

P-values of t-test results

	CI	COSINE	CI-COSINE
Epinions			
$\{s^p, s^r\}$	3.93e-124	6.07e-193	-2.71-111
$\{s^p, s^n\}$	3.12e-37	6.83e-65	2.35e-47
Slashdot	CI	COSINE	CI-COSINE
$\{s^p, s^r\}$	6.79e-140	5.62e-107	8.61e-85
$\{s^p, s^n\}$	1.83e-31	7.37e-27	3.89e-21

- Rejecting H_0 : Users are likely to be more similar with their friend circles than their foe circles.
- Rejecting H_0 : Users are likely to be similar with their friend circles.

Capturing Local Information from SSN

Dividing users into three groups :

- Users who have only positive links as OP$=\{u_i | P_i \neq \emptyset \cap N_i = \emptyset\}$
- Users who have only negative links as ON$=\{u_i | P_i = \emptyset \cap N_i \neq \emptyset\}$
- Users who have both positive and negative links as PN$=\{u_i | P_i \neq \emptyset \cap N_i \neq \emptyset\}$

- $\bar{U}_i^p$: the average user preferences of u_i's friend circle

$$\bar{U}_i^p = \frac{\sum_{u_j \in P_i} S_{ij} U_j}{\sum_{u_j \in P_i} S_{ij}}$$

- $\bar{U}_i^n$: the average user preferences of u_i's foe circle

$$\bar{U}_i^n = \frac{\sum_{u_j \in N_i} S_{ij} U_j}{\sum_{u_j \in N_i} S_{ij}}$$

where S_{ij} is the connection strength between u_i and u_j.

How to capture local information for these groups ?

Capturing Local Information from SSN

- For $u_i \in$ OP : we force u_i's preference close to P_i by minimizing the term :

$$min \; || \; U_i - \bar{U}_i^p \; ||_2^2$$

- For $u_i \in$ ON : untrustworthy, we ignore local information from these users. [7]

- For $u_i \in$ PN : u_i's preference is closer to that of his friend circle P_i than that of his foe circle N_i :

  - if u_i sits closer to P_i than N_i : $|| \; U_i - \bar{U}_i^p \; ||_2^2 - || \; U_i - \bar{U}_i^n \; ||_2^2 < 0$

  → **we should not penalize this case**

  - if u_i sits closer to N_i than P_i : $|| \; U_i - \bar{U}_i^p \; ||_2^2 - || \; U_i - \bar{U}_i^n \; ||_2^2 > 0$

  → **we should add a penalty to pull u_i closer to P_i than N_i**

We force u_i's preference closer to P_i than N_i by minimizing the term :

$$\min \max(0, || \; U_i - \bar{U}_i^p \; ||_2^2 - || \; U_i - \bar{U}_i^n \; ||_2^2)$$

Capturing Local Information from SSN

- If we define $\bar{U}_i^n = U_i$ for $u_i \in$ OP, the minimizing term :

$$\| U_i - \bar{U}_i^p \|_2^2 \; = \max(0, \| U_i - \bar{U}_i^p \|_2^2 - \| U_i - \bar{U}_i^n \|_2^2)$$

- If we define $\bar{U}_i^p = U_i$ for $u_i \in$ ON, the term :

$$\max(0, \| U_i - \bar{U}_i^p \|_2^2 - \| U_i - \bar{U}_i^n \|_2^2) = 0$$

- **we can find a unified term to capture local information from SSN as :**

$$min \sum_{i=1}^{n} max(0, \| U_i - \bar{U}_i^p \|_2^2 - \| U_i - \bar{U}_i^n \|_2^2)$$

where :

$$\bar{U}_i^p = \begin{cases} \frac{\sum_{u_j \in N_i} s_{ij} U_j}{\sum_{u_j \in N_i} s_{ij}} & \text{if } u_i \in \text{OP} \cup \text{PN} \\ u_i & \text{if } u_i \in \text{ON}. \end{cases} \qquad \bar{U}_i^p = \begin{cases} \frac{\sum_{u_j \in N_i} s_{ij} U_j}{\sum_{u_j \in N_i} s_{ij}} & \text{if } u_i \in \text{OP} \cup \text{PN} \\ u_i & \text{if } u_i \in \text{ON}. \end{cases}$$

Capturing global Information from SSN

- Calculating users reputations while taking into account negative links.
- $r_i \in \{1,2,...,N\}$ is the reputation ranking of u_i
  where $r_i = 1$ denotes that u_i has the highest reputation.
- $w_i = f(r_i)$ is the reputation score of u_i
  where f is a decreasing function of r_i that makes $w_i \in [0,1]$.
- Recommendations from users with high reputations are more likely to be trustworthy [8].

  $\longrightarrow$ Utilize users reputation scores to weight the importance of their recommendations.

Capturing global Information from SSN

- Originally :

$$\min \sum_{i=1}^{N}\sum_{j=1}^{m} \mathbf{W}_{ij}\|\mathbf{R}_{ij}-\mathbf{U}_i^{\top}\mathbf{V}_j\|_2^2+\alpha(\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2)$$

- We define the new weight for R_{ij} as $\hat{w}_{ij} = \mathrm{g}(W_{ij}\ , w_i)$

- **The formulation to capture global information from SSN is computed as :**

$$\min \sum_{i=1}^{N}\sum_{j=1}^{m} g(\mathbf{W}_{ij},\mathbf{w}_i)\|\mathbf{R}_{ij}-\mathbf{U}_i^{\top}\mathbf{V}_j\|_2^2+\alpha(\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2)$$

RecSSN Algorithm

- The proposed RecSSN framework solves the following optimization problem :

$$\min \sum_{i=1}^{N} \sum_{j=1}^{m} \underbrace{g(\mathbf{W}_{ij}, \mathbf{w}_i)}_{\text{Global Information}} \|(\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^\top)\|_2^2 + \alpha(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) + \underbrace{\beta \sum_{i=1}^{n} \max(0, \|\mathbf{U}_i - \bar{\mathbf{U}}_i^p\|_2^2 - \|\mathbf{U}_i - \bar{\mathbf{U}}_i^n\|_2^2)}_{\text{Local Information}}$$

- By setting g(W_{ij} , w_i) $= W_{ij}$ and ignoring all negative links, the proposed formulation will be equivalent to that of recommender systems with positive networks **SocialMF** :

$$\min \sum_{i=1}^{N} \sum_{j=1}^{m} \mathbf{W}_{ij} \|(\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^\top)\|_2^2 + \alpha(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) + \beta \sum_{i=1}^{n} \|\mathbf{U}_i - \bar{\mathbf{U}}_i^p\|_2^2$$

→ **A unified recommendation framework with unsigned and signed social networks**

RecSSN Algorithm

Using gradient decent method :

- We define at the k-th iteration for u_i :

$$\mathbf{M}_i^k = \begin{cases} 1 & \|\mathbf{U}_i - \bar{\mathbf{U}}_i^p\|_2^2 - \|\mathbf{U}_i - \bar{\mathbf{U}}_i^n\|_2^2 > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{J} = \sum_{i=1}^{N}\sum_{j=1}^{m} g(\mathbf{W}_{ij}, \mathbf{w}_i)\|(\mathbf{R}_{ij} - \mathbf{U}_i\mathbf{V}_j^\top)\|_2^2 + \alpha(\sum_{i=1}^{N}\|\mathbf{U}\|_2^2 + \sum_{j=1}^{m}\|\mathbf{V}_j\|_2^2)$$

$$+\beta\sum_{i=1}^{N}\mathbf{M}_i^k(\|\mathbf{U}_i - \frac{\sum_{u_j\in\mathcal{P}_i}\mathbf{S}_{ij}\mathbf{U}_j}{\sum_{u_j\in\mathcal{P}_i}\mathbf{S}_{ij}}\|_2^2 - \|\mathbf{U}_i - \frac{\sum_{u_j\in\mathcal{N}_i}\mathbf{S}_{ij}\mathbf{U}_j}{\sum_{u_j\in\mathcal{N}_i}\mathbf{S}_{ij}}\|_2^2)$$

$$\frac{\partial\mathcal{J}}{\partial\mathbf{U}_i} = -2\sum_j g(\mathbf{W}_{ij}, \mathbf{w}_i)(\mathbf{R}_{ij} - \mathbf{U}_i\mathbf{V}_j^\top)\mathbf{V}_j + 2\alpha\mathbf{U}_i + 2\beta\mathbf{M}_i^k(\mathbf{U}_i - \bar{\mathbf{U}}_i^p) - 2\beta\mathbf{M}_i^k(\mathbf{U}_i - \bar{\mathbf{U}}_i^n)$$

$$-2\beta\sum_{u_j\in\mathcal{P}_i}\mathbf{M}_j^k(\mathbf{U}_j - \bar{\mathbf{U}}_j^p)\frac{1}{\sum_{u_j\in\mathcal{P}_i}\mathbf{S}_{ji}} + 2\beta\sum_{u_j\in\mathcal{N}_i}\mathbf{M}_j^k(\mathbf{U}_j - \bar{\mathbf{U}}_j^n)\frac{1}{\sum_{u_j\in\mathcal{N}_i}\mathbf{S}_{ji}}$$

$$\frac{\partial\mathcal{J}}{\partial\mathbf{V}_j} = -2\sum_i g(\mathbf{W}_{ij}, \mathbf{w}_i)(\mathbf{R}_{ij} - \mathbf{U}_i\mathbf{V}_j^\top)\mathbf{U}_i + 2\alpha\mathbf{V}_j$$

Pseudo code

Algorithm 1: The Proposed Recommendation Framework RecSSN with Signed Social Networks.

Input: The rating information $\mathbf{R}$, positive links $\mathbf{A}_n$, negative links $\mathbf{A}_p$, the number of latent factors K and β
Output: The user preference matrix $\mathbf{U}$ and the item characteristic matrix $\mathbf{V}$

Initialize $\mathbf{U}$ and $\mathbf{V}$ randomly and set $k = 1$
while Not convergent **do**
  for $i = 1 : N$ **do**
    Calculate $\bar{\mathbf{U}}_i^p$ and $\bar{\mathbf{U}}_i^n$ according to

$$\bar{\mathbf{U}}_i^p = \begin{cases} \frac{\sum_{u_j \in \mathcal{P}_i} \mathbf{S}_{ij} \mathbf{U}_j}{\sum_{u_j \in \mathcal{P}_i} \mathbf{S}_{ij}} & \text{for } u_i \in \mathcal{OP} \cup \mathcal{PN} \\ \mathbf{U}_i & \text{for } u_i \in \mathcal{ON} \end{cases}$$

$$\bar{\mathbf{U}}_i^n = \begin{cases} \frac{\sum_{u_j \in \mathcal{N}_i} \mathbf{S}_{ij} \mathbf{U}_j}{\sum_{u_j \in \mathcal{N}_i} \mathbf{S}_{ij}} & \text{for } u_i \in \mathcal{ON} \cup \mathcal{PN} \\ \mathbf{U}_i & \text{for } u_i \in \mathcal{OP} \end{cases}$$

    Calculate $\mathbf{M}_i^k$ according to

$$\mathbf{M}_i^k = \begin{cases} 1 & \|\mathbf{U}_i - \bar{\mathbf{U}}_i^p\|_2^2 - \|\mathbf{U}_i - \bar{\mathbf{U}}_i^n\|_2^2 > 0 \\ 0 & \text{otherwise} \end{cases}$$

  end for
  Calculate $\frac{\partial \mathcal{J}}{\partial \mathbf{U}}$ and $\frac{\partial \mathcal{J}}{\partial \mathbf{V}}$
  Update $\mathbf{U} \leftarrow \mathbf{U} - \gamma_u \frac{\partial \mathcal{J}}{\partial \mathbf{U}}$
  Update $\mathbf{V} \leftarrow \mathbf{V} - \gamma_v \frac{\partial \mathcal{J}}{\partial \mathbf{V}}$
  k = k + 1
end while

Proposed framework

- Can the proposed RecSSN framework improve the recommendation performance by exploiting signed social networks ?

- Which model components of RecSSN contribute to the performance improvement ?

Outline

Experimental Settings

- **Epinions :** Scores in the user-item matrix denote the rating scores from users to items (from 1 to 5). we choose two metrics :

$$RMSE = \sqrt{\frac{\sum_{(u_i,v_j)\in\mathcal{T}} (\mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij})^2}{|\mathcal{T}|}}$$

$$MAE = \frac{1}{|\mathcal{T}|} \sum_{(u_i,v_j)\in\mathcal{T}} |\mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij}|,$$

where T is the set of ratings in the testing set

- **Slashdot :** Scores in the user-item matrix indicate whether users are associated with certain items. we choose two metrics :

$$precision@N = \frac{\sum_{u_i\in\mathcal{U}} |TopN_i \bigcap I_i|}{\sum_{u_i\in\mathcal{U}} |TopN_i|}$$

$$recall@N = \frac{\sum_{u_i\in\mathcal{U}} |TopN_i \bigcap I_i|}{\sum_{u_i\in\mathcal{U}} |I_i|}$$

where $TopN_i$ is the set of N items recommended to user u_i that u_i has not been associated in the training set, and I_i is the set of items that have been associated with u_i in the testing set

Performance Comparison of Recommender Systems

- Comparing the proposed RecSSN framework with existing recommender systems : Traditional collaborative filtering
- Can be grouped into memory-based systems and model-based systems

memory-based systems

- **UCF :** aggregating recommendations from ones' similar users based only on the user-item matrix.
- **pUCF :** combines recommendations from ones' similar users and their friends using both user-item matrix and positive links.
- **pnUCF :** excludes recommendations from ones' foes by exploiting negative links using user-item matrix, positive and negative links.

Performance Comparison of Recommender Systems

model-based systems

- **MF :** performs matrix factorization on the user-item matrix.
- **SocialMF :** combines both user-item matrix and positive links (a special case of the proposed framework with only positive links).
- **SoReg :** leverages both user-item matrix and positive links, and defines social regularization to capture positive links.
- **LOCABAL :** captures local and global information of positive links under the matrix factorization framework.
- **disSoReg :** two systems are proposed to exploit positive and negative links, respectively. disSoReg is a combination of these two systems to exploit positive and negative links

Performance Comparison of Recommender Systems

- Using cross-validation to determine parameters for all baseline methods.
- Empirically setting : $\alpha = 0.1$, K = 10, $f(x) = \frac{1}{log(x+1)}$, g(x,y) = x∗y

Comparison of Different Recommender Systems in Epinions

Training	Metrics	Memory-based Methods			Model-based Methods					
		UCF	pUCF	pnUCF	MF	SocialMF	SoReg	LOCABAL	disSoReg	RecSSN
50%	MAE	1.0323	0.9764	0.9683	1.0243	0.9592	0.9589	0.9437	0.9679	0.9273
	RMSE	1.2005	1.1477	1.1392	1.1902	1.1397	1.1354	1.1212	1.1407	1.0886
70%	MAE	1.0074	0.9493	0.9402	0.9988	0.9341	0.9327	0.9274	0.9425	0.8981
	RMSE	1.1758	1.1301	1.1196	1.1692	1.1163	1.1127	1.1009	1.1237	1.0697
90%	MAE	0.9817	0.9272	0.9187	0.9779	0.9189	0.9153	0.9017	0.9263	0.8863
	RMSE	1.1592	1.1059	1.0885	1.1525	1.0986	1.0951	1.0821	1.1032	1.0479

Comparison of Different Recommender Systems in Slashdot

Metrics	Memory-based Methods			Model-based Methods					
	UCF	pUCF	pnUCF	MF	SocialMF	SoReg	LOCABAL	disSoReg	RecSSN
P@5	0.0343	0.0372	0.0381	0.0354	0.0387	0.0386	0.0394	0.0379	0.0419
R@5	0.0438	0.0479	0.0485	0.0453	0.0492	0.0488	0.0498	0.0473	0.0511
P@10	0.0332	0.0358	0.0364	0.0338	0.0365	0.0368	0.0375	0.0359	0.0388
R@10	0.0413	0.0454	0.0463	0.0427	0.0463	0.0467	0.0479	0.0457	0.0497

- **The proposed RecSSN framework always obtains the best performance.**

Impact of Negative Links on RecSSN

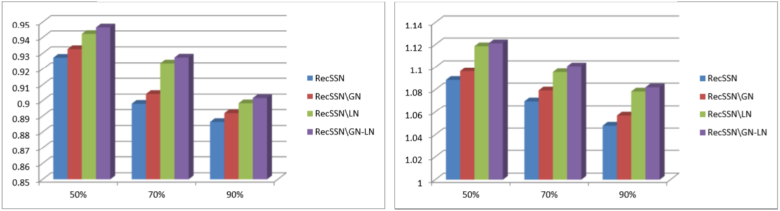
Investigating the impact of negative links on the proposed framework by eliminating their effect systematically

We define the following algorithmic variants :

- **RecSSN\GN :** Eliminating the effect of negative links from global information of signed social networks by computing status scores of users with only positive links.
- **RecSSN\LN :** Eliminating the effect of negative links from local information of signed social networks by replacing :
  $$\sum_{i=1}^{n} max(0, \| U_i - \bar{U}_i^p \|_2^2 - \| U_i - \bar{U}_i^n \|_2^2) \text{ with } \sum_{i=1}^{n} \| U_i - \bar{U}_i^p \|_2^2$$
  in the main objective function.
- **RecSSN\GN-LN :** Eliminating the effect of negative links from global and local information of signed social networks.

Impact of Negative Links on RecSSN

Impact of Negative Links on The Proposed Framework RecSSN in Epinions



(a) AME (b) RMSE

Relative Performance Reductions for Variants Compared to RecSSN

Variants	50%		70%		90%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
RecSSN\GN	-0.88%	-1.02%	-0.98%	-1.21%	-0.92%	-1.15%
RecSSN\LN	-2.06%	-3.06%	-3.15%	-2.71%	-1.67%	-3.21%
RecSSN\GN-LN	-2.59%	-3.29%	-3.56%	-3.22%	-2.04%	-3.56%

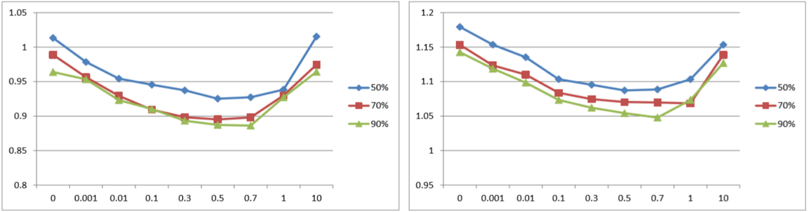
* similar results for Slashdot

- **Both local and global information of negative links in the proposed RecSSN framework can help improve the recommendation performance**

Parameter Analysis for RecSSN

Investigating how changes of β affect the performance of RecSSN by Varying its value

Performance Variations of The Proposed RecSSN Framework w.r.t. β in Epinions



(a) MAE (b) RMSE

* similar results for Slashdot

- β controls the contribution of local information in signed social networks.
- Local information is helpful in improving recommendation performance.
- After a point, The estimates of U and V will overfit to the local information.

Outline

Conclusion

- A novel recommendation framework RecSSN : Exploiting local and global information from signed social networks.
- Experimental results demonstrate that RecSSN outperforms various state-of-the art recommender systems.
- Further experiments are conducted to understand the importance of signed social networks in RecSSN.

Future Work

- Matrix factorization is the basic model on top of which the algorithms are constructed : Investigating whether other types of models can be used.
- User preferences and signed social networks might evolve : Incorporating temporal information into the proposed RecSSN framework.

References

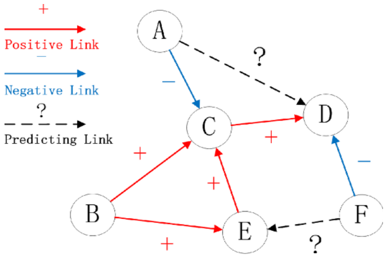
- [1] J. Kunegis, J. Preusse, and F. Schwagereit. What is the added value of negative links in online social networks ? In *Proceedings of the 22nd international conference on World Wide Web*, pages 727–736. International World Wide Web Conferences Steering Committee, 2013.
- [2] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, pages 403–412. ACM, 2004
- [3] J. Tang, X. Hu, and H. Liu. Is *distrust the negation of trust ? the value of distrust in social media*. In *ACM Hypertext conference*, 2014.
- [4] J. Tang, J. Tang, and H. Liu. Recommendation in social media : recent advances and new frontiers. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1977–1977. ACM, 2014.
- [5] J. Tang and H. Liu. Trust in social computing. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 207–208. International World Wide Web Conferences Steering Committee, 2014.
- [6] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8) :30–37, 2009.
- [7] P. Victor, C. Cornelis, M. De Cock, and A. Teredesai. Trust-and distrust-based recommendations for controversial reviews. In *Web Science Conference (WebSci'09 : Society On-Line)*, number 161, 2009.
- [8] P. Massa. A survey of trust use and modeling in real online systems. *Trust in E-services : Technologies, Practices and Challenges*, 2007

What about Link recommendation?

Link Recommendation

Treats the problem of potential links[1] as a Personalized Ranking Problem

- aims to suggest a list of people (or items) to each user with whom the user might create new connections.
- In the ranked list, people (or items) are recommended in decreasing order of ranking scores (which estimate the user's preferences).



[1] : Unknown links, Predicting links

Different Link Recommendation approaches

Network Topology based approaches

- **Neighbor-based methods** : recommend links based upon their neighborhood structure.
- **Path-based methods** : produce ranking scores by considering the ensemble of all paths between two nodes.

Latent Feature based approaches

- **Pointwise methods** : treat link recommendation as a matrix completion problem and reconstruct the adjacency matrix of a partially observed social network from a low rank model.
- **Pairwise methods** : treat link recommendation as a learning to rank problem based upon pairwise comparisons
- **Listwise methods** : aim to learn a ranking function by taking individual lists as instances and minimizing a loss function defined on the predicted list and the ground truth list.

Unsigned Social Networks VS Signed Social Networks

Unsigned Social Networks

- Can be represented as a binary adjacency matrix :
  - **1** Existence of a link
  - **0** Unknown Status of a link
- Link recommendation aims to suggest to each user a list of people/items to whom the user probably will create new connections.

Signed Social Networks

- Can be represented as an adjacency matrix with :
  - **1** Positive relationship
  - **-1** Negative relationship
  - **0** Unknown link
- Given a user, the aim is to rank people (items) this user is interested in (i.e., positive links) on the top and people (items) this user is not interested in (i.e., negative links) at the bottom.

Signed Directed Social Networks

Wikipedia

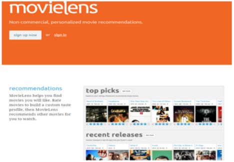
The Wikipedia data comprise a voting network for promoting candidates to the role of admin.
vote(1 :support, 0 :neutral, -1 :oppose)

Slashdot

Slashdot features news stories on science and technology that are submitted and evaluated by site users.
In 2002, it introduces SlashdotZoo feature which allows users to tag each other as friends or foes based up on articles.

MovieLens

1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000.
The ratings of 4 and 5 : Positive links
The ratings of 1 and 2 : Negative links
Other ones : Unknown status links



ROC(Receiver Operating Characteristic) Curve

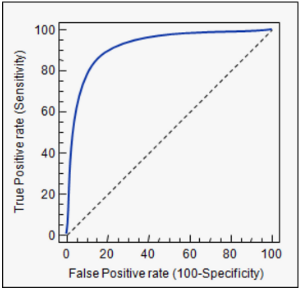
A graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied.The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

- Sensitivity (True Positive Rate) :

$$\frac{\textit{Correctly classified positive}}{\textit{Total positive}}$$

- Specificity (True Negative Rate) :

$$\frac{\textit{Correctly classified negative}}{\textit{Total negative}}$$



*Ref [1]

Area Under the Curve ROC

$$AUC = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{a_i \in \mathcal{P}} \sum_{a_j \in \mathcal{N}} I\big(f(a_i) > f(a_j)\big)$$

- f a binary classifier,(a_i,b_i) $_{i=1..n}$ a training set with $a_i \in \mathbb{R}^d$ and $b_i \in \{$-1,1$\}$
- P : set of positive samples, N : set of negative samples
- AUC is the probability that a random element of one set f(a_i) is larger than a random element of another f(a_j).

- For an ideal ranking list : AUC should be 1 representing each positive sample is ranked higher than all the negative samples.
- For a random ranking list : AUC will be 0.5
  *Ref [2]

Recommending Positive Links in Signed Social Networks by Optimizing a Generalized AUC

Dongjin Song, David A.Meyer (2015)

Khouloud Sallami

Outline

Outline

Motivation

AUC only applies to the binary case. In signed networks with triplet (positive, negative, and unknown) :

- If AUC treats positive links as positive samples and the other links as negative samples, it will be impossible to quantify the ranking quality of negative links.
- If AUC treats negative links as negative samples and the other links as positive samples, it will not measure the ranking performance of positive links correctly.

$$\Downarrow$$

AUC as a traditional ranking measure is not an appropriate way to quantify the ranking performance in signed networks.

Hypothesis

- Introduce a generalized AUC which will be maximized only if all positive links are ranked on top, all negative links are ranked at the bottom, and all unknown status links are in the middle.
- Develop a link recommendation approach by directly minimizing the loss of the proposed GAUC.
- Demonstrate the effectiveness of the generalized AUC for quantifying link recommendation in signed social networks based upon experimental studies with three real world datasets

Outline

A Generalized AUC

GAUC can jointly quantify the ranking quality over positive links and negative links, in the presence of unknown links :

$$GAUC = \frac{\eta}{|\mathcal{P}|(|\mathcal{O}| + |\mathcal{N}|)} \sum_{a_i \in \mathcal{P}} \sum_{a_s \in \mathcal{O} \bigcup \mathcal{N}} I\big(f(a_i) > f(a_s)\big) + \frac{1-\eta}{|\mathcal{N}|(|\mathcal{O}| + |\mathcal{P}|)} \sum_{a_t \in \mathcal{O} \bigcup \mathcal{P}} \sum_{a_j \in \mathcal{N}} I\big(f(a_j) < f(a_t)\big)$$

- f a binary classifier,$(a_i, b_i)_{\, i=1..n}$ a training set with $a_i \in \mathbb{R}^d$ and $b_i \in \{\textbf{-1,0,1}\}$
- P : set of positive samples, N : set of negative samples, O : set of unknown samples
- The parameter η controls the trade-off between the two terms of GAUC, $\eta = \frac{|P|}{|P|+|N|}$

$$GAUC = \frac{1}{(|\mathcal{P}| + |\mathcal{N}|)} \Big(\frac{1}{(|\mathcal{O}| + |\mathcal{N}|)} \sum_{a_i \in \mathcal{P}} \sum_{a_s \in \mathcal{O} \bigcup \mathcal{N}} I\big(f(a_i) > f(a_s)\big) + \frac{1}{(|\mathcal{O}| + |\mathcal{P}|)} \sum_{a_j \in \mathcal{N}} \sum_{a_t \in \mathcal{O} \bigcup \mathcal{P}} I\big(f(a_j) < f(a_t)\big) \Big)$$

- For a perfect ranking list : GAUC will be 1
- For a random ranking : GAUC will be 0.5

Problem Statement

- A partially observed signed network X $\in \mathbb{R}^{nxn}$ with $X_{i,j} \in \{1, -1, 0, ?\}$ where 1 denotes a positive link, -1 represents a negative link, 0 is an unknown status link, and ? denotes a potential positive or negative link.
- Objective : To learn a mapping function f such that a ranking score for the link at i-th row and j-th column of X can be produced as $f(i,j,X)=\widehat{X_{i,j}}$
- Many real world signed social networks are sparse graphs with low rank structure $\Longrightarrow f(i,j,U,V) = f(U_i^T, V_j)$
  where $U_i \in \mathbb{R}^r$,$V_j \in \mathbb{R}^r$, and r is the rank $r \ll n$
  *Ref [4]

Link Recommendation Model

- The loss of GAUC is defined as :

$$1 - GAUC(U,V) = \frac{1}{(|\mathcal{P}| + |\mathcal{N}|)}\Big(\frac{1}{(|\mathcal{O}| + |\mathcal{N}|)} \sum_{X_{ij}\in\mathcal{P}} \sum_{X_{is}\in\mathcal{O}\bigcup\mathcal{N}} I(U_i^T V_j \leq U_i^T V_s)$$
$$+ \frac{1}{(|\mathcal{O}| + |\mathcal{P}|)} \sum_{X_{ij}\in\mathcal{N}} \sum_{X_{is}\in\mathcal{O}\bigcup\mathcal{P}} I(U_i^T V_j \geq U_i^T V_s)\Big)$$

- Based on the non-convexity of $I(.)$:

$$I(U_i^T V_j \geq U_i^T V_s) \leq \max\Big(0, U_i^T (V_j - V_s) + 1\Big)$$
$$I(U_i^T V_j \leq U_i^T V_s) \leq \max\Big(0, U_i^T (V_s - V_j) + 1\Big)$$

- Based on the sparsity of real signed graphs :

$$|\mathcal{O}| \gg |\mathcal{P}|$$
$$|\mathcal{O}| \gg |\mathcal{N}|$$

Link Recommendation Model

$\Longrightarrow$ The upper bound of GAUC loss is written as the following objective :

$$\mathcal{Q}(U,V) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{s=1}^{n} \max\Big(0, U_i^T(V_s - V_j) + 1)\Big) \cdot$$
$$I(X_{ij} = 1, X_{is} \neq 1) + \lambda_U \sum_i U_i^T U_i +$$
$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{s=1}^{n} \max\Big(0, U_i^T(V_j - V_s) + 1)\Big) \cdot$$
$$I(X_{ij} = -1, X_{is} \neq -1) + \lambda_V \sum_j V_j^T V_j$$

where the second and fourth terms are regularization terms used for preventing over-fitting λ_U and λ_V are two hyper-parameters for controlling the scale of regularization terms

Optimizing the Generalized AUC

$$\frac{\partial Q(U,V)}{\partial U_i} = \begin{cases} \sum_{j=1}^{n}\sum_{s=1}^{n}(V_s - V_j) + \lambda_U U_i, & \text{if } X_{ij} = 1, X_{is} \neq 1, \\ & \text{and } U_i^T(V_s - V_j) > -1; \\ \sum_{j=1}^{n}\sum_{s=1}^{n}(V_j - V_s) + \lambda_U U_i, & \text{if } X_{ij} = -1, X_{is} \neq -1, \\ & \text{and } U_i^T(V_j - V_s) > -1; \\ \lambda_U U_i, & \text{otherwise} \end{cases}$$

$$\frac{\partial Q(U,V)}{\partial V_s} = \begin{cases} \sum_{i=1}^{n}\sum_{j=1}^{n} U_i + \lambda_V V_s, & \text{if } X_{ij} = 1, X_{is} \neq 1, \\ & \text{and } U_i^T(V_s - V_j) > -1; \\ -\sum_{i=1}^{n}\sum_{j=1}^{n} U_i + \lambda_V V_s, & \text{if } X_{ij} = -1, X_{is} \neq -1, \\ & \text{and } U_i^T(V_j - V_s) > -1; \\ \lambda_V V_s, & \text{otherwise.} \end{cases}$$

$$\frac{\partial Q(U,V)}{\partial V_s} = \begin{cases} \sum_{i=1}^{n}\sum_{j=1}^{n} U_i + \lambda_V V_s, & \text{if } X_{ij} = 1, X_{is} \neq 1, \\ & \text{and } U_i^T(V_s - V_j) > -1; \\ -\sum_{i=1}^{n}\sum_{j=1}^{n} U_i + \lambda_V V_s, & \text{if } X_{ij} = -1, X_{is} \neq -1, \\ & \text{and } U_i^T(V_j - V_s) > -1; \\ \lambda_V V_s, & \text{otherwise.} \end{cases}$$

Pseudo Code

Algorithm Optimization of $\mathcal{Q}(U, V)$

Input: X, U, V, α, number of batches b, number of iterations t, threshold ς, maximum iteration T.
Initialize: set $t = 0$, initialize U_0 and V_0 randomly
repeat

$t = t + 1$;

Calculate $\frac{\partial \mathcal{Q}(U_t, V_t)}{\partial U_t}$

$U_{t+1} = U_t - \alpha \frac{\partial \mathcal{Q}(U_t, V_t)}{\partial U_t}$;

Calculate $\frac{\partial \mathcal{Q}(U_t, V_t)}{\partial V_t}$

$V_{t+1} = V_t - \alpha \frac{\partial \mathcal{Q}(U_t, V_t)}{\partial V_t}$;

until $|\mathcal{Q}(U_{t+1}, V_{t+1}) - \mathcal{Q}(U_t, V_t)| < \varsigma$ or $t > T$

Outline

Experiment

- A fully observed signed social network $X \in \mathbb{R}^{nxn}$ with $X_{i,j} \in \{-1, 0, 1\}$
  - To remove randomly a fraction (80%, 60% and 40%) of positive and negative links and use the rest to form a partially observed network for training (as X_{Train}).
  - The removed links form a test set X_{Test}
- **Parameter Setting :**Three hyper-parameters in our model : λ_U, λ_V and k
  - Set λ_U=λ_V for simplicity and search over the grid of {1,5,10,20,50,100,200} to find the optimal setting for λ_U and λ_V
  - Search over the grid of {10,30,50,70,90} to find the optimal setting for k.
  - Conduct 5 fold cross-validation on X_{train} and to employ the parameter combination which achieves the best average GAUC for test.

Datasets

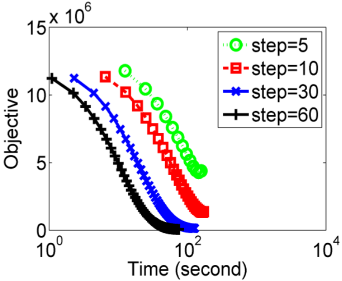
Datasets	Wikipedia	MovieLens	Slashdot
Nodes	7,118	6040/3952	82,144
Edges	103,747	739,012	549,202
+edges	78.78%	77.84%	77.4%
−edges	21.21%	22.16%	22.6%
Density	0.0020	0.0309	0.000081

Evaluation methods :

1. Employ Stochastic Sub-Gradient.
2. Show the GAUC/AUC/MAP and their associated standard deviations of various approaches on three datasets when the size of training set varies from 20% to 60%.
3. Investigate the effectiveness of the proposed approach by comparing its Precision@k and Recall@k with baseline methods.

Efficiency

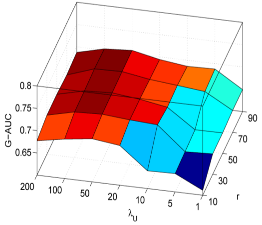
- To study the efficiency of the (OPT+ GAUC) approach : we employ Stochastic Sub-gradient Descent over 20% of the Wikipedia dataset.



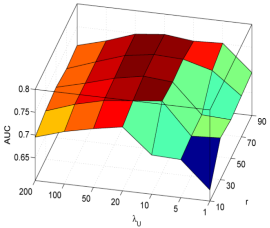
$\Longrightarrow$ When b varies from 5 to 60, the objective function of our proposed approach converges faster.

Parameter sensitivity

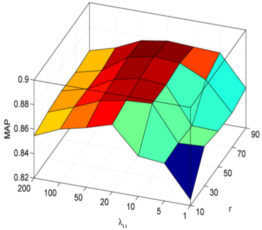
- GAUC on 20%Wikipedia



- AUC on 20%Wikipedia



- MAP on 20%Wikipedia



$\Longrightarrow$ (OPT+GAUC) is very stable as it achieves good GAUC,AUC and MAP when λ_U varies from 10 to 100 and k varies from 30 to 90.

Baslines methods

- Common Neighbor (CN)
- Katz

$\Longrightarrow$ Two approaches obtained based upon the network topological structure

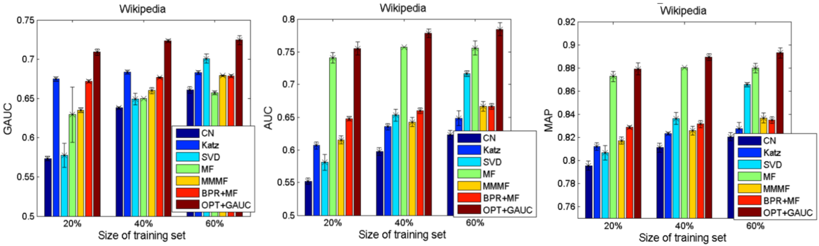
- Singular Value Decomposition (SVD)
- Matrix Factorization (MF)

$\Longrightarrow$ Two representative point-wise approaches for collaborative filtering

- Maximum Margin Matrix Factorization (MMMF)
- Bayesian Personalized Ranking Based upon Matrix Factorization (BPR+MF)

$\Longrightarrow$ Two popular pairwise approaches for personalized ranking

GAUC, AUC and MAP on Wikipedia dataset :



Mean Average Precision : $MAP@k = \frac{1}{N}\sum_{i=1}^{N} AP@k(i)$ with $AP@k = \frac{1}{min(t,k)}\sum_{t=1}^{k} Precision(t)$ *Ref[3]

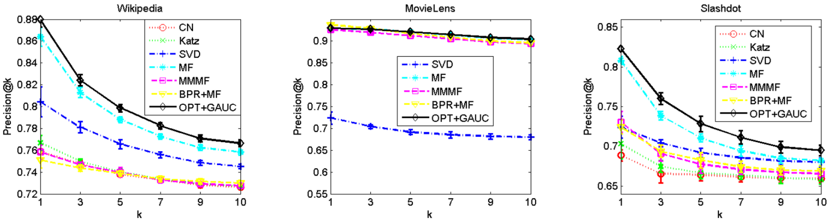
Link recommendation

- For GAUC :
  - Pairwise approaches (MMMF and BPR+MF) outperform pointwise approaches (SVD and MF) in most cases.
  - SVD tends to overfit the data and MF only reconstructs the partially observed network based upon observed positive as well as negative links (it neglects unknown status links).
- For AUC and MAP :
  - SVD cannot perform as well as MF because SVD tends to over-fit the data (especially for smaller datasets).
  - MMMF and BPR+MF are outperformed by MF as they do not directly model negative links.

$\Longrightarrow$ OPT+GAUC outperforms all baseline algorithms regarding GAUC/AUC/MAP

Top-k link recommendation

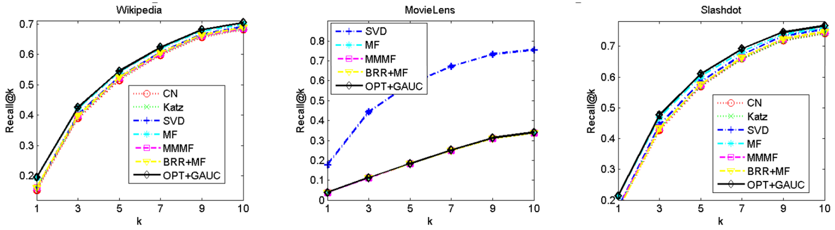
Effectiveness of OPT+GAUC by comparing its Precision@k with baseline methods :



$$Precision@k = \frac{\#\ Positive\ links\ in\ the\ top\ k}{\#\ Positive\ links\ and\ negative\ links\ at\ the\ top\ k}$$

Top-k link recommendation

Effectiveness of OPT+GAUC by comparing its Recall@k with baseline methods :



$$Recall@k = \frac{\#\ Positive\ links\ in\ the\ top\ k}{\#\ Positive\ links}$$

Outline

Conclusion

- Generalized AUC (GAUC) : the new proposed criteria to quantify the ranking performance in signed networks.
- A link recommendation model derived by directly minimizing this loss and introducing an optimization procedure.

$\Longrightarrow$ Results of the experimental studies demonstrated the effectiveness and efficiency of the proposed approach

BUT what about GAUC quadratic time calculation ?

References

1. wikipedia.Receiver_operating_characterictic
2. Cortes, C., and Mohri, M. 2004. Auc optimization vs. error rate minimization. In Proceedings of Advances in Neural Information Processing Systems, 537–546. MIT Press.
3. wikipedia.Information_retrieval
4. Menon, A. K., and Elkan, C. 2011. Link prediction via matrix factorization. In Proceedings of the European Conference on Machine Learning, Part II, LNAI2912, 437–452.
5. Evaluation Metrics, Jaime Arguello, INLS 509 : Information Retrieval, jarguellemail.unc.edu, March 25, 2013

Paper 3: Efficient Latent Link Recommendation in Signed Networks

Dongjin Song, David A.Meyer, Dacheng Tao (2015)

Turan Bilalov

Outline

Outline

Motivation

- Assumption: A plausible model for user behavior analytics in signed networks is that more extreme positive and negative relationships are explored and exploited before less extreme ones.

- Link recommendation in signed networks: we aim to produce a personalized ranking list with positive links on the top, negative links at the bottom and unknown status links in between.

Outline

Intro

- derive two lower bounds for GAUC which can be computed in linear time.

- develop two linear time probabilistic models, entitled efficient latent link recommendation (ELLR) algorithms

- compare these two ELLR algorithms with top performing baseline approaches over four benchmark datasets

Outline

Notation

Let $X \in \mathbb{R}^{n\times m}$ be an n by m matrix; we use $X_j \in \mathbb{R}^n$ to represent its j-th column, which is a n dimensional vector, and use X_{ij} to denote the entry in its i-th row and j-th column. $\|X\|_F = \sqrt{\mathrm{Tr}(XX^T)}$ denotes the Frobenius norm of the matrix X, where $\mathrm{Tr}(XX^T) = \sum_{i=1}^{n}\sum_{j=1}^{m} X_{ij}^2$ represents the trace of an n by n square matrix XX^T.

Outline

Lower Bound-I

THEOREM 1. *GAUC for user i is lower bounded by:*

$$GAUC(i) \geq \frac{1}{|\mathcal{P}| + |\mathcal{N}|} \Bigg(\sum_{(i,j)\in\mathcal{P}} \prod_{(i,s)\in\mathcal{O}\bigcup\mathcal{N}} I(\widehat{X}_{ij} > \widehat{X}_{is}) + \sum_{(i,j)\in\mathcal{N}} \prod_{(i,t)\in\mathcal{O}\bigcup\mathcal{P}} I(\widehat{X}_{ij} < \widehat{X}_{it}) \Bigg).$$

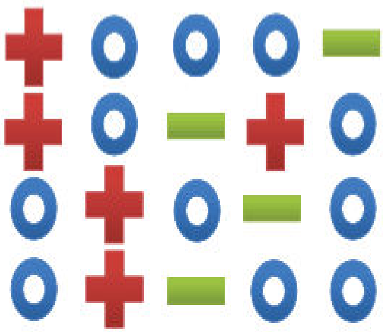
with equality holding if within each product operator the condition for each indicator function is jointly satisfied or jointly not satisfied.

PROPOSITION 1. *GAUC's lower bound in Theorem 1 is equivalent to*

$$\frac{1}{|\mathcal{P}| + |\mathcal{N}|} \Bigg(\sum_{(i,j)\in\mathcal{P}} I(\widehat{X}_{ij} > \max_{(i,s)\in\mathcal{O}\bigcup\mathcal{N}} (\widehat{X}_{is})) + \sum_{(i,j)\in\mathcal{N}} I(\widehat{X}_{ij} < \min_{(i,t)\in\mathcal{O}\bigcup\mathcal{P}} (\widehat{X}_{it})) \Bigg),$$

which can be calculated in linear time.

Lower Bound-II

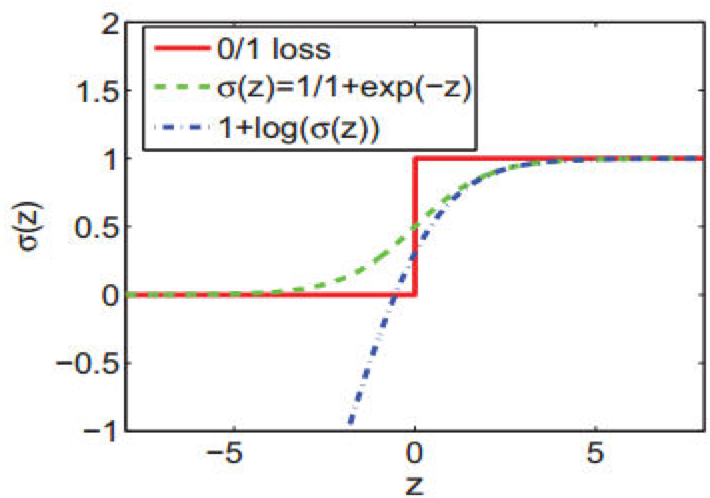


AUC	GAUC	Bound-I	Bound-II
1	1	1	1
2/3	11/17	1/3	0
3/4	3/4	0	0
3/4	5/8	0	0

THEOREM 2. *GAUC and its lower bound in Proposition 1 can be further bounded by:*

$$GAUC(i) \geq \frac{|\mathcal{P}|}{|\mathcal{P}|+|\mathcal{N}|} \prod_{(i,j)\in\mathcal{P}} I(\widehat{X}_{ij} > \max_{(i,s)\in\mathcal{O}\bigcup\mathcal{N}} (\widehat{X}_{is})) + \frac{|\mathcal{N}|}{|\mathcal{P}|+|\mathcal{N}|} \prod_{(i,j)\in\mathcal{N}} I(\widehat{X}_{ij} < \min_{(i,t)\in\mathcal{O}\bigcup\mathcal{P}} (\widehat{X}_{it})),$$

with equality holding if within each product operator the condition for each indicator function is jointly satisfied.



PROPOSITION 2. *GAUC's lower bound in Theorem 2 is equivalent to*

$$\frac{|\mathcal{P}|}{|\mathcal{P}|+|\mathcal{N}|} I(\min_{(i,j)\in\mathcal{P}} (\widehat{X}_{ij}) > \max_{(i,s)\in\mathcal{O}\bigcup\mathcal{N}} (\widehat{X}_{is})) + \frac{|\mathcal{N}|}{|\mathcal{P}|+|\mathcal{N}|} I(\max_{(i,j)\in\mathcal{N}} (\widehat{X}_{ij}) < \min_{(i,t)\in\mathcal{O}\bigcup\mathcal{P}} (\widehat{X}_{it})),$$

which can be calculated in linear time.

Outline

Problem Statement

- Let $X \in \{1, -1, ?\}^{n \times m}$ be a partially observed signed network in which ±1 denote an observed positive/negative link and ? denotes an unknown status link which could be either.

- *X* can be approximated with two low rank matrices $U \in R^{r \times n}$ and $V \in R^{r \times m}$

- Our aim can be recast as learning the following ranking function

$$f(U, V, i, j) = f_{ij}(U, V) = U_i^T V_j,$$

- When X is a symmetric signed network, $X = X^T$ we can set U = V for simplicity.

Efficient Latent Link Recommendation-I

- ELLR-I is formulated as a Bayesian model aiming to produce the correct personalized ranking lists based upon pairwise comparisons between positive links and the latent or negative link which has the largest ranking score.

- ELLR-I maximizes the following posterior distribution:

$$P(U,V|>_f,X) \propto P(>_f,X|U,V)P(U)P(V),$$

- Assuming that each user is acting independently and each pair of users' (or user and item's) ranking scores is compared independently, the right hand side of becomes:

$$\begin{aligned}
&P(>_f,X|U,V)P(U)P(V)\\
&=\Pi_i\Pi_{(i,j)\in\mathcal{P}\cup\mathcal{N}}\Pi_{(i,s)\in C_{i,j,s}}P(>_f,X_{ij},X_{is}|U_i,V_j,V_s)P(U_i)P(V_j)\\
&=\Pi_i\Pi_{(i,j)\in\mathcal{P}}\Big(\Pi_{(i,s)\in\mathcal{O}\cup\mathcal{N}}P(>_f,X_{ij}=1,X_{is}\neq 1|U_i,V_j,V_s)\Big)\cdot\\
&\quad\Pi_i\Pi_{(i,j)\in\mathcal{N}}\Big(\Pi_{(i,s)\in\mathcal{O}\cup\mathcal{P}}P(>_f,X_{ij}=-1,X_{is}\neq -1|U_i,V_j,V_s)\Big)\cdot\\
&\quad P(U_i)P(V_j)
\end{aligned}$$

Efficient Latent Link Recommendation-I

- Objective of ELLR-I:

$$
\begin{aligned}
L_{\text{ELLR-I}}&(U, V) \\
&= \log P(U, V| >_f, X) \\
&= \sum_{i=1}^{n} \sum_{(i,j)\in\mathcal{P}} \log\left(\sigma(U_i^T V_j - \max_{(i,s)\in\mathcal{O}\bigcup\mathcal{N}} U_i^T V_s)\right) \\
&\quad + \sum_{i=1}^{n} \sum_{(i,j)\in\mathcal{N}} \log\left(\sigma(-U_i^T V_j + \min_{(i,s)\in\mathcal{O}\bigcup\mathcal{P}} U_i^T V_s)\right) \\
&\quad - \frac{\lambda_U}{2} \sum_{i=1}^{n} U_i^T U_i - \frac{\lambda_V}{2} \sum_{j=1}^{n} V_j^T V_j + c,
\end{aligned}
$$

Efficient Latent Link Recommendation-II

- ELLR-II can be formulated as a Bayesian probabilistic model aiming to produce the correct personalized ranking list based upon the fact that the positive link with the smallest ranking score should be larger than the latent or negative link which has the largest ranking score.

- Table 1: Detailed statistics of the four datasets. Note that MovieLens10M is a bipartite network with 71,567 users and 10,681 items.

Datasets	Wikipedia	Slashdot	Epinions	MovieLens10M
Nodes	7,118	82,144	119,217	71,567/10,681
Edges	103,747	549,202	841,372	7,643,378
+edges	78.78%	77.4%	85.0%	77.0%
−edges	21.21%	22.6%	15.0%	23.0%

Efficient Latent Link Recommendation-II

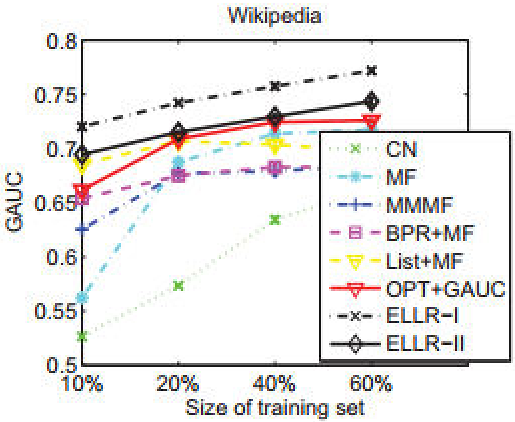
- Objective of ELLR-II:

$$
\begin{aligned}
&L_{\mathrm{ELLR-II}}(U,V) \\
&\quad = \log P(U,V| >_f, X) \\
&\quad = \sum_{i=1}^{n} \log\left(\sigma\Big(\min_{(i,j)\in\mathcal{P}} U_i^T V_j - \max_{(i,s)\in\mathcal{O}\bigcup\mathcal{N}} U_i^T V_s\Big)\right) \\
&\qquad + \sum_{i=1}^{n} \log\left(\sigma\Big(-\max_{(i,j)\in\mathcal{N}} U_i^T V_j + \min_{(i,s)\in\mathcal{O}\bigcup\mathcal{P}} U_i^T V_s\Big)\right) \\
&\qquad - \frac{\lambda_U}{2}\sum_{i=1}^{n} U_i^T U_i - \frac{\lambda_V}{2}\sum_{j=1}^{n} V_j^T V_j + c,
\end{aligned}
$$

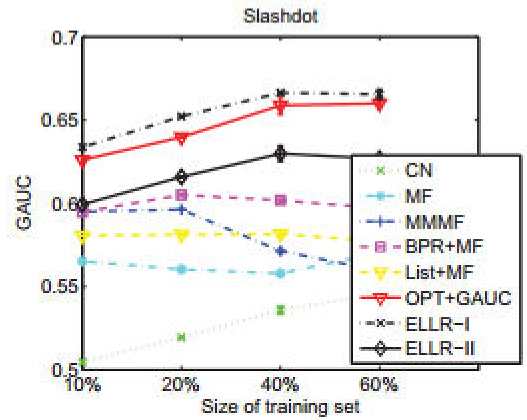
Optimization

- The computational complexity for a full gradient of ELLR-I over U or V is around O(anpr) where a is the average number of positive and negative links for each user in the network, and $r \ll n$ is the rank.
- When a is very large, computation of a full gradient of ELLR-I may be infeasible. In this case, ELLR-II can be used since the computational complexity for a full gradient of ELLR-II over U or V is only around O(qnpr) where $q \leq a$, p, and r are relatively small and fixed.
- To further reduce training time of ELLR-I and ELLR-II, we can sample a subset of unknown status links and use stochastic gradient ascent to train these two models.

GAUC on Datasets

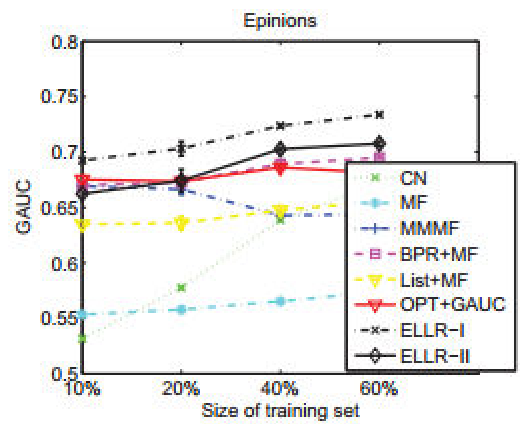


(a) GAUC on Wikipedia

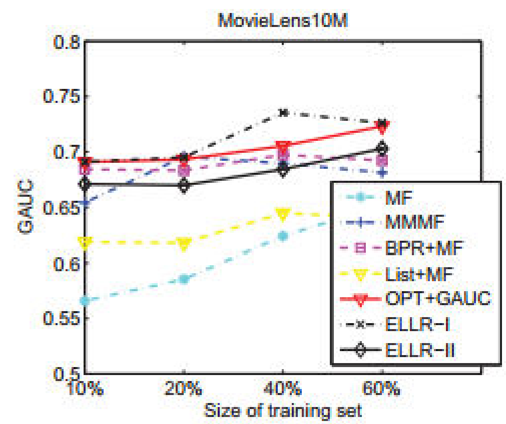


(b) GAUC on Slashdot

Figure 1

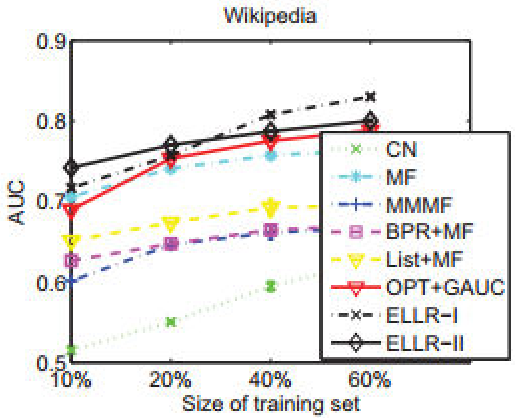


(c) GAUC on Epinions

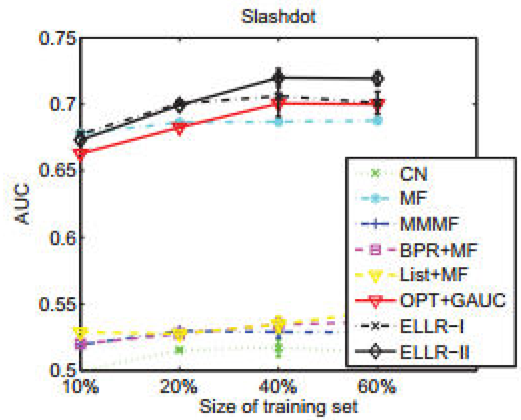


(d) GAUC on MovieLens

AUC on Datasets

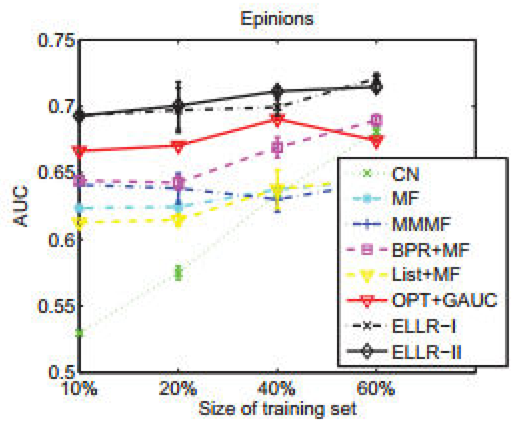


(a) AUC on Wikipedia

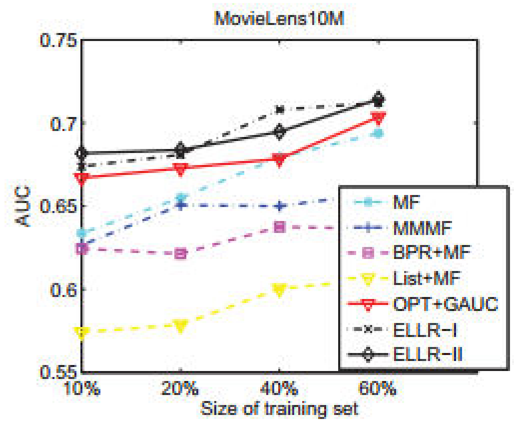


(b) AUC on Slashdot

Figure 2

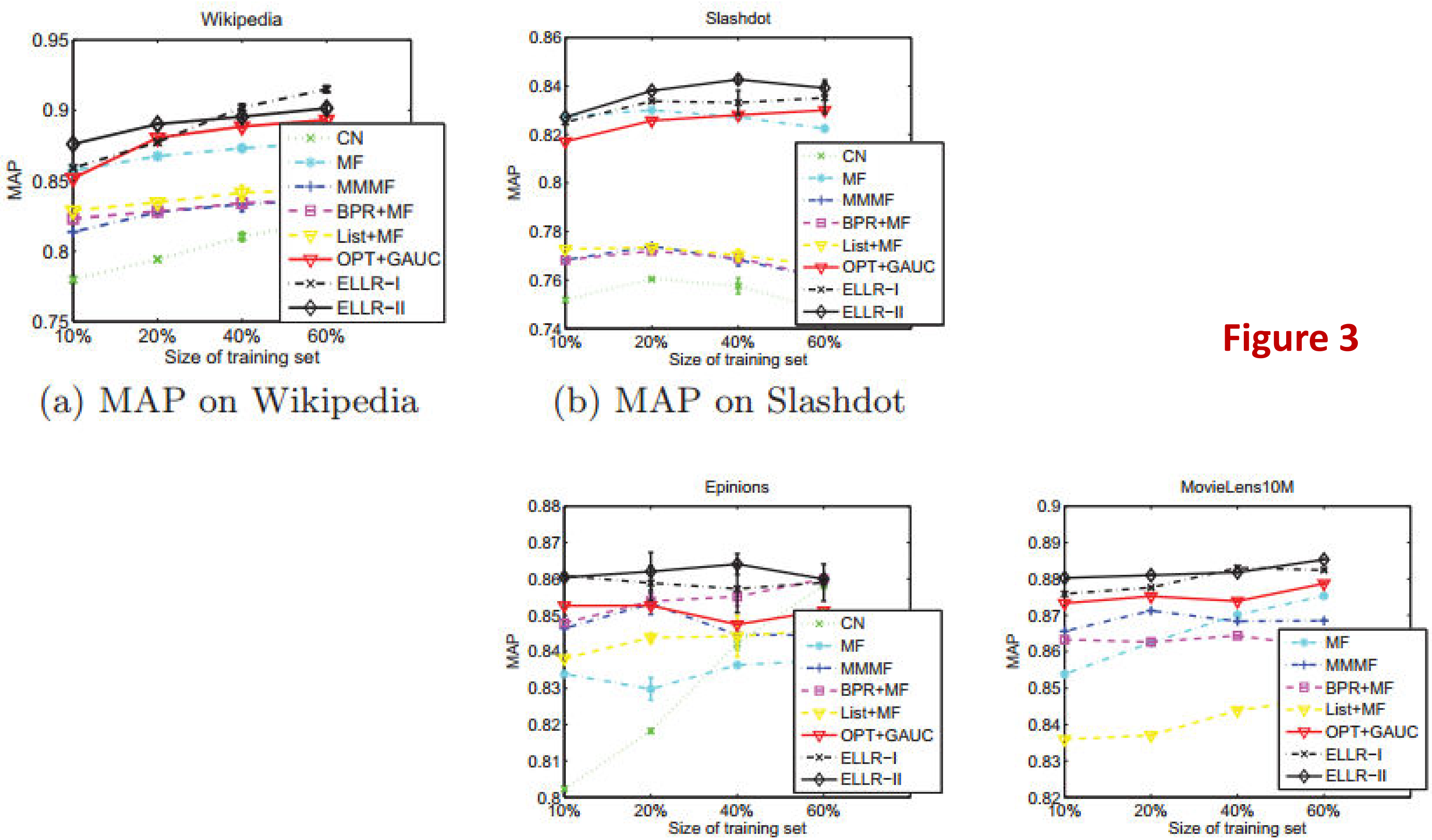


(c) AUC on Epinions



(d) AUC on MovieLens

MAP on Datasets



(a) MAP on Wikipedia (b) MAP on Slashdot

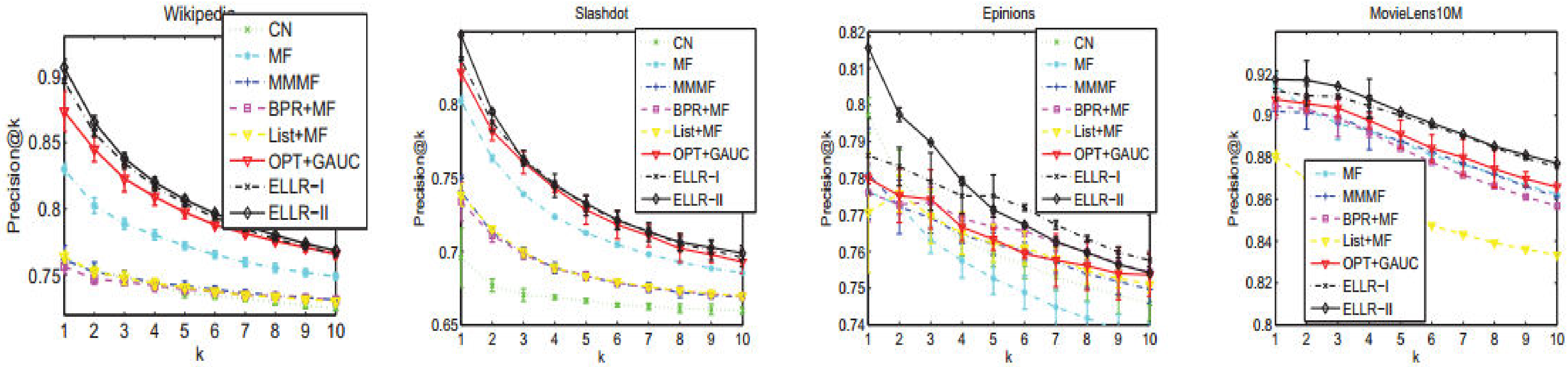
(c) MAP on Epinions (d) MAP on MovieLens

Figure 3

Outline

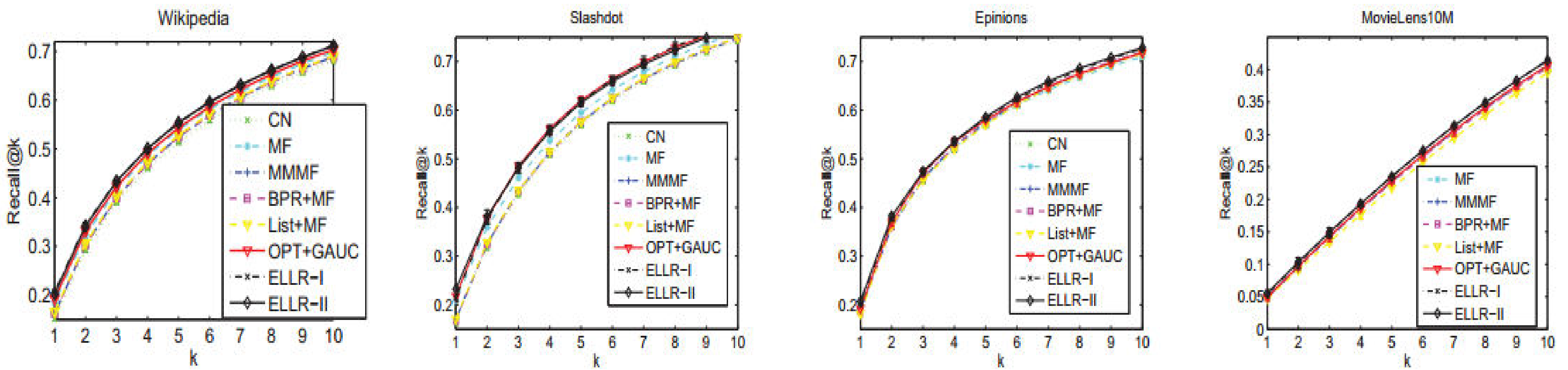
Setup and Evaluation

Given a fully observed signed network $X \in \mathcal{R}^{n\times n}$ in which $X_{ij} \in \{-1, 0, 1\}$, $X_{ij} = 1$ denotes that the i-th user trusts (or likes) the j-th user and $X_{ij} = -1$ denotes that the i-th user distrusts (or dislikes) the j-th user. We randomly select a fraction (*e.g.*, 10%, 20%, 40%, 60%) of the observed positive and negative links (as X_{Train}) for training, and evaluate over a test set (*i.e.*, X_{Test}) comprising the remaining non-zero entries. The zero entries in X_{Train} are called *latent* links since each link has the potential to either be a positive or a negative link.



(a) P@k on Wikipedia (b) P@k on Slashdot (c) P@k on Epinions (d) P@k on MovieLens

Figure 4



(a) Recall@k on Wikipedia (b) Recall@k on Slashdot (c) Recall@k on Epinions (d) Recall@k on MovieLens

Figure 5

Results-I: Effectiveness

Link Recommendation

- Figures 1, 2 and 3 show the GAUC/AUC/MAP and their associated standard deviations over four datasets when the size of training set varies from 10% to 60%.

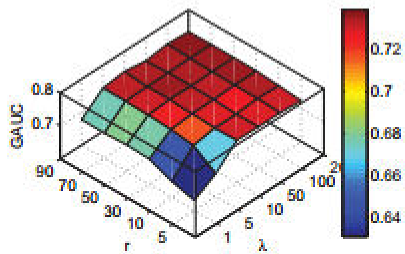
Top k Link Recommendation:

- We study the effectiveness of two proposed approaches by comparing their Precision@k and Recall@k with baseline methods when the size of training set is 40% for Wikipedia, Slashdot, Epinions, and MovieLens10M. In Figure 4 and 5, we observe that ELLR-I and ELLR-II consistently outperform the baseline approaches.

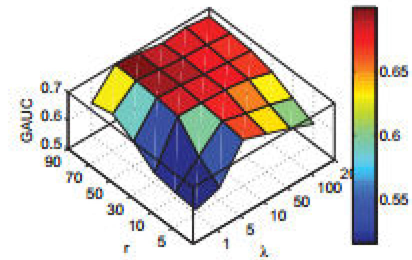
Results-II: Parameter Sensitivity

- We investigate the sensitivity of ELLR-I and ELLR-II with respect to the regularization parameters $\lambda U = \lambda V \in \{1, 5, 10, 50, 100, 200\}$ and $r \in \{5, 10, 30, 50, 70, 90\}$ for the Wikipedia (20%) dataset.
- When we vary the value of λU or r, we keep the other parameters fixed.
- We plot the GAUC/AUC/ MAP with respect to λU or r in Figure 9.
- We observe that both ELLR-I and ELLR-II are very stable and they achieve good performance when λU varies from 10 to 200 and r varies from 10 to 90.

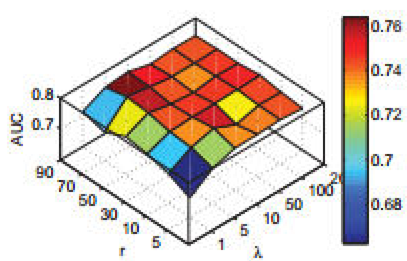
Results-II: Parameter Sensitivity



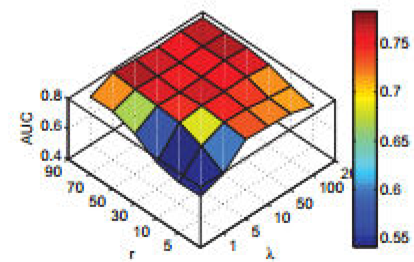
(a) GAUC for ELLR-I.



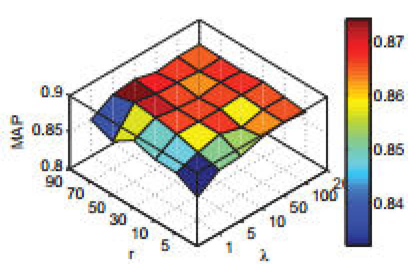
(b) GAUC for ELLR-II.



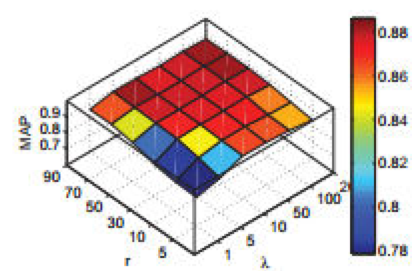
(c) AUC for ELLR-I.



(d) AUC for ELLR-II.



(e) MAP for ELLR-I.



(f) MAP for ELLR-II.

Outline

Conclusion

- We derived two lower bounds for GAUC which can be computed in linear time.

- We compared ELLR-I and ELLR-II with top-performed baseline approaches over four benchmark datasets; our experimental results demonstrate that the proposed ELLR algorithms outperform state-of-the-art methods for link recommendation in signed networks with no loss of efficiency.

Outline

REFERENCES

1. L. A. Adamic and E. Adar. Friends and neighbors on the web. Social Networks, 25(3):211 – 230, 2003.
2. L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social network. In Proceedings of the 4th ACM international Conference on Web Search and Data Mining, pages 635–644, Hong Kong, China, 2011.
3. Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Ho. Adapting ranking svm to document retrieval. In Proceedings of the 29th SIGIR Conference on Research and Development, 2006
4. C. J. Hsieh, K. Y. Chiang, and I. S. Dhillon. Low rank modeling of signed network. In ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Beijing, China, 2012.
5. J. Lee, S. Bengio, S. Kim, G. Lebanon, and Y. Singer. Local collaborative filtering. In Proceedings of the 23th International Conference on World Wide Web, 2014.
6. A. K. Menon and C. Elkan. Link prediction via matrix factorization. In Proceedings of the European Conference on Machine Learning, Part II, LNAI2912, pages 437–452, 2011.

Comparison

Comparison of the papers

Complementary

Previous Papers ⟷ **Paper 1**

- Unsigned Social Networks :
  - Exploiting only positive links
  - Exploiting only local information

- Signed Social Networks :
  - Exploiting positive and negative links
  - Exploiting global and local information

$\Longrightarrow$ **Use of Matrix Factorization as basic model.**
RecSSN outperforms previous methods.

Comparison of the papers

Complementary

Paper 2 ⟷ **Paper 3**

Paper 2

- Link recommendation model by optimizing the loss of GAUC
- measure both the head and tail of a ranking list but the calculation of GAUC requires quadratic time.

Paper 3

- derive two lower bounds of GAUC which can be computed in linear time
- Two Efficient Latent Link Recommendation (ELLR) algorithms by optimizing the two lower bounds.

$\Longrightarrow$ **The approach proposed in Paper 3 outperforms the approach in Paper2 terms of effectiveness (GAUC/AUC/MAP/*Precision@k*/*Recall@k*) and efficiency(time training).**

Questions

BACKUP Slides

Similarity measures

- In this work, we consider three ways of calculating s_i^p :
  - CI : s_i^p is the number of common items scored by both u_i and FR_i :

$$s_i^p = |\mathcal{I}_i^p|, \quad \mathcal{I}_i^p = \{v_j | \mathbf{R}_{ij} \neq 0 \wedge \bar{\mathbf{R}}_j^{ip} \neq 0\}$$

  - COSINE : s_i^p is calculated as cosine similarity of scores between u_i and FR_i over all items :

$$s_i^p = \frac{\sum_{v_j} \mathbf{R}_{ij} \cdot \bar{\mathbf{R}}_j^{ip}}{\sqrt{\sum_{v_j} \mathbf{R}_{ij}^2} \sqrt{\sum_{v_j} (\bar{\mathbf{R}}_j^{ip})^2}}$$

  - CI-COSINE : computes the cosine similarity over common items I_i^p :

$$s_i^p = \frac{\sum_{v_j \in \mathcal{I}_i^p} \mathbf{R}_{ij} \cdot \bar{\mathbf{R}}_j^{ip}}{\sqrt{\sum_{v_j \in \mathcal{I}_i^p} \mathbf{R}_{ij}^2} \sqrt{\sum_{v_j \in \mathcal{I}_i^p} (\bar{\mathbf{R}}_j^{ip})^2}}$$

Where $\bar{R}_j^{ip}$ is the average score of FR_i to the j-th item