

# Efficient Latent Link Recommendation in Signed Networks

Dongjin Song  
Department of ECE,  
University of California,  
San Diego  
9500 Gilman Dr., La Jolla,  
CA, 92093-0409, USA  
dosong@ucsd.edu

David A. Meyer  
Department of Mathematics,  
University of California,  
San Diego  
9500 Gilman Dr., La Jolla,  
CA, 92093-0112, USA  
dmeyer@math.ucsd.edu

Dacheng Tao  
Centre for Quantum Comp. &  
Intelligent Sys.  
Faculty of Engineering and IT  
University of Technology,  
Sydney  
81 Broadway Street, Ultimo,  
NSW 2007, Australia.  
dacheng.tao@uts.edu.au

## ABSTRACT

Signed networks, in which the relationship between two nodes can be either positive (indicating a relationship such as trust) or negative (indicating a relationship such as distrust), are becoming increasingly common. A plausible model for user behavior analytics in signed networks can be based upon the assumption that more extreme positive and negative relationships are explored and exploited before less extreme ones. Such a model implies that a personalized ranking list of latent links should place positive links on the top, negative links at the bottom, and unknown status links in between. Traditional ranking metrics, *e.g.*, area under the receiver operating characteristic curve (AUC), are however not suitable for quantifying such a ranking list which includes positive, negative, and unknown status links. To address this issue, a generalized AUC (GAUC) which can measure both the head and tail of a ranking list has been introduced. Since GAUC weights each pairwise comparison equally and the calculation of GAUC requires quadratic time, we derive two lower bounds of GAUC which can be computed in linear time and put more emphasis on ranking positive links on the top and negative links at the bottom of a ranking list. Next, we develop two efficient latent link recommendation (ELLR) algorithms in order to recommend links by directly optimizing these two lower bounds, respectively. Finally, we compare these two ELLR algorithms with top-performing baseline methods over four benchmark datasets, among which the largest network has more than 100 thousand nodes and seven million entries. Thorough empirical studies demonstrate that the proposed ELLR algorithms outperform state-of-the-art approaches for link recommendation in signed networks at no cost in efficiency.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).  
KDD '15, August 10-13, 2015, Sydney, NSW, Australia.  
© 2015 ACM. ISBN 978-1-4503-3664-2/15/08 ...\$15.00.  
DOI: <http://dx.doi.org/10.1145/2783258.2783358>.

## Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information Filtering

## Keywords

Recommender systems; signed networks; link recommendation; GAUC

## 1. INTRODUCTION

In the past few years, social networking websites, *e.g.*, Facebook, Twitter, LinkedIn, *etc.*, as well as online E-commerce websites such as Amazon and Ebay, have become increasingly common in people's daily life and have dramatically reshaped people's social behavior as well as their consumption habits. Thus a considerable amount of effort has been devoted to investigating their underlying social mechanisms so as to enhance user experience [17, 20, 29]. For example, link prediction [17, 18] and recommendation [2] are two fundamental problems which can help users connect to entities (*e.g.*, people, items, *etc.*) in which they are interested.

Specifically, given a set of potential links, link prediction is essentially a binary classification problem which aims to indicate the presence or absence of these links using either explicit network topological structure [1, 10, 17] (*e.g.*, common friends) or latent features [21, 18, 14]. Link recommendation, which treats the same problem 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).

Recently, signed networks, *i.e.*, networks in which the relationship between two nodes can be either positive (indicating a relation such as trust) or negative (indicating a relation such as distrust), have become increasingly common. For instance, in Epinions [7], which is a product review website with an active user community, users can indicate whether they trust or distrust each other based upon their reviews; in Slashdot [3, 13], a technology related news website, users can tag each other as "friend" or "foe" based upon their comments. A plausible model for user behavior in signed networks can be based upon the assumption

that more extreme positive and negative relationships are explored and exploited before less extreme relations. Such a model implies that a personalized ranking list of latent links should place positive links on the top, negative links at the bottom, and unknown status links in between.

Traditional ranking measures, *e.g.*, area under the ROC curve (AUC) [8], are not suitable to quantify such a ranking list in signed networks since they only apply to the binary case, rather than to the triplet (positive, negative, and unknown). Although a generalized AUC (GAUC) [26] which can measure both the head and tail of a ranking list has been introduced, it does not explicitly enforce that the links ranked on the top are all positive and the links ranked at the bottom are all negative. Moreover, the calculation of GAUC requires quadratic time, which is computationally intractable for large-scale networks.

In this paper, we aim to tackle both issues. The contributions include:

- we derive two lower bounds for GAUC which can be computed in linear time. The first quantifies the fraction of positive and negative links which are ranked at the optimal positions (*i.e.*, positive links on the top and negative links at the bottom); the second is more strict and it measures whether all the positive links are ranked on the top and whether all the negative links are ranked at the bottom of a ranking list.
- we develop two linear time probabilistic models, entitled efficient latent link recommendation (ELLR) algorithms, to infer personalized ranking lists of latent links by directly optimizing these two lower bounds, respectively.
- we compare these two ELLR algorithms with top performing baseline approaches over four benchmark datasets, among which the largest network has more than 100 thousand nodes and seven million entries. The 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.

The rest of this paper is organized as follows. In Section 2, we summarize related work. In Section 3, we introduce GAUC and derive its two lower bounds. In Section 4, we present two efficient latent link recommendation (ELLR) algorithms and introduce the optimization procedure. In Section 5, we conduct experiments to demonstrate the effectiveness and efficiency of ELLR algorithms. In Section 6, we draw some conclusions.

## 1.1 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.  $\|X\|_F = \sqrt{\text{Tr}(XX^T)}$  denotes the Frobenius norm of the matrix  $X$ , where  $\text{Tr}(XX^T) = \sum_{i=1}^n \sum_{j=1}^m X_{ij}^2$  represents the trace of an  $n$  by  $n$  square matrix  $XX^T$ .

## 2. RELATED WORK

Various approaches have been developed to recommend links in social networks. In general, these approaches fall

into two main categories: network topology-based approaches and latent feature-based approaches.

### 2.1 Network topology-based approaches

Network topology-based approaches can be further divided into two sub-categories: neighbor-based approaches and path-based approaches. Neighbor-based approaches, *e.g.*, common neighbors, Jaccard's coefficient [23], Adamic and Adar [1], *etc.*, recommend links based upon their neighborhood structure. Path-based methods produce ranking scores by considering the ensemble of all paths between two nodes. For instance, Katz [10] computes the sum over all paths, of their lengths, exponentially damped to count short paths more heavily, *i.e.*,

$$f(i, j) = \sum_{l=1}^{\infty} \beta^l |X_{i,j}^{(l)}|, \quad (1)$$

where  $X \in \{0, 1\}^{n \times n}$  is the adjacency matrix,  $X_{i,j}^{(l)}$  denotes the set of all length  $l$  paths from  $i$  to  $j$ , and  $0 < \beta < 1$  is a parameter which controls the damping. Other examples include PageRank [17], supervised random walks [2], the model of consistent node types [25], *etc.*

These approaches, however, may not perform well when little topological information is available for nodes (*i.e.*, for example, when a node has few direct connections or high-order connections to other nodes). To handle this situation, latent feature-based approaches have been developed.

### 2.2 Latent feature-based approaches

Latent feature-based approaches aim to learn a low rank model which can recover the values, or the relative ordering of the values, of entries in the (weighted) adjacency matrix associated to a network. These approaches can be further divided into pointwise methods, pairwise methods, and listwise methods.

#### 2.2.1 Pointwise methods

Pointwise methods [21, 22, 11, 12, 6, 9] treat link recommendation as a matrix completion problem and reconstruct the adjacency matrix of a partially observed social network from a low rank model. It has been shown that pointwise methods can outperform various neighborhood models as well as the SVD++ [11] for collaborative filtering and can be employed to perform top- $k$  recommendation. Specifically, in a matrix factorization model [12][27], all the observed entries in the adjacency matrix  $X \in \{0, 1\}^{n \times n}$  are reconstructed with

$$\hat{X} = U^T V + B, \quad (2)$$

where  $\hat{X} \in \mathbb{R}^{n \times n}$  is the approximated adjacency matrix,  $U \in \mathbb{R}^{r \times n}$ ,  $V \in \mathbb{R}^{r \times n}$ ,  $r \ll n$  is the rank, and  $B \in \mathbb{R}^{n \times n}$  is the offset. This model learns the latent features (parameters) by minimizing the squared error:

$$\sum_i \sum_j M_{ij} (X_{ij} - \hat{X}_{ij})^2 + \lambda (\|U\|_F^2 + \|V\|_F^2 + \|B\|_F^2), \quad (3)$$

where  $\lambda > 0$  is a regularization parameter and  $M$  is a mask which is 1 if  $X_{ij} > 0$  and is 0 otherwise.

Although pointwise methods have shown their effectiveness for collaborative filtering and  $\hat{X}$  can be employed to perform link recommendation, they do not explicitly model

the relative order of the values in the adjacency matrix and thus may not perform well for link recommendation.

### 2.2.2 Pairwise methods

Pairwise methods [20, 18, 14] treat link recommendation as a learning to rank problem based upon pairwise comparisons. Most of them aim to optimize AUC, which is given by:

$$\text{AUC}(i) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{(i,j) \in \mathcal{P}} \sum_{(i,s) \in \mathcal{N}} I(\hat{X}_{ij} > \hat{X}_{is}) \quad (4)$$

where  $\hat{X}_{ij}$  and  $\hat{X}_{is}$  are predicted ranking scores for the  $i$ -th user,  $\mathcal{P}$  and  $\mathcal{N}$  are the sets of positive and negative, respectively, links in a ranking list, and  $|\cdot|$  denotes set cardinality.  $I(\cdot)$  is an indicator function which is 1 if the condition in the parenthesis is satisfied and is 0 otherwise. For instance, Rendle et al. [20] employ a smooth approximation of the indicator function to perform personalized recommendation; Menon and Elkan [18] employ a hinge loss to relax the indicator in (4) and recommend links. Lee et. al. [14] employ the local structures of the networks as well as surrogate losses to perform the recommendation task.

The AUC, however, is not suitable for quantifying a ranking list in signed networks which include positive, negative, and unknown status links since it only applies to the binary case.

### 2.2.3 Listwise methods

Listwise methods [5, 31, 24] 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. For instance, listMLE [31] employs the likelihood loss to perform information retrieval, which could be adapted to perform link recommendation; listwise learning to rank with matrix factorization (List+MF) [24] employs the cross entropy loss to perform collaborative filtering and item recommendation.

Although most existing approaches can be adapted or directly employed to recommend latent links in signed networks by treating positive links as positive samples and the others (including negative links and unknown status links) as negative samples, they may not perform well because their objectives conflate types of links. The goal of latent link recommendation should be to order a list of latent links (positive, negative, and unknown status) so that positive links are on the top and negative links are at the bottom. Traditional measures, *e.g.*, area under the ROC curve (AUC), which are useful for binary rankings, cannot handle this situation properly because they are not sensitive to the position of negative links in the ranking. Therefore, a generalized AUC (GAUC), which can measure both the head and tail of a ranking list, has been introduced [26]. Although directly optimizing GAUC with hinge loss has shown its effectiveness for positive link recommendation in signed networks [26], we note that GAUC does not explicitly enforce that the links ranked on the top are all positive and the links ranked at the bottom are all negative. Furthermore, the calculation of GAUC requires quadratic time which is computationally intractable for large-scale networks. Therefore, this paper presents two lower bounds for GAUC to address these issues and shows that they can be optimized in linear time to

achieve state-of-the-art latent link recommendation performance in signed networks.

## 3. GAUC AND ITS LOWER BOUNDS

In this section, we first introduce the generalized AUC for measuring the performance of latent link recommendation in signed networks. Then we derive two lower bounds for GAUC which can be calculated in linear time and explicitly put more emphasis on ranking the positive links on the top and negative links at the bottom of a ranking list.

### 3.1 Generalized AUC

Given a set of latent links which includes potential positive, negative, and unknown status links, an ideal personalized ranking list in a signed network ranks positive links (indicating a relationship such as trust) on the top, negative links (indicating a relationship such as distrust) at the bottom, and unknown status links in the middle. Traditional ranking measures, *e.g.*, AUC cannot completely quantify closeness to such a ranking list because they only handle two types. One could use the AUC, treating positive links as positive samples and the other links as negative samples, but then it would be insensitive to the positions of negative links and thus would not quantify the ranking quality of negative links. Although mean average precision (MAP) and normalized discounted cumulative gain (NDCG) can be used in signed networks, they may not perform well since they tend to overestimate the positive links on the top and thus cannot quantify the negative links at the bottom of a ranking list appropriately. To solve this problem, a generalized AUC (GAUC) was introduced as follows [26]:

**DEFINITION 1.** Assuming  $X \in \{1, -1, ?\}^{n \times m}$  is a partially observed signed network, given a ranking list for user  $i$ , 1 denotes an observed positive link in set  $\mathcal{P}$ ,  $-1$  represents an observed negative link belong to set  $\mathcal{N}$ ,  $?$  denotes an unknown status link within set  $\mathcal{O}$ . Then the GAUC for user  $i$  is defined as:

$$\begin{aligned} \text{GAUC}(i) = & \frac{1}{|\mathcal{P}| + |\mathcal{N}|} \cdot \\ & \left( \frac{1}{|\mathcal{O}| + |\mathcal{N}|} \sum_{(i,j) \in \mathcal{P}} \sum_{(i,s) \in \mathcal{O} \cup \mathcal{N}} I(\hat{X}_{ij} > \hat{X}_{is}) \right. \\ & \left. + \frac{1}{|\mathcal{O}| + |\mathcal{P}|} \sum_{(i,j) \in \mathcal{N}} \sum_{(i,t) \in \mathcal{O} \cup \mathcal{P}} I(\hat{X}_{ij} < \hat{X}_{it}) \right), \end{aligned} \quad (5)$$

where  $\hat{X}_{ij}$ ,  $\hat{X}_{is}$ , and  $\hat{X}_{it}$  are predicted ranking scores for the  $i$ -th user. The first term and the second term quantify the ranking performance of positive links and negative links, respectively.

As is AUC, GAUC is 1 for a perfect ranking list and is 0.5 for a randomly ordered ranking list. The main difference between them is that GAUC jointly quantifies the ranking quality over positive links and negative links, in the presence of unknown status links. GAUC also differs from other variants of AUC [19, 16] by focusing on the head and tail of a ranking list.

Although GAUC has shown its effectiveness for positive link recommendation in signed networks [26], we note that GAUC does not explicitly enforce that the links ranked on the top are all positive and the links ranked at the bottom

are all negative. Furthermore, the calculation of GAUC requires quadratic time which may make the underlying optimization problem intractable for large-scale networks. Therefore, two lower bounds of GAUC are derived to resolve these issues.

### 3.2 Lower Bound-I

Since GAUC considers each pairwise comparison equally, it does not explicitly quantify the fraction of positive and negative links which achieve the optimal ranking (positive links on the top and negative links at the bottom) in a ranking list; for this purpose, we derive a lower bound of GAUC.

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

$$GAUC(i) \geq \frac{1}{|\mathcal{P}| + |\mathcal{N}|} \left( \sum_{(i,j) \in \mathcal{P}} \prod_{(i,s) \in \mathcal{O} \cup \mathcal{N}} I(\hat{X}_{ij} > \hat{X}_{is}) + \sum_{(i,j) \in \mathcal{N}} \prod_{(i,t) \in \mathcal{O} \cup \mathcal{P}} I(\hat{X}_{ij} < \hat{X}_{it}) \right). \quad (6)$$

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

Theorem 1 states that the fraction of positive and negative links which achieve the optimal ranking cannot be greater than GAUC. It can be proved using the fact that the arithmetic mean is always greater than the geometric mean.

Note that the calculation of the GAUC lower bound in Theorem 1 still requires quadratic time, which may make the underlying optimization problem intractable for large-scale networks. An equivalent, more tractable, form can be derived.

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

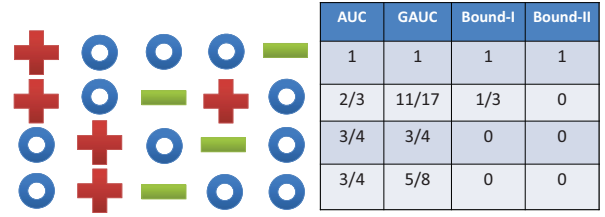
$$\frac{1}{|\mathcal{P}| + |\mathcal{N}|} \left( \sum_{(i,j) \in \mathcal{P}} I(\hat{X}_{ij} > \max_{(i,s) \in \mathcal{O} \cup \mathcal{N}} (\hat{X}_{is})) + \sum_{(i,j) \in \mathcal{N}} I(\hat{X}_{ij} < \min_{(i,t) \in \mathcal{O} \cup \mathcal{P}} (\hat{X}_{it})) \right), \quad (7)$$

which can be calculated in linear time.

Proposition 1 suggests that instead of exhaustively searching for the pairwise comparisons which are jointly satisfied, we only need to compare with the maximum or minimum ranking score over the set  $\mathcal{O} \cup \mathcal{N}$  or  $\mathcal{O} \cup \mathcal{P}$ , respectively. Therefore, the complexity of computing (7) mainly depends on searching for the maximum and the minimum over  $\mathcal{O} \cup \mathcal{N}$  and  $\mathcal{O} \cup \mathcal{P}$ , respectively, and the pairwise comparisons between  $\mathcal{P}$  and  $\mathcal{O} \cup \mathcal{N}$  as well as between  $\mathcal{N}$  and  $\mathcal{O} \cup \mathcal{P}$ .

### 3.3 Lower Bound-II

In order to quantify whether all the positive links are ranked on the top and all the negative links are ranked at the bottom of a ranking list, a stricter lower bound for GAUC can be derived.



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

Figure 1: Comparisons of AUC, GAUC, lower bound-I, and lower bound-II over four different ranking lists, listed from top (left) to bottom (right). Note that GAUC is sensitive to the positions of the positive and negative links while AUC is only sensitive to the positions of positive links here; Lower Bound-I aims to quantify the fraction of positive and negative links which are ranked on the top or bottom of a ranking list; Lower Bound-II indicates whether all the positive links or negative links in the ranking list are all ranked on the top or at the bottom of a ranking list. “+” denotes a positive link, “o” denotes an unknown status link, and “-” denotes a negative link.

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

$$GAUC(i) \geq \frac{|\mathcal{P}|}{|\mathcal{P}| + |\mathcal{N}|} \prod_{(i,j) \in \mathcal{P}} I(\hat{X}_{ij} > \max_{(i,s) \in \mathcal{O} \cup \mathcal{N}} (\hat{X}_{is})) + \frac{|\mathcal{N}|}{|\mathcal{P}| + |\mathcal{N}|} \prod_{(i,j) \in \mathcal{N}} I(\hat{X}_{ij} < \min_{(i,t) \in \mathcal{O} \cup \mathcal{P}} (\hat{X}_{it})), \quad (8)$$

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

Although the calculation of this lower bound takes only linear time, an equivalent form can be derived to further simplify it.

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

$$\frac{|\mathcal{P}|}{|\mathcal{P}| + |\mathcal{N}|} I\left(\min_{(i,j) \in \mathcal{P}} (\hat{X}_{ij}) > \max_{(i,s) \in \mathcal{O} \cup \mathcal{N}} (\hat{X}_{is})\right) + \frac{|\mathcal{N}|}{|\mathcal{P}| + |\mathcal{N}|} I\left(\max_{(i,j) \in \mathcal{N}} (\hat{X}_{ij}) < \min_{(i,t) \in \mathcal{O} \cup \mathcal{P}} (\hat{X}_{it})\right), \quad (9)$$

which can be calculated in linear time.

The complexity of computing (9) depends on searching for the maxima and minima over the four different sets.

Although GAUC is a good measure in signed networks since it can quantify both the head and tail of a ranking list which includes positive, negative, and unknown status links, we note that directly optimizing GAUC may not achieve satisfactory link recommendation performance (especially for top- $k$  link recommendation performance) because it is computationally inefficient and does not explicitly put emphasis on ranking positive links on the top, and negative links at the bottom, of a ranking list. Therefore, in this paper we develop two efficient and effective latent link recommendation algorithms by optimizing the two lower bounds (7) and (9), respectively.



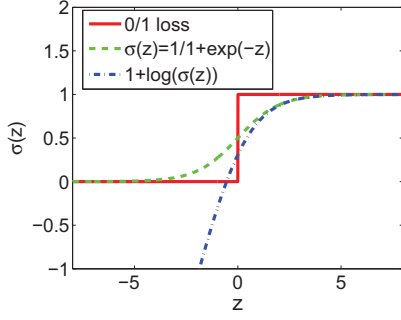


Figure 2: The sigmoid function  $\sigma(z) = \frac{1}{1+\exp(-z)}$  is a smooth approximation of 0/1 loss.

## 4. EFFICIENT LATENT LINK RECOMMENDATION ALGORITHMS

In this section, we first state the problem we aim to study. Then we develop two probabilistic models: efficient latent link recommendation-I (ELLR-I) and efficient latent link recommendation-II (ELLR-II), based upon the two lower bounds for GAUC given by (7) and (9), respectively, to do personalized latent link recommendation in signed networks. Finally, we describe the optimization procedure for these two algorithms.

### 4.1 Problem Statement

Let  $X \in \{1, -1, ?\}^{n \times m}$  be a partially observed signed network in which  $\pm 1$  denote an observed positive/negative link and  $?$  denotes an unknown status link which could be either. In the training stage, we study the underlying mechanism for ranking observed positive links on the top, observed negative links at the bottom, and unknown status links in between. In the test phase, we evaluate how these *latent* (i.e., unknown status) links are ranked based upon the positions of potential positive and negative links.

In particular, we aim to learn a function  $f(X, i, j) = \hat{X}_{ij}$  which can produce a ranking score ( $\hat{X}_{ij}$ ) for each entry of  $X$  in the training stage. In many real world applications  $X$  will be a sparse matrix which has low rank structure. Therefore,  $X$  can be approximated with two low rank matrices  $U \in \mathbb{R}^{r \times n}$  and  $V \in \mathbb{R}^{r \times m}$  and our aim can be recast as learning the following ranking function:

$$f(U, V, i, j) = f_{ij}(U, V) = U_i^T V_j, \quad (10)$$

such that the ranking lists can be optimized in the sense of lower bounds for GAUC. Note that  $U_i$  and  $V_j$  denote two latent user feature vectors, respectively.  $r \ll \min(m, n)$  is the rank. When  $X$  is a symmetric signed network, i.e.,  $X = X^T$ , we can set  $U = V$  for simplicity.

### 4.2 Efficient Latent Link Recommendation-I

We propose a probabilistic model to perform efficient latent link recommendation in signed networks based upon lower bound-I for GAUC. 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, as well as between negative links and the latent or positive link which has the smallest ranking score. Specifically, ELLR-I

maximizes the following posterior distribution,

$$P(U, V | >_f, X) \propto P(>_f, X | U, V) P(U) P(V), \quad (11)$$

where  $>_f$  denotes the orderings on  $\{f_{ij} \mid 1 \leq j \leq n\}$  determined by  $X$ , using the conditions that the positive links have higher scores than the other links and the negative links have lower scores than the other links. 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 (11) becomes:

$$\begin{aligned} & P(>_f, X | U, V) P(U) P(V) \\ &= \Pi_i \Pi_{(i,j) \in \mathcal{P}} \Pi_{(i,s) \in C_{i,j,s}} P(>_f, X_{ij}, X_{is} | U_i, V_j, V_s) P(U_i) P(V_j) \\ &= \Pi_i \Pi_{(i,j) \in \mathcal{P}} \left( \Pi_{(i,s) \in \mathcal{O} \cup \mathcal{N}} P(>_f, X_{ij} = 1, X_{is} \neq 1 | U_i, V_j, V_s) \right) \\ & \quad \Pi_{(i,s) \in \mathcal{N}} \left( \Pi_{(i,s) \in \mathcal{O} \cup \mathcal{P}} P(>_f, X_{ij} = -1, X_{is} \neq -1 | U_i, V_j, V_s) \right) \\ & \quad P(U_i) P(V_j) \end{aligned} \quad (12)$$

where  $C_{i,j,s} = \{(i, s) \mid X_{ij} \in \mathcal{P} \text{ and } X_{is} \in \mathcal{O} \cup \mathcal{N}\} \cup \{(i, s) \mid X_{ij} \in \mathcal{N} \text{ and } X_{is} \in \mathcal{O} \cup \mathcal{P}\}$ .  $\mathcal{P}$  and  $\mathcal{N}$  are the set of positive and that of negative links for user  $i$ .

To achieve the objective of ELLR-I,  $\Pi_{(i,s)} P(>_f, X_{ij} = 1, X_{is} \neq 1 | U_i, V_j, V_s)$  should contribute to the first term of (7) via the exponent  $I(U_i^T V_j > \max_{(i,s) \in \mathcal{O} \cup \mathcal{N}} U_i^T V_s)$ , i.e., this probability should be close to 1 when the difference between  $U_i^T V_j$  and the maximum value of  $U_i^T V_s$  is large and should be close to 0 when this difference is small. Similarly,  $\Pi_{(i,s)} P(>_f, X_{ij} = -1, X_{is} \neq -1 | U_i, V_j, V_s)$  should contribute to the second term of (7) via its exponent  $I(U_i^T V_j < \min_{(i,s) \in \mathcal{O} \cup \mathcal{P}} U_i^T V_s)$ , i.e., this probability should be close to 1 when the difference between  $U_i^T V_j$  and minimum value of  $U_i^T V_s$  is small and should be close to 0 when this difference is large. Previous work such as that of Rendle et al. [20] has shown that the sigmoid function  $\sigma(z) = 1/(1 + \exp(-z))$  is an ideal smooth version of 0/1 loss (as shown in Figure 2); thus  $\Pi_{(i,s) \in \mathcal{O} \cup \mathcal{N}} P(>_f, X_{ij} = 1, X_{is} \neq 1 | U_i, V_j, V_s)$  can be defined as:

$$\begin{aligned} & \Pi_{(i,s) \in \mathcal{O} \cup \mathcal{N}} P(>_f, X_{ij} = 1, X_{is} \neq 1 | U_i, V_j, V_s) \\ &= \sigma(U_i^T V_j - \max_{(i,s) \in \mathcal{O} \cup \mathcal{N}} U_i^T V_s), \end{aligned} \quad (13)$$

and  $\Pi_{(i,s) \in \mathcal{O} \cup \mathcal{P}} P(>_f, X_{ij} = -1, X_{is} \neq -1 | U_i, V_j, V_s)$  should be written as:

$$\begin{aligned} & \Pi_{(i,s) \in \mathcal{O} \cup \mathcal{P}} P(>_f, X_{ij} = -1, X_{is} \neq -1 | U_i, V_j, V_s) \\ &= \sigma(-U_i^T V_j + \min_{(i,s) \in \mathcal{O} \cup \mathcal{P}} U_i^T V_s), \end{aligned} \quad (14)$$

where the  $\max(\cdot)$  and  $\min(\cdot)$  are evaluated over  $s$ .

We also put two zero-mean spherical Gaussian priors over user  $i$ 's and user  $j$ 's feature vectors, respectively, i.e.,

$$P(U_i) = \mathcal{N}(U_i | 0, \sigma_U^2), \quad (15)$$

and

$$P(V_j) = \mathcal{N}(V_j | 0, \sigma_V^2), \quad (16)$$

where  $\sigma_U^2$  and  $\sigma_V^2$  are variances for  $U$  and  $V$ , respectively.

Therefore, the log-likelihood of ELLR-I is given by:

$$\begin{aligned}
L_{\text{ELLR-I}}(U, V) &= \log P(U, V | >_f, X) \\
&= \sum_{i=1}^n \sum_{(i,j) \in \mathcal{P}} \log \left( \sigma(U_i^T V_j - \max_{(i,s) \in \mathcal{O} \cup \mathcal{N}} U_i^T V_s) \right) \\
&\quad + \sum_{i=1}^n \sum_{(i,j) \in \mathcal{N}} \log \left( \sigma(-U_i^T V_j + \min_{(i,s) \in \mathcal{O} \cup \mathcal{P}} U_i^T V_s) \right) \\
&\quad - \frac{\lambda_U}{2} \sum_{i=1}^n U_i^T U_i - \frac{\lambda_V}{2} \sum_{j=1}^n V_j^T V_j + c,
\end{aligned} \tag{17}$$

where  $c$  is a constant which is independent of  $U$  and  $V$ , and  $\lambda_U = 1/\sigma_U^2$ ,  $\lambda_V = 1/\sigma_V^2$  are parameters for controlling the trade-off between the first two terms and the two regularization terms.

In practical applications, simply considering the maximum ranking score in  $\mathcal{O} \cup \mathcal{N}$  (or the minimum ranking score in  $\mathcal{O} \cup \mathcal{P}$ ) may make the gradient for (17) unstable when the most extreme example is an outlier. To address this issue, instead of using the maximum ranking score (or the minimum ranking score), we can utilize the mean of a number of (*e.g.*,  $p = 30$  or  $50$ ) largest ranking scores in  $\mathcal{O} \cup \mathcal{N}$  (or the mean of a number of smallest ranking scores in  $\mathcal{O} \cup \mathcal{P}$ ). In this way, the gradient of (17) becomes more reliable. A local maximum of the objective function given by (17) can be found by performing gradient ascent in  $U$  and  $V$  iteratively. The details are omitted here due to space limitations.

### 4.3 Efficient Latent Link Recommendation-II

We propose another probabilistic model to perform efficient latent link recommendation in signed networks based upon lower bound-II for GAUC. Like ELLR-I, 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, and the negative link with the largest score should be smaller than the latent or positive link which has the smallest ranking score. Using Bayes' rule, the assumption that each user is acting independently, and each pair of users' (or user and item's) ranking scores is compared independently, we can derive (11) and (12) as we did for ELLR-I.

To achieve the objective of ELLR-II, the probability product  $\prod_{(i,j) \in \mathcal{P}} \prod_{(i,s)} P(>_f, X_{ij} = 1, X_{is} \neq 1 | U_i, V_j, V_s)$  should contribute to the first term of (9) through the exponent  $I(\min_{(i,j) \in \mathcal{P}} U_i^T V_j > \max_{(i,s) \in \mathcal{O} \cup \mathcal{N}} U_i^T V_s)$ , *i.e.*, this probability should be close to 1 when the difference between the minimum value of  $U_i^T V_j$  and the maximum value of  $U_i^T V_s$  is large and should be close to 0 when this difference is small. Also,  $\prod_{(i,j) \in \mathcal{N}} \prod_{(i,s)} P(>_f, X_{ij} = -1, X_{is} \neq -1 | U_i, V_j, V_s)$  should contribute to the second term of (9) via its exponent  $I(\max_{(i,j) \in \mathcal{N}} U_i^T V_j < \min_{(i,s) \in \mathcal{O} \cup \mathcal{P}} U_i^T V_s)$ , *i.e.*, this probability should be close to 1 when the difference between the maximum value of  $U_i^T V_j$  and minimum value of  $U_i^T V_s$  is small and should be close to 0 when this difference is large. Therefore,  $\prod_{(i,j) \in \mathcal{P}} \prod_{(i,s) \in \mathcal{O} \cup \mathcal{N}} P(>_f, X_{ij} = 1, X_{is} \neq 1 | U_i, V_j, V_s)$  can be defined as:

$$\sigma \left( \min_{(i,j) \in \mathcal{P}} U_i^T V_j - \max_{(i,s) \in \mathcal{O} \cup \mathcal{N}} U_i^T V_s \right), \tag{18}$$

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%

and  $\prod_{(i,j) \in \mathcal{N}} \prod_{(i,s) \in \mathcal{O} \cup \mathcal{P}} P(>_f, X_{ij} = -1, X_{is} \neq -1 | U_i, V_j, V_s)$  should be defined as:

$$\sigma \left( -\max_{(i,j) \in \mathcal{N}} U_i^T V_j + \min_{(i,s) \in \mathcal{O} \cup \mathcal{P}} U_i^T V_s \right), \tag{19}$$

In the objective of ELLR-II,  $P(U_i)$  and  $P(V_j)$  are defined as in (15) and (16), respectively. Therefore, the log-likelihood of ELLR-II is given by:

$$\begin{aligned}
L_{\text{ELLR-II}}(U, V) &= \log P(U, V | >_f, X) \\
&= \sum_{i=1}^n \log \left( \sigma \left( \min_{(i,j) \in \mathcal{P}} U_i^T V_j - \max_{(i,s) \in \mathcal{O} \cup \mathcal{N}} U_i^T V_s \right) \right) \\
&\quad + \sum_{i=1}^n \log \left( \sigma \left( -\max_{(i,j) \in \mathcal{N}} U_i^T V_j + \min_{(i,s) \in \mathcal{O} \cup \mathcal{P}} U_i^T V_s \right) \right) \\
&\quad - \frac{\lambda_U}{2} \sum_{i=1}^n U_i^T U_i - \frac{\lambda_V}{2} \sum_{j=1}^n V_j^T V_j + c,
\end{aligned} \tag{20}$$

where  $c$  is a constant which is independent of  $U$  and  $V$ , and as before  $\lambda_U = 1/\sigma_U^2$  and  $\lambda_V = 1/\sigma_V^2$ .

As with ELLR-I, in real world applications, simply considering the maximum ranking score in  $\mathcal{N}$  and  $\mathcal{O} \cup \mathcal{N}$  as well as the minimum ranking score in  $\mathcal{P}$  and  $\mathcal{O} \cup \mathcal{P}$  may make the gradient of (20) unstable when the most extreme example is an outlier. To resolve this problem, we replace the maximum ranking score in  $\mathcal{N}$  (or the minimum ranking score in  $\mathcal{P}$ ) with the mean of  $q$  (*e.g.*,  $q = 5$  or  $10$ ) largest ranking scores in  $\mathcal{N}$  (or with the mean of  $q$  smallest ranking scores in  $\mathcal{P}$ ). We also replace the maximum ranking score in  $\mathcal{O} \cup \mathcal{N}$  (or the minimum ranking score in  $\mathcal{O} \cup \mathcal{P}$ ) with the mean of  $p$  (*e.g.*,  $p = 30$  or  $50$ ) largest ranking scores in  $\mathcal{O} \cup \mathcal{N}$  (or with the mean of  $p$  the smallest ranking scores in  $\mathcal{O} \cup \mathcal{P}$ ) as we did in ELLR-I. As with the objective of ELLR-I in (17), a local maximum of the objective function given by (20) can be obtained by performing gradient ascent in  $U$  and  $V$  iteratively. The details are omitted here due to space limitations.

### 4.4 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.

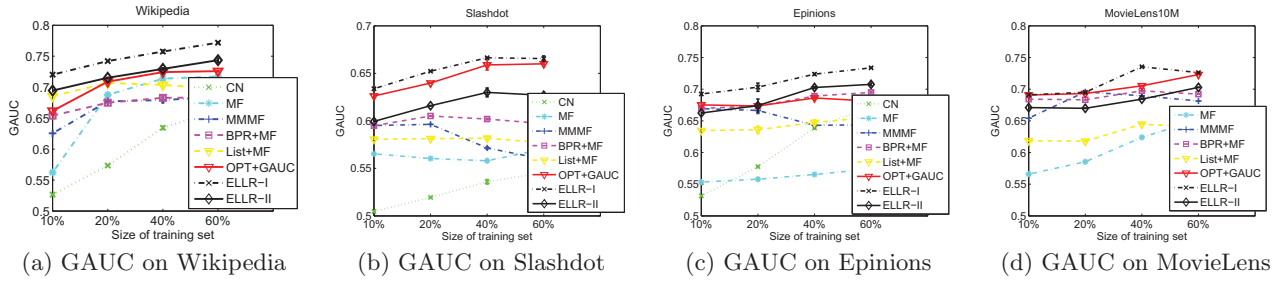


Figure 3: GAUC on Wikipedia, Slashdot, Epinions, and MovieLens10M. Error bars denote standard deviations.

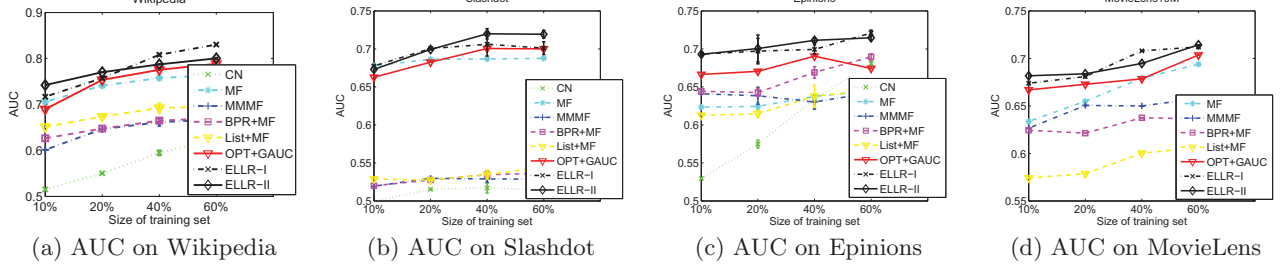


Figure 4: AUC on Wikipedia, Slashdot, Epinions, and MovieLens10M. Error bars denote standard deviations.

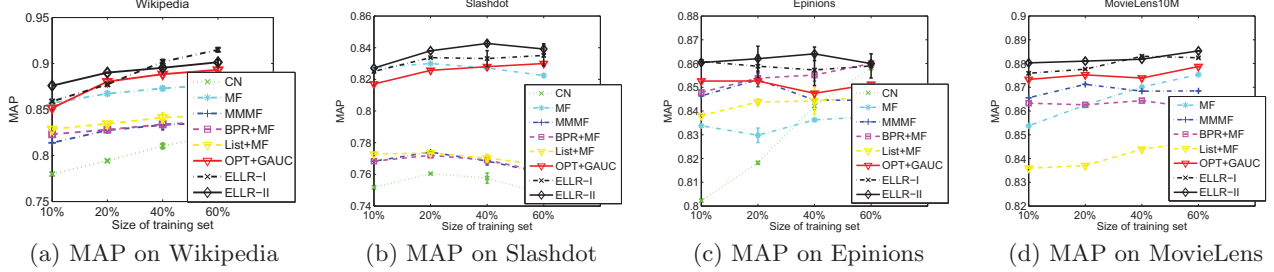


Figure 5: MAP on Wikipedia, Slashdot, Epinions, and MovieLens10M. Error bars denote standard deviations.

## 5. EXPERIMENT

To demonstrate the effectiveness and efficiency of ELLR-I and ELLR-II, we compare them against various baseline methods based upon four publicly available datasets.

We consider three well-known signed directed social networks, *i.e.*, Wikipedia [4], Slashdot [13] and Epinions [7]<sup>1</sup>. The Wikipedia data comprise a voting network for promoting candidates to the role of admin. Each voter can indicate a positive (for supporting) or negative (for opposing) vote with respect to the promotion of a candidate [15]. Slashdot is a social website focusing on technology related news. In Slashdot Zoo, users can tag each other as friends (like) or foes (dislike) based upon comments on articles. Epinions, which is a product review website, is a trust network in which users can indicate whether they trust or distrust each other based upon their reviews.

We also consider a collaborative filtering dataset, *i.e.*, MovieLens10M<sup>2</sup> which includes 71,567 users, 10,681 items, and more than 10 million ratings ranging from 1 to 5. Although this dataset is originally used for movie rating prediction, we preprocess it such that rating values larger than

3 are treated as positive links and rating values smaller than 3 are treated as negative links.

The detailed statistics of these four datasets are provided in Table 1.

### 5.1 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  $X_{\text{Train}}$ ) for training, and evaluate over a test set (*i.e.*,  $X_{\text{Test}}$ ) comprising the remaining non-zero entries. The zero entries in  $X_{\text{Train}}$  are called *latent* links since each link has the potential to either be a positive or a negative link.

In order to evaluate the effectiveness of ELLR-I and ELLR-II for link recommendation, we utilize GAUC (defined over  $\pm 1$  and 0) in (5), AUC (defined over  $\pm 1$ ) in (4), and mean average precision (MAP) (defined over  $\pm 1$ ) to quantify the ranking performance over  $X_{\text{Test}}$ .

To evaluate the effectiveness of these two proposed approaches for top- $k$  link recommendation, we also report the associated precision at top  $k$  positions (*i.e.*, Precision@ $k$  or P@ $k$ ) and recall at top  $k$  positions (*i.e.*, Recall@ $k$ ). In particular, Precision@ $k$  is defined as:

<sup>1</sup>These datasets are available online at <http://snap.stanford.edu/data/>.

<sup>2</sup>This dataset is available online at <http://grouplens.org/datasets/movielens/>.

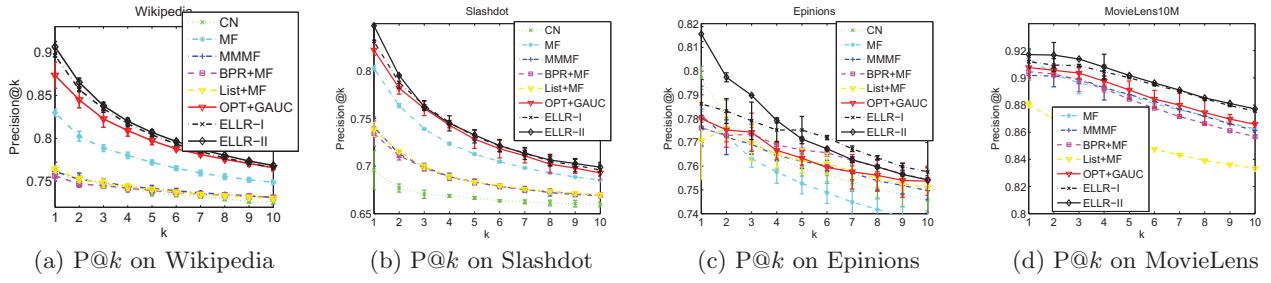


Figure 6: Precision@ $k$  on Wikipedia, Slashdot, Epinions, and MovieLens10M. Error bars represent standard deviations.

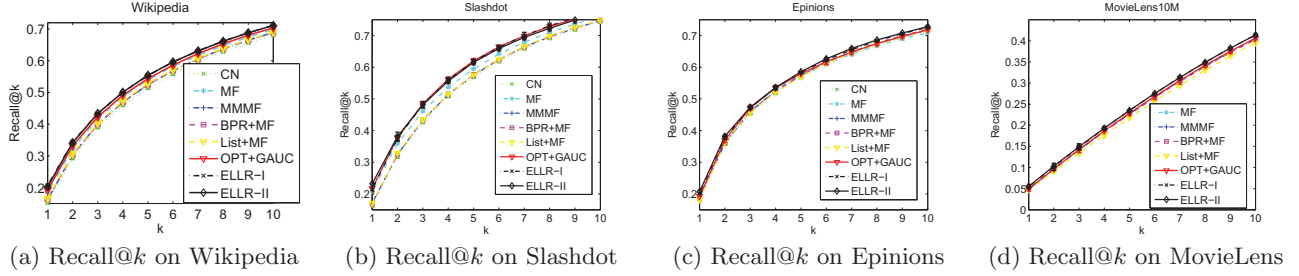


Figure 7: Recall@ $k$  for Wikipedia, Slashdot, Epinions, and MovieLens10M. Error bars represent standard deviations.

$$\text{Precision@}k = \frac{\# \text{ of positive links in the top } k}{\# \text{ of positive and negative links in top } k},$$

and Recall@ $k$  is given as:

$$\text{Recall@}k = \frac{\# \text{ of positive links in the top } k}{\# \text{ of positive links}}.$$

## 5.2 Parameter Settings

There are three hyper-parameters in ELLR-I and ELLR-II, *i.e.*,  $r$ ,  $\lambda_U$ , and  $\lambda_V$ . For simplicity, we set  $\lambda_U = \lambda_V$ , which is optimized by searching over the grid  $\{1, 5, 10, 50, 100, 200\}$ .  $r$  is optimized by searching over the grid  $\{5, 10, 30, 50, 70, 90\}$ . To determine the optimal parameters for test, we conduct 5-fold cross-validation over the training set  $X_{\text{Train}}$  and the parameter combination which achieves the best average GAUC is utilized for test.

## 5.3 Results-I: Effectiveness

**Baselines:** we evaluate the effectiveness of ELLR-I and ELLR-II by comparing them with six different baseline algorithms. Among them, common neighbor (CN) [17] is a representative network topology-based method; matrix factorization (MF) [12] is a pointwise approach; maximum margin matrix factorization (MMMF) [30] and Bayesian personalized ranking based matrix factorization (BPR+MF) [20] are pairwise methods; list-wise learning to rank with matrix factorization (List+MF) [24] is a listwise algorithm; and OPT+GAUC [26], which directly optimizes GAUC, is also a pairwise approach. The hyper-parameters of these baseline approaches (except CN) are selected as they are for ELLR-I and ELLR-II. To ensure that our results are reliable, we conduct all experiments 5 times; the average GAUC/AUC/MAP and their associated standard deviations are reported.

**Link Recommendation:** Figures 3, 4 and 5 show the GAUC/AUC/MAP and their associated standard deviations over four datasets when the size of training set varies from

10% to 60%. We observe that CN is generally outperformed by other approaches since it only considers the neighborhood structure of these networks. Note that CN is not reported on MovieLens10M because that is a bipartite network. For GAUC in Figure 3, pairwise approaches, *i.e.*, MMMF and BPR+MF outperform MF which is a pointwise approach. This may be because MF only reconstructs the observed entries in the adjacency matrix and neglects unknown status links. For AUC and MAP in Figure 4 and 5, respectively, MF outperforms MMMF and BPR+MF most of the time. This may be because they do not model negative links explicitly. OPT+GAUC generally outperforms the other five baseline algorithms for GAUC/AUC/MAP since it models the positive, negative, and unknown status links in signed networks more completely. ELLR-I and ELLR-II can further improve the performance of OPT+GAUC for AUC/MAP because they explicitly put emphasis on ranking positive links on the top and negative links at the bottom of a ranking list (as shown in Proposition 1 and Proposition 2). In general, the GAUC of ELLR-II is worse than that of ELLR-I. This is because the lower bound in Proposition 2 (the objective of ELLR-II) is more loose than that in Proposition 1 (the objective of ELLR-I) for GAUC.

**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 6 and 7, we observe that ELLR-I and ELLR-II consistently outperform the baseline approaches. This indicates that emphasizing putting positive links on the top and negative links at the bottom of a ranking list helps to enhance top- $k$  link recommendation performance. Note that ELLR-II generally outperforms ELLR-I in Figure 6 and 7; this is because ELLR-II is optimizing a stricter bound as shown in Proposition 1 and Proposition 2. Note that we also observe similar results over other size of training sets; these results are omitted here due to space limitations.



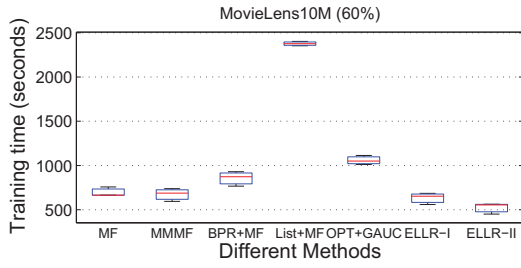


Figure 8: Comparison of training time (seconds) for different approaches over MovieLens10M (60 %).

## 5.4 Results-II: Efficiency

We check the efficiency of two proposed approaches, *i.e.*, ELLR-I and ELLR-II, by comparing their training time with baseline algorithms over MovieLens10M (60%) as shown in Figure 8. For fair comparison, we use the same hyper-parameters for all the approaches.

We observed that ELLR-I and ELLR-II require less training time than other approaches. This is because: for MF, the computational complexity of its gradient is linearly proportional to the number of observed entries in signed networks; for MMMF and BPR+MF, the computational complexities of their gradients depend on the number of pairwise comparisons of positive links to non-positive links (*i.e.*, unknown+negative links). Although a sampling technique was used for non-positive links so as to facilitate the optimization, MMMF and BPR+MF still consume substantial time to calculate the gradient at each step. List+MF converges slower than other methods since it must perform listwise comparison for each positive link. OPT+GAUC converges slower than MMMF and BPR+MF since it involves an additional cost for pairwise comparisons of negative links to non-negative links at each step. ELLR-I, however, only considers the most extreme non-positive links as well as the most extreme non-negative links and thus converges faster than the baseline methods (Proposition 1). ELLR-II can further reduce the time cost of ELLR-I since it only further considers the most extreme positive and negative links at each step (shown in Proposition 2). We also observe similar results over the other three datasets. These results are omitted here due to space limitations.

## 5.5 Results-III: 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  $\lambda_U$  or  $r$ , we keep the other parameters fixed. We plot the GAUC/AUC/ MAP with respect to  $\lambda_U$  or  $r$  in Figure 9. We observe that both ELLR-I and ELLR-II are very stable and they achieve good performance when  $\lambda_U$  varies from 10 to 200 and  $r$  varies from 10 to 90.

## 6. CONCLUSION AND DISCUSSION

In this paper, we derived two lower bounds for GAUC which can be computed in linear time. The first quantifies the fraction of positive and negative links which are ranked at the optimal positions (*i.e.*, positive links on the top and negative links at the bottom); the second is even stricter and measures whether all the positive links are ranked on

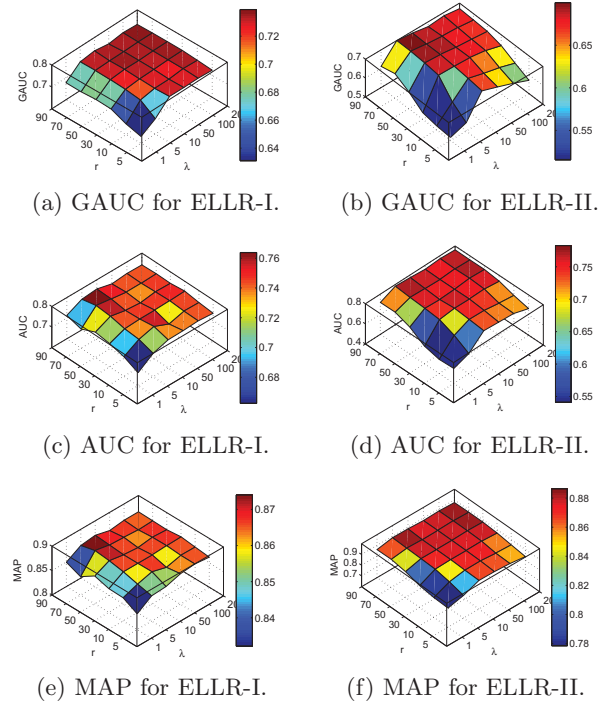


Figure 9: The parameter sensitivity of ELLR-I and ELLR-II with respect to the regularization parameters  $\lambda_U = \lambda_V \in \{1, 5, 10, 50, 100, 200\}$  and  $r \in \{5, 10, 30, 50, 70, 90\}$  over Wikipedia (20%). We can observe that both ELLR-I and ELLR-II are very stable with respect to  $\lambda_U$  ( $\lambda_V$ ) and  $r$ .

the top and whether all the negative links are ranked at the bottom of a ranking list. With these two lower bounds, we developed two efficient probabilistic models, *i.e.*, ELLR-I and ELLR-II, to infer personalized ranking lists of latent links by directly optimizing them. 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.

One limitation of the current approaches is that they only consider latent features of signed networks and do not incorporate explicit features. It would be interesting to combine these two types of features together to perform latent link recommendation in the future. Furthermore, it is also interesting to investigate side information of users/items and utilize this information to improve latent link recommendation in signed networks. Finally, inferring negative links [28] with our proposed algorithms is also an interesting problem to study.

## Acknowledgement

The work of Dongjin Song and David A. Meyer were supported by the U.S. Department of Defense Minerva Research Initiative/Army under Grant W911NF-09-1-0081 and in part by the National Science Foundation-Division of Mathematical Sciences under Grant 1223137. The work of Dacheng Tao was supported by the Australian Research Council under Project DP-140102164 and Project FT-130101457.

## 7. 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] M. J. Brozzowski, T. Hogg, and G. Szabo. Friends and foes: ideological social networking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 817–820, Florence, Italy, 2008.
- [4] M. Burke and R. Kraul. Mopping up: Modeling wikipedia promotion decisions. In *Proceedings of the 2008 ACM conference on Computer Supported Cooperative Work*, pages 27–36, 2008.
- [5] 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.
- [6] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of ACM Conference on Recommender Systems*, pages 39–46, 2010.
- [7] R. V. Guha, R. Kumar, P. Raghavan, and A. Tomkin. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web*, pages 403–412, New York, NY, 2004.
- [8] J. A. Hanley and B. J. Mcneil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 1982.
- [9] 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.
- [10] L. Katz. A new status index dervied from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.
- [11] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 426–434, 2008.
- [12] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.
- [13] C. Lampe, E. Johnston, and R. Resnick. Follow the reader: filtering comments on slashdot. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1253–1262, San Jose, CA, 2007.
- [14] 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.
- [15] J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 20th International Conference on World Wide Web*, pages 641–650, 2010.
- [16] Y. Li. A generalization of auc to an ordered multiclass diagnosis and application to longitudianl data analysis on intellectual outcome in pediatric brain-tumor patients. *Ph.D Thesis of Geogia State University*, 2009.
- [17] D. Liben-Nowell and J. Kleinberg. The link prediction for social networks. *Journal of American Society for Information Science and Technology*, 58(7):1019–1031, 2007.
- [18] 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.
- [19] C. T. Nakas and C. T. Yiannoutsos. Ordered multi-class roc analysis with continuous measurements. *Statistics in Medicine*, 23:3437–3449, 2004.
- [20] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 452–461, Arlington, Virginia, United States, 2009.
- [21] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative filtering. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 713–719, 2005.
- [22] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Proceedings of the Neural Information Processing Systems*, 2007.
- [23] G. Salton and M. J. McGill. *Introduction to modern information retrieval*. New York: McGraw Hill, 1983.
- [24] Y. Shi, M. Larson, and A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 269–272, 2010.
- [25] D. Song and D. A. Meyer. A model of consistent node types in signed directed social networks. In *Proceedings of the 6th IEEE/ACM International Conference on Advances in Social Network Analysis and Mining*, pages 82–90, Beijing, China,, 2014.
- [26] D. Song and D. A. Meyer. Recommnding positive links in signed social networks by optimizing a generalized auc. In *Proceedings of 29th AAAI Conference on Artificial Intelligence*, Austin, USA, January, 2015.
- [27] D. Song, D. A. Meyer, and M. R. Min. Fast nonnegative matrix factorization with rank-one admm. In *NIPS Workshop on Optimization for Machine Learning*, Montreal, Canada, 2014.
- [28] J. Tang, S. Chang, C. C. Aggarwal, and H. Liu. Negative link prediction in social media. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, pages 87–96, 2015.
- [29] J. Tang, S. Wu, J. Sun, and H. Su. Cross-domain collaboration recommendation. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1285–1293, Beijing, China, August, 2012.
- [30] M. Weimer, A. Karatzoglou, and A. Smola. Improving maximum margin matrix factorization. *Machine Learning*, 72(3):263–276, 2008.
- [31] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach learning to rank - theory and algorithm. In *Proceedings of International Conference on Machine Learning*, pages 1192–1199, 2008.