Negative Link Prediction in Social Media

Jiliang Tang[‡], Shiyu Chang[♯], Charu Aggarwal[†] and Huan Liu[‡]
[‡]Computer Science & Engineering, Arizona State University
[♯] Beckman Institute, University of Illinois at Urbana-Champaign
[†] IBM T.J. Watson Research Center
{Jiliang.Tang,Huan.Liu}@asu.edu[‡],{chang87}@illinois.edu[♯],{charu}@us.ibm.com[†]

ABSTRACT

Signed network analysis has attracted increasing attention in recent years. This is in part because research on signed network analysis suggests that negative links have added value in the analytical process. A major impediment in their effective use is that most social media sites do not enable users to specify them explicitly. In other words, a gap exists between the importance of negative links and their availability in real data sets. Therefore, it is natural to explore whether one can predict negative links automatically from the commonly available social network data. In this paper, we investigate the novel problem of negative link prediction with only positive links and content-centric interactions in social media. We make a number of important observations about negative links, and propose a principled framework NeLP, which can exploit positive links and content-centric interactions to predict negative links. Our experimental results on real-world social networks demonstrate that the proposed NeLP framework can accurately predict negative links with positive links and content-centric interactions. Our detailed experiments also illustrate the relative importance of various factors to the effectiveness of the proposed framework.

Categories and Subject Descriptors

H3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

General Terms

Algorithms; Design; Experimentation

Keywords

Negative Links; Negative Link Prediction; Signed Social Networks; Social Media

1. INTRODUCTION

Social networks have enabled a vast diversity of relations between users such as friendships in Facebook[1], follower relations in Twitter[2] and trust relations in Epinions[3]. The increasing availability of large-scale online social network data is useful not only for various tasks in social network analysis such as community detection [20] and link prediction [15], but it is also leveraged for various traditional data mining tasks such as feature selection [22] and recommendations [21]. The vast majority of existing research has overwhelmingly focused on social networks with only positive links. However, social networks can contain both positive and negative links. Examples of signed social networks include Epinions with trust and distrust links, and Slashdot[4] with friend and foe links. The recent availability of signed social networks in social media sites such as Epinions and Slashdot has motivated increasing research on signed network analysis [11, 13, 3].

It is evident from recent work that negative links have significant added value over positive links in various analytical tasks. For example, a small number of negative links can significantly improve positive link prediction [7, 13], and they can also improve the performance of recommender systems in social media [24, 18]. Similarly, trust and distrust relations in Epinions can help users find high-quality and reliable reviews [7]. Furthermore, the specification of negative links is interesting in its own right. On the other hand, it is generally not very desirable for online social networks to explicitly collect negative links [8, 12]. As a consequence, most social media sites such as Facebook and Twitter do not enable users to explicitly specify negative links. Therefore, it is natural to question whether one can predict negative links automatically from the available data in social networks. A key assumption is that while *explicit* data is often not available about negative links, the combination of content-centric and structural data in social networks may contain implicit information about negative linkages. While this problem is very challenging [3], the results of such an approach have the potential to improve the quality of the results of a vast array of applications.

To preserve the generality of our approach, it is important to use social network data which is pervasively available across social networks in various forms. First, an obvious source of useful data are the positive linkages which

---

[1]https://www.facebook.com/
[2]https://twitter.com/
[3]http://www.epinions.com/
[4]http://slashdot.org/

(a) Positive and Negative Link Prediction

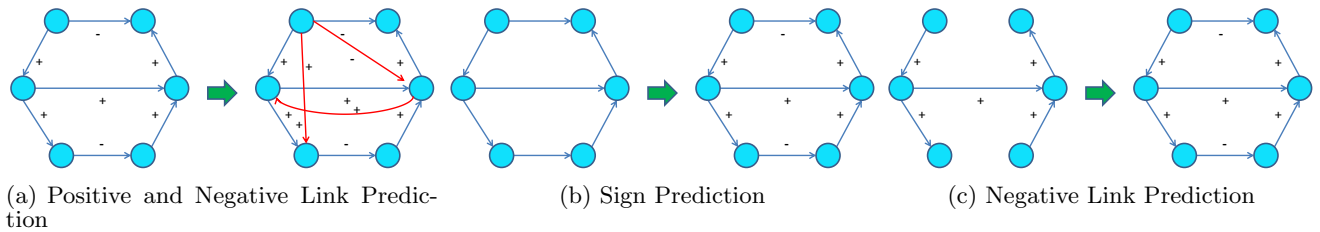(b) Sign Prediction

(c) Negative Link Prediction

Figure 1: An Illustration of the Differences of Positive and Negative Link Prediction, Sign Prediction and Negative Link Prediction.

are commonly available in most social networks. Second, in most social media sites, users can create or post content[5] and other users can comment, like/dislike and rate such content. In fact, such user interactions form the dominant social media activities[6], today. For example, users can comment, like, or dislike content such posts and videos. Another example is the case of Epinions, where users can rate the helpfulness of reviews written by others. In this paper, we study the novel problem of negative link prediction from these two pervasive sources in social media. However, we do not assume that examples of negative links are available. To achieve this goal, we need (a) theories explaining the relevance of positive links and content-centric user interactions for negative link prediction, and (b) combining these theories with mathematical models to predict negative links. These goals are achieved by our novel framework for the negative link prediction problem (NeLP). Our main contributions are summarized below,

- We provide a principled way to exploit positive links and content-centric user interactions for the problem of negative link prediction;

- We propose a novel framework NeLP to predict negative links by incorporating positive links and content-centric user interactions mathematically; and

- We evaluate the proposed NeLP framework in real-world social media datasets to understand the effectiveness and mechanisms of the proposed framework.

The rest of paper is organized as follows. In Section 2, we formally define the negative link prediction problem. We perform preliminary analysis on these datasets in Section 3 to study the interaction between existing social theories and negative link presence. In Section 4, we combine these theories with a mathematical formulation for negative link prediction. This is referred to as the NeLP framework. Section 5 presents experimental results with discussions. Section 6 briefly reviews related work. Finally, Section 7 concludes with future work.

2. PROBLEM STATEMENT

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ be the set of m users in the social network. A signed network can be decomposed into a positive network component $\mathcal{G}_p(\mathcal{U}, E_p)$ and a negative network component $\mathcal{G}_n(\mathcal{U}, E_n)$ where E_p and E_n are the sets of positive and negative links, respectively. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_M\}$

---

[5] The term "content" may be manifested in diverse ways such as statuses, tweets, images, or videos.

[6] http://www.marketingprofs.com/charts/2010/4101/social-media-brand-followers-hunting-for-deals

be the set of M pieces of content such as posts. We use $\mathbf{A} \in \mathbb{R}^{m \times M}$ to denote the user-content relationships where $\mathbf{A}_{ij} = 1$ if p_j is created by u_i, and $\mathbf{A}_{ij} = 0$ otherwise. Users can express opinions on content via comments, likes/dislikes, and ratings. Some social media sites provide explicit ways of enabling user feedback on content. Examples include likes/dislikes in Youtube, and "very helpful"/"not helpful" ratings in Epinions. Other more common forms of feedback in large-scale social networks such as Facebook and Twitter allow users to express their opinions in the form of textual comments and replies. In such cases, we adapt off-the-shelf opinion mining tools to extract user opinions from such texts. We use $\mathbf{O} \in \mathbb{R}^{m \times M}$ to represent the user-post opinion relations where $\mathbf{O}_{ij} = 1$, $\mathbf{O}_{ij} = -1$ and $\mathbf{O}_{ij} = 0$, if u_i expresses positive, negative and neutral (or no) opinions, respectively, on p_j.

With the aforementioned notations and definitions, the problem of negative link prediction in social media is formally defined as follows:

Given the positive network $\mathcal{G}_p$, and content-centric user interactions $\mathbf{A}$ and $\mathbf{O}$, we aim to develop a predictor f to predict the negative network $\mathcal{G}_n$ with $\mathcal{G}_p$, $\mathbf{A}$ and $\mathbf{O}$ as,

$$f : \{\mathcal{G}_p, \mathbf{A}, \mathbf{O}\} \to \mathcal{G}_n \tag{1}$$

The negative link prediction problem in this paper is quite different and much more challenging than the existing frameworks for positive/negative link prediction [13], and the sign prediction problem [26]. An illustration of their differences for the existing variations of the problem is demonstrated in Figure 1 as follows:

- One of the existing variations predicts positive and negative links from existing positive and negative links. On the other hand, as illustrated in Figure 1(c), we do not assume the existence of negative links.

- The second variation (Figure 1(b)) predicts signs of *already existing* links. On the other hand, the negative link prediction problem needs to identify the pairs of nodes between which negative links *are predicted to exist*.

3. DATA ANALYSIS

Because positive link prediction is dependent on "typical" behavior of social networks such as triadic closure, it is natural to explore similar properties of negative links with respect to other positive links, and content-centric interactions. Such an understanding lays the groundwork for a meaningful negative link-prediction model. For the purpose of this study, we collected two datasets from Epinions and Slashdot, that explicitly allow users to express both positive

Table 1: Statistics of the Epinions and Slashdot Datasets.

	Epinions	Slashdot
# of Users	14,765	7,275
# of Positive Links	272,513	67,705
# of Negative Links	52,704	20,851
# of Posts	612,321	300,932
# of Positive Opinions	6,937,986	1,742,763
# of Negative Opinions	163,502	42,260

and negative links. Note that the negative links in these two datasets only serve as a ground-truth about typical properties and the underlying social theories. However, they are not explicitly used in the proposed framework for the problem of negative link prediction.

Epinions is a popular product review site. Users can create both positive (trust) and negative (distrust) links to other users. They can write reviews for various products and other users can express opinions on these reviews with the use of "helpfulness" ratings from 1 to 6. In this work, we view ratings larger than 3 as positive, and those lower than 3 as negative. This assumption is used to populate the user-content opinion matrix **O** of the Epinions dataset.

Slashdot is a technology news platform in which users can create friend (positive) and foe (negative) links to other users. They can also post news articles. Other users may annotate these articles with their comments and opinions. In this case, we computed the sentiment polarities of comments based on an off-the-shelf manually labeled sentiment lexicon, i.e., MPQA Subjectivity Lexicon [7]. These sentiment polarities are used to populate the user-content opinion matrix.

Some additional preprocessing was performed on these two datasets by filtering users without any positive and negative links. A number of key statistics of these datasets are illustrated in Table 1[8]. It is evident from these statistics that users are more likely to express positive opinions than negative opinions in social media.

3.1 Where Are our "Enemies"?

Our first analytical task is to examine the typical structural relationships of "enemies" within the positive network. In other words, if u_i has a negative link to u_j in the negative network $\mathcal{G}_n$, we investigate the typical position of u_j with respect to u_i in the positive network $\mathcal{G}_p$. In the following sections, we will use $u_i + u_j$, $u_i - u_j$ and $u_i ? u_j$ to denote positive, negative and missing links between u_i to u_j, respectively.

For each negative link $u_i - u_j$ in $\mathcal{G}_n$, we use breadth-first search to compute the shortest path from u_i to u_j in $\mathcal{G}_p$. If paths exist from u_i to u_j, we report the length of the shortest path. Otherwise we report the length as "inf" to indicate there is no path from u_i to u_j in $\mathcal{G}_p$. The ratio distributions of the lengths of the shortest paths for all negative links are demonstrated in Figures 2(a) and 2(b) for Epinions and Slashdot, respectively. In both datasets, more than 45% of negative links $u_i - u_j$ have shortest path lengths less than 3, and more than 80% of them have shortest path lengths less than 4. These results suggest that our "enemies" are often close to us in the positive network $\mathcal{G}_p$. For example, about 82.64% and 87.86% of enemy-pairs are within 3-hops of each

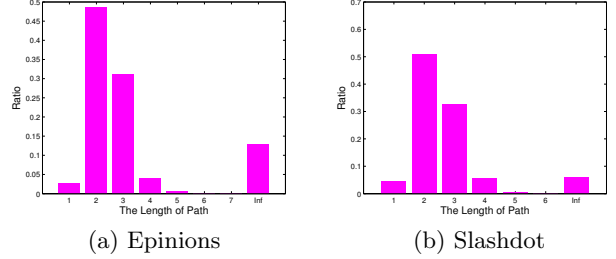(a) Epinions   (b) Slashdot

Figure 2: Ratio Distributions of the Length of Shortest Path for Pairs with Negative Links in the Positive Networks.

other in the positive networks of Epinions and Slashdot, respectively.

3.2 Social Theories in Signed Networks

In this subsection, we investigate two of the most important social theories for signed networks, i.e., balance theory [9, 2] and status theory [14].

In general, balance theory is based on the intuition that "the friend of my friend is my friend" and "the enemy of my enemy is my friend" [9]. This theory relates the balance of signs on a triad involving three users in a social network with positive and negative links. We use s_{ij} to denote the sign of the link between u_i and u_j where $s_{ij} = 1$ (or $s_{ij} = -1$) if there is a positive link (or a negative link) between u_i and u_j. Balance theory suggests that a triad $\langle u_i, u_j, u_k \rangle$ is balanced if the following is true - (a) $s_{ij} = 1$ and $s_{jk} = 1$, then $s_{ik} = 1$; or (b) $s_{ij} = -1$ and $s_{jk} = -1$, then $s_{ik} = 1$.

For a triad $\langle u_i, u_j, u_k \rangle$, there are four possible signed combinations (+,+,+), (+,+,-), (+,-,-) and (-,-,-), while only (+,+,+) and (+,-,-) are balanced. Based on the discussion in [13], the directions of links are ignored in the study of balance theory because balance theory is designed for undirected networks. We computed the relative presence of these four possible combinations and find that 92.31% and 93.01% of triads in Epinions and Slashdot are balanced, respectively.

In status theory, a positive link from u_i to u_j indicates that u_i has a higher status than u_j; while a negative link from u_i to u_j indicates that u_i has a lower status than u_j. For a triad, status theory suggests that if we take each negative link, reverse its direction, and flip its sign to positive, then the resulting triangle (with all positive link) should be acyclic. We first obtain all triads and then follow the above way to examine whether these triads satisfy status theory or not. We find that 94.73% and 93.38% of triads in Epinions and Slashdot satisfy status theory, respectively.

3.3 Negative Links and Content-centric Interactions

Content-centric interactions relate the opinion of user u_i on the content posted by user u_j. The user u_i can express negative opinions on content posted by another user u_j by disliking, giving negative comments, or negative ratings. Such types of content-centric interactions may be viewed as negative interactions between u_i and u_j. A negative interaction from u_i to u_j is often a manifestation of user u_i's disagreement and antagonism toward u_j. It is therefore reasonable to surmise that negative interactions might be correlated with negative links. In this subsection, we study the correlation between negative interactions and negative links.
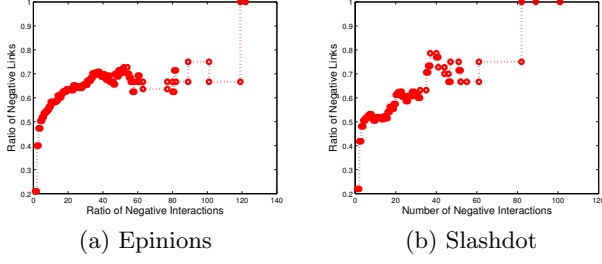
(a) Epinions   (b) Slashdot

Figure 3: The Ratios of Negative Links with respect to the Number of Negative Interactions.

Let $\mathbf{N} \in \mathbb{R}^{m \times m}$ be a user-user negative interaction matrix where $\mathbf{N}_{ij}$ denotes the number of negative interactions from u_i to u_j. We can obtain $\mathbf{N}$ from the user-content authorship matrix $\mathbf{A}$ and the user-content opinion matrix $\mathbf{O}$ as $\mathbf{N} = -\mathbf{A}(\mathbf{O}^-)^\top$ where $\mathbf{O}^- = \frac{\mathbf{O}-|\mathbf{O}|}{2}$ is the negative part of $\mathbf{O}$. To verify the correlation between negative interactions and negative links, we aim to answer the following question: Are pairs of users with negative interactions more likely to have negative links than those without negative interactions?

For each pair $\langle u_i, u_j \rangle$ with negative interactions (or $\mathbf{N}_{ij} \neq 0$), we first randomly select a user u_k that u_i does not have negative interactions with (or $\mathbf{N}_{ij} = 0$), and then use S (or R) to indicate whether $\langle u_i, u_j \rangle$ (or $\langle u_i, u_k \rangle$) has a negative link where $S = 1$ (or $R = 1$) if u_i has a negative link to u_j (or u_i has a negative link to u_k), otherwise $S = 0$ (or $R = 0$). Let $\mathbf{s}$ be a vector of Ss over all pairs of users with negative interactions and $\mathbf{r}$ be the corresponding vector of Rs. We conduct a two-sample t-test on $\mathbf{s}$ and $\mathbf{r}$. The null hypothesis and the alternative hypothesis are defined as $H_0 : \mathbf{s} \leq \mathbf{r}$, $\quad H_1 : \mathbf{s} > \mathbf{r}$. The null hypothesis is rejected at significance level $\alpha = 0.01$ with p-values of 5.72e-89 and 1.93e-109 for Epinions and Slashdot, respectively. Evidence from the t-test suggests a positive answer to the question: *there is a strong correlation between negative interactions and negative links, and users with negative interactions are likely to have negative links.*

We further investigate the direct impact of negative interactions on negative links. For a given value of K, we calculated the ratio of pairs with both negative links and at least K negative interactions over all pairs with at least K negative interactions. The ratio distributions with respect to the number of negative interactions are demonstrated in Figures 3(a) and 3(b), respectively. Note that the ratios of randomly selected pairs with negative links among all $n(n-1)$ pairs of users are 2.4177e-04 and 3.9402e-04 in Epinions and Slashdot, respectively. From the figures, we note that the ratios are much higher than the random ones even when K is very small. This observation further supports the existence of the correlation between negative interactions and negative links. Furthermore with increase of K, the ratios tend to increase. Therefore, an increase in the number of negative interactions increases the likelihood of negative links between users.

3.4 Discussion

We summarize the insights obtained in the aforementioned discussion as follows:

- Most of our "enemies" are close to us within a few (e.g., 2 or 3) hops in the positive network.

- Most of triads in signed networks satisfy balance theory and status theory.

- Pairs of users with negative interactions are more likely to have negative links than those without them.

- Negative interactions between users increase the propensity of negative links.

These observations provide the groundwork for our proposed framework for negative link prediction.

4. THE PROPOSED FRAMEWORK - NeLP

Algorithms for all variations of the link prediction problem are either unsupervised methods [15, 10] or supervised methods [17, 13]. Supervised methods consider the link prediction problem as a classification problem by using the existence of links as labels and have several advantages over unsupervised methods such as superior performance, adaptability to different data domains, and variance reduction [17]. Similar to traditional supervised link prediction, we can consider the negative link prediction problem as a classification problem where we need to construct labels and extract features. Different from traditional link prediction, there are unique challenges in preparing training data in the negative link prediction problem. For example, existing links are given in traditional link prediction such as positive links in positive link prediction, and positive and negative links in positive and negative link prediction, while existing negative links are not given in the negative link prediction problem. Next we will first give details about label construction and feature extraction for the negative link prediction problem. Finally, the optimization formulation and solution will be discussed.

4.1 Label Construction

Let E_o denote pairs of users with missing links. In most previous formulations of link prediction, including the signed version, label construction is trivial because the presence of links is specified. However, we study the scenario where no negative links are provided, and therefore the labels for E_n are really an unspecified subset of $E_o \cup E_n$. What is worse, the sizes of E_n and E_o are extremely imbalanced. For example, the imbalance ratios $E_n : E_o$ are 1:4131 and 1:2534 in Epinions and Slashdot respectively. We treat missing links as positive samples and negative links as negative samples. Label construction is to construct positive and negative samples from $E_o \cup E_n$. Since the ratio of E_o in $E_n \cup E_o$ are often bigger than 99.9%, we can randomly select a subset of samples from $E_n \cup E_o$ as positive samples PS. Next we introduce a way to select samples from $E_n \cup E_o$ as negative samples based on previous observations. The process is shown in Algorithm 1.

Next, we describe Algorithm 1 for negative sample construction. The strong correlation between negative interactions and negative links suggests that users with negative interactions are likely to have negative links. Therefore from line 2 to line 4 in Algorithm 1, we construct negative sample candidates based on this observation. With the positive links from $\mathcal{G}_p$ and negative links u_i-u_j from NS, we construct a signed network $\mathcal{G}$ in line 5. Most of the triads in signed networks satisfy status theory. Therefore we refine NS by (a) excluding $\langle u_i, u_j \rangle$ from NS if u_i-u_j is in any triads of $\mathcal{G}$ that does not satisfy status theory in line 6; and (b)

Algorithm 1: Negative Sample Construction

Input : The positive network $\mathcal{G}_p$ and user-user negative interaction matrix $\mathbf{N}$

Output : Negative sample set NS and the reliability weight matrix $\mathbf{W}$

1: Initialize $NS = \emptyset$
2: **for all** $\mathbf{N}_{ij} \neq 0$ **do**
3:    $NS = NS \cup \{\langle u_i, u_j \rangle\}$
4: **end for**
5: Construct $\mathcal{G}$ as a signed network with positive links from $\mathbf{G}_p$ and negative links u_i-u_j from NS
6: Remove samples $\langle u_i, u_j \rangle$ from NS if u_i-u_j is in any triads of $\mathcal{G}$ that does not satisfy status theory
7: Add samples $\langle u_i, u_k \rangle$ into NS if u_i-u_k can make all triads that involve u_i and u_k in $\mathcal{G}$ satisfying status theory
8: **for all** $\langle u_i, u_j \rangle \in NS$ **do**
9:    Calculate a reliability weight $\mathbf{W}_{ij}$
10: **end for**

---

adding samples $\langle u_i, u_k \rangle$ into NS if u_i-u_k can make all triads that involve u_i and u_k in $\mathcal{G}$ satisfying status theory in line 7. The reliability of these negative samples may vary. For example, observations from data analysis indicate that negative sample candidates with more negative interactions are more likely to have negative links, and are therefore more likely to be reliable. Therefore, we associate each $\langle u_i, u_j \rangle$ with a reliability weight $\mathbf{W}_{ij}$, which is defined as follows:

$$\mathbf{W}_{ij} = \begin{cases} f(\mathbf{N}_{ij}) & \text{if } \mathbf{N}_{ij} \neq 0 \\ r & \text{otherwise} \end{cases}. \qquad (2)$$

if the pair $\langle u_i, u_j \rangle \in NS$ has negative interactions, we define the reliability weight as a function f of the number of negative interactions $\mathbf{N}_{ij}$ where $f(x) \in [0, 1]$ is a non-decreasing function of x. This is because the more negative interactions two users have, the more likely it is that a negative link exists between them. Otherwise, the pair $\langle u_i, u_j \rangle \in NS$ is added by line 7 in Algorithm 1 and we set the reliability weight to a constant r.

4.2 Feature Extraction

We extract three types of features corresponding to user features, pair features and sign features. User features and pair features are extracted from two given sources, such as positive links and content-centric interactions, as follows:

- User features are extracted for each user u_i including u_i's indegree (or outdegree) in terms of positive links, the number of triads that u_i involved in, the number of content-centric items (e.g., posts) that u_i creates, the number of u_i's posts that obtain positive (or negative) opinions, and the number of positive (or negative) opinions u_i expresses; and

- Pair features are extracted for a pair of users $\langle u_i, u_j \rangle$ including the number of positive (or negative) interactions from u_i to u_j, the number of positive (or negative) interactions from u_j to u_i, Jaccard coefficients of indegree (or outdegree) of u_i and u_j in terms of positive links, and the length of the shortest path between u_i and u_j.

We construct a weighted signed network with the given positive links and negative links from NS where the weights of positive links are 1 and the weights of negative links are their reliability weights. For a pair $\langle u_i, u_j \rangle$, signed features include weighted indegree (or outdegee) in terms of negative links of u_i, weighted indegree (and outdegee) in terms of negative links of u_j, Jaccard coefficients of indegree (or outdegree) of u_i and u_j in terms of negative links and 16 weighted triads suggested by [13].

With definitions of user features, pair features and sign features, we extract 45 features in total for each pair $\langle u_i, u_j \rangle$ including 8 user features of u_i, 8 user features of u_j, 7 pair features, and 22 signed features.

4.3 The NeLP Optimization Framework

Through label construction and feature extraction, we prepare training data to learn classifiers for the negative link prediction problem. However, the labels of the training data are noisy, and especially so for negative samples. Therefore, it is necessary for the base classifier to be tolerant to training data noise. In this paper, we choose a soft-margin version of support vector machines as our basic classifier because it has been proven to be highly noise-tolerant [5].

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ be the set of user pairs in $E_o \cup E_n$ and $\mathbf{x}_i$ be the feature vector representation of the pair x_i. The standard soft-margin support vector machine for the negative link prediction problem is as follows:

$$\min_{\mathbf{w}, b, \epsilon} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{x_i \in PS \cup NS} \epsilon_i$$
$$s.t. \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \epsilon_i, \; x_i \in PS \cup NS$$
$$\epsilon_i \geq 0 \;\; x_i \in PS \cup NS \qquad (3)$$

Eq. (3) introduces the term ϵ_i for the soft-margin slack variable of x_i, which can be viewed as the allowance for the noise in this training sample. The parameter C controls the degree of impact of this term. In the negative link-prediction problem, the noise-levels of positive and negative samples are different because positive samples PS are generally more robust than the (indirectly derived) negative samples. As discussed earlier, the reliability of negative samples is explicitly quantified with their weights. These intuitions suggest that we should allow more errors in negative samples especially when their weights suggest unreliability. This yields the following formulation:

$$\min_{\mathbf{w}, b, \epsilon} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C_p \sum_{x_i \in PS} \epsilon_i + C_n \sum_{x_j \in NS} c_j \epsilon_j$$
$$s.t. \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \epsilon_i, \;\; x_i \in PS$$
$$y_j(\mathbf{w}^\top \mathbf{x}_j + b) \geq 1 - \epsilon_j, \;\; x_j \in NS$$
$$\epsilon_i \geq 0, \; \epsilon_j \geq 0 \qquad (4)$$

In Eq. (4), we use two parameters C_p and C_n to weight the positive and negative errors differently. We use a larger value for C_p compared to C_n to reflect the differential behavior of the positive and negative samples. For a negative sample x_j, we introduce a weight c_j to further control its error based on its quantified reliability weight. For the negative sample x_j corresponding to the pair $\langle u_i, u_k \rangle$, we set $c_j = \mathbf{W}_{ik}$ where $\mathbf{W}_{ik}$ is the reliability weight for $\langle u_i, u_k \rangle$. This additional term allows differential control of the noise in negative samples of varying reliability.

Balance theory suggests that triads in signed networks are likely to be balanced. Therefore, if there is a positive link between u_i and u_j, and both u_i and u_j do not have positive links with another user u_k, the types of (u_i, u_k) and (u_j, u_k) in the negative graph $\mathcal{G}_n$ are likely to be the same. In other words, to ensure the structural balance, it is likely that both are negative links where $\langle u_i, u_j, u_k \rangle$ forms a balanced triad or both are missing links where there is no triad among $\langle u_i, u_j, u_k \rangle$. With this intuition, we introduce a matrix $\mathbf{B}$ where $\mathbf{B}_{h\ell} = 1$ if there is a positive link between u_i and u_j where we assume that x_h and x_ℓ denote pairs $\langle u_i, u_k \rangle$ and $\langle u_j, u_k \rangle$ respectively. Otherwise, we assume that $\mathbf{B}_{h\ell} = 0$. Then, we force x_h and x_ℓ to have the same types of links if $\mathbf{B}_{h\ell} = 1$ by introducing a balance-theory regularization:

$$\min \quad \frac{1}{2} \sum_{h,\ell} \mathbf{B}_{h\ell} (\mathbf{w}^\top \mathbf{x}_h - \mathbf{w}^\top \mathbf{x}_\ell)_2^2 = \mathbf{w}^\top \mathbf{X} \mathcal{L} \mathbf{X}^\top \mathbf{w} \qquad (5)$$

Here, $\mathcal{L}$ is the Laplacian matrix based on $\mathbf{B}$. The number of pairs in $E_n \cup E_o$ is usually very large, which leads to a large number of terms in the balance theory regularization. The observation from data analysis suggests that our "enemies" are usually close to us in the positive network. Hence, in this work, we only consider pairs whose shortest path lengths are 2, and pairs in NS and PS in the balance theory regularization. We assume that there are $l + \mu$ samples in $\mathbf{X}$ where the first l ones are from $PS \cup NS$. The significance of the introduction of the balance theory regularization is two-fold. First, it allows us to model balance theory. Second, it allows us to include more samples during the learning process in addition to NS and PS. A similar function is achieved by this approach, as achieved by unlabeled samples in semi-supervised learning [28]. With these components, the proposed NeLP framework is able to solve the following optimization problem:

$$\min_{\mathbf{w}, b, \epsilon} \quad \frac{1}{2} \|\mathbf{w}\|_2^2 + C_p \sum_{x_i \in PS} \epsilon_i + C_n \sum_{x_j \in NS} c_j \epsilon_j$$
$$+ \frac{C_b}{2} \mathbf{w}^\top \mathbf{X} \mathcal{L} \mathbf{X}^\top \mathbf{w}$$
$$s.t. \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \epsilon_i, \quad x_i \in PS$$
$$y_j(\mathbf{w}^\top \mathbf{x}_j + b) \geq 1 - \epsilon_j, \quad x_j \in NS$$
$$\epsilon_i \geq 0, \ \epsilon_j \geq 0 \qquad (6)$$

We solve the optimization problem in Eq. (6) based on the dual form [1]. The classical representer theorem states that the solution to this minimization problem of Eq.(6) exists in $\mathcal{H}_K$ and can be written as follows:

$$\mathbf{w}^* = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) \qquad (7)$$

Eq.(6) can be rewritten as follows:

$$\min_{\alpha, b, \epsilon} \quad \frac{1}{2} \alpha^\top \mathbf{K} \alpha + C_p \sum_{u_i \in PS} \epsilon_i + C_n \sum_{u_j \in NS} c_j \epsilon_j + \frac{C_b}{2} \alpha^\top \mathbf{K} \mathcal{L} \mathbf{K} \alpha$$
$$s.t. \quad y_i(\sum_k \alpha_k K(\mathbf{x}_k, \mathbf{x}_i) + b) \geq 1 - \epsilon_i, \quad u_i \in PS$$
$$y_j(\sum_k \alpha_k K(\mathbf{x}_k, \mathbf{x}_j) + b) \geq 1 - \epsilon_j, \quad u_j \in NS$$
$$\epsilon_i \geq 0, \ \epsilon_j \geq 0 \qquad (8)$$

where $\mathbf{K}$ is the Gram matrix over all samples.

We define s_i for x_i as follows:

$$s_i = \begin{cases} C_p & \text{for } x_i \in PS, \\ C_n c_i & \text{for } x_i \in NS. \end{cases} \qquad (9)$$

After the introduction of two sets of multipliers β and γ, the Lagrangian function of Eq.(8) is as follows:

$$L(\mathbf{w}, b, \epsilon, \alpha, \gamma) = \frac{1}{2} \alpha^\top (\mathbf{K} + C_b \mathbf{K} \mathcal{L} \mathbf{K}) \alpha + \sum_{i=1}^l s_i \epsilon_i$$
$$- \sum_{i=1}^l \beta_i [y_i (\sum_k \alpha_k K(\mathbf{x}_k, \mathbf{x}_i) + b) - 1 + \epsilon_i] - \sum_{i=1}^l \gamma_i \epsilon_i \quad (10)$$

where β and γ are Lagrange multipliers.

To obtain the dual representation, we set

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^l \beta_i y_i = 0$$
$$\frac{\partial L}{\partial \epsilon_i} = 0 \Rightarrow s_i - \beta_i - \gamma_i = 0 \Rightarrow 0 \leq \beta_i \leq s_i \qquad (11)$$

With Eq. (11), we can rewrite the Lagrangian as a function of only α and β as follows:

$$L(\alpha, \beta) = \frac{1}{2} \alpha^\top (\mathbf{K} + C_b \mathbf{K} \mathcal{L} \mathbf{K}) \alpha - \alpha^\top \mathbf{K} \mathbf{J}^\top \mathbf{Y} \beta + \sum_{i=1}^l \beta_i \qquad (12)$$

in Eq. (12), $\mathbf{J} = [\mathbf{I} \ \mathbf{0}]$ where $\mathbf{I}$ is an $l \times l$ identity matrix and $\mathbf{0}$ is a $l \times \mu$ rectangular matrix with all zeros, and $\mathbf{Y}$ is a $l \times l$ diagonal matrix composed by labels of samples in PS and NS.

By setting $\frac{\partial L}{\partial \alpha} = 0$, we obtain

$$\alpha = (\mathbf{I} + C_b \mathbf{K} \mathcal{L})^{-1} \mathbf{J}^\top \mathbf{Y} \beta \qquad (13)$$

After substituting back in the Lagrangian function, we obtain the dual problem as a quadratic programming problem:

$$\max_\beta \quad \sum_{i=1}^l \beta_i - \frac{1}{2} \beta^\top \mathbf{Q} \beta$$
$$s.t. \quad \sum_{i=1}^l \beta_i y_i = 0$$
$$0 \leq \beta_i \leq s_i \qquad (14)$$

where $\mathbf{Q}$ is defined as follows:

$$\mathbf{Q} = \mathbf{Y} \mathbf{J} \mathbf{K} (\mathbf{I} + C_b \mathbf{K} \mathcal{L})^{-1} \mathbf{J}^\top \mathbf{Y} \qquad (15)$$

5. EXPERIMENTS

In this section, we present experiments which (a) quantify the performance of the proposed NeLP framework in predicting negative links, and (b) evaluate the contribution of various model components to the performance. We begin by introducing performance evaluation metrics, which are useful in both contexts.

5.1 Evaluation Metrics

All forms of link prediction can be viewed as highly imbalanced classification problems. In such cases, straightforward accuracy measures are well known to be misleading [25]. For example, in a sparse network, the trivial classifier that labels all samples as missing links can have a 99.99% accuracy. In negative link prediction, we aim to achieve high

Table 2: Confusion Matrix of a Binary Classifier.

	True class = -1	True class = 1
Predicted class = -1	true pos. (tp)	false pos. (fp)
Predicted class = 1	false neg. (fn)	true neg. (tn)

precision and recall over negative links, defined in terms of the confusion matrix of a classifier as shown in Table 2: $precision = \frac{tp}{tp+fp}$ and $recall = \frac{tp}{tp+fn}$. Usually precision and recall are combined into their harmonic mean, the F-measure. Therefore we will adopt F1-measure as one metric for the performance evaluation. As suggested in [25], in some scenarios, we put more emphasis on precision because the most challenging task is to seek some negative links with high probability, even at the price of increasing false negatives. Hence, we also report the precision performance.

5.2 Performance of Negative Link prediction

In this subsection, we assess the proposed framework in terms of (a) the performance of NeLP with respect to baseline methods; and (b) the generalization of the proposed framework across social media sites. For the first evaluation, we define the following baseline methods:

- *Random*: This predictor randomly guesses the labels of samples. As suggested in positive link prediction [16], a random predictor should be used as a baseline method to meaningfully demonstrate the performance significance of other predictors;

- *sPath*: Observations in data analysis suggest that our "enemies" are always close to us in the positive network and *sPath* assigns negative links to pairs whose shortest path lengths is L;

- *negIn*: Given the strong correlation between negative interactions and negative links, *negIn* suggests negative links to these pairs with negative interactions;

- *negInS*: after obtaining negative link candidates via *negIn*, *negInS* further refines these candidates by performing a removing step and an adding step as shown in Algorithm 1; and

- *NeLP-negIn*: NeLP-negIn is a variant of the proposed NeLP framework. Instead of using negative links suggested by *negInS* as NeLP, NeLP-negIn uses negative links found by *negIn*.

For parameterized methods, we report the best performance of each baseline method. For NeLP, we set its parameters as $\{C_p = 1, C_n = 0.5, C_b = 0.1\}$ and $\{C_p = 1, C_n = 0.7, C_b = 0.01\}$ in Epinions and Slashdot, respectively. We empirically find that $f(x) = 1 - \frac{1}{\log(1+x)}$ works well for the proposed framework. More details about parameter sensitivity of NeLP will be discussed in later subsections. The comparison results are demonstrated in Table 3.

We make the following observations:

- *sPath* obtains much better performance than random guessing, which further supports the hypothesis that our "enemies" are close to us in the positive network;

- *negIn* improves the performance significantly in both datasets. These results suggest the existence of correlation between negative interactions and negative links;

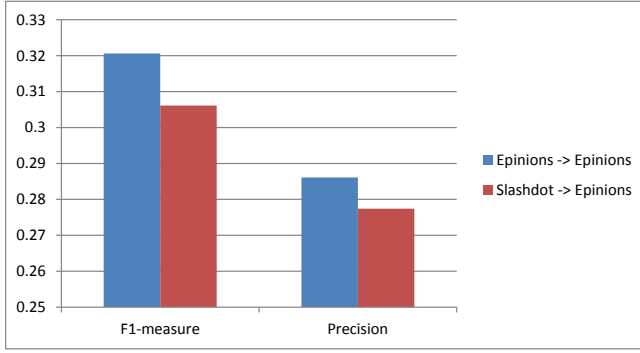**Table 3: Performance Comparison of Negative Link Prediction in Epinions and Slashdot.**

Algorithms	Epinions		Slashdot	
	F1	Precision	F1	Precision
random	0.0005	0.0002	0.0008	0.0004
sPath	0.0040	0.0075	0.0090	0.0172
negIn	0.2826	0.2097	0.1986	0.1483
negInS	0.2893	0.2124	0.2072	0.1524
NeLP-negIn	0.3206	0.2812	0.2394	0.2083
NeLP	0.3242	0.2861	0.2441	0.2139

- by removing candidates suggested by *negIn* that do not satisfy status theory and adding candidates to make open triads closure to satisfy status theory, *negInS* outperforms *negIn*. For example, *negInS* gains 2.37% and 1.45% relative improvement in terms of F1-measure in Epinions and Slashdot, respectively. These results indicate that status theory can help us remove some noisy samples and add some useful samples for training. These observations can also be used to explain the reason why the performance of NeLP based on negative links suggested by *negInS* is better than that based on *negIn*; and

- the proposed framework always obtains the best performance. There are three important components of NeLP. First, NeLP introduces C_n to control errors in negative samples. Second, NeLP introduces c_j to control the error in the sample x_j, which is related to the number of negative interactions based on our observations from data analysis. Third, NeLP introduces balance theory regularization to model balance theory, which also allows us to include more samples in the classifier learning process. More details about the effects of these components will be discussed in a later subsection.
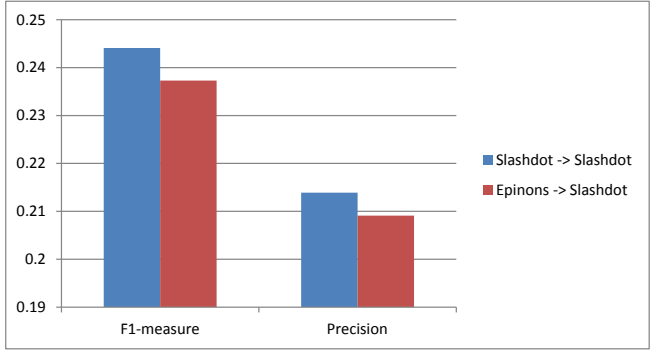
Because the classifier learned by the proposed framework is based on the same set of features extracted from pervasively available sources for most social media sites, it is possible to generalize the classifier learned in one site to other sites and we further investigate how well the learned classifier generalizes across social media sites. In particular, we evaluate the performance of the classifier on Epinions (or Slashdot), which is learned from Slashdot (or Epinions). The results are shown in Figure 4. Note that in the figure $x \to y$ denotes training on x and evaluating on y. These results show that there is very good generalization of the classifier learned by NeLP although there is remarkably little decrease in performance regardless of which dataset is used for training. In summary, compared to baseline methods, the proposed framework obtains significant performance improvement, and it also has very good generalization across social media sites.

5.3 Component Analysis of NeLP

In this subsection, we investigate the effects of various NeLP components on its performance. In NeLP, we introduce C_n, c_j and C_b to control three components of NeLP. In particular, C_n controls errors from negative samples, c_j controls the error from the negative sample x_j and C_b controls the contribution from the balance theory regularization. By setting $C_p = 1$ and varying different values of C_n, c_j and C_b, we can examine the impact of these components on the

(a) Epinions   (b) Slashdot

Figure 4: The Negative Link Prediction Performance across Epinions and Slashdot where $x \to y$ denotes training on x and evaluating on y.

Table 4: Component analysis for NeLP in Epinions.

C_n	c_j	C_b	F1-measure	Precision
0.5	$f(x) = 1 - \frac{1}{\log(1+x)}$	0.1	0.3242	0.2861
1	$f(x) = 1 - \frac{1}{\log(1+x)}$	0.1	0.3188	0.2793
0.5	$f(x) = 1$	0.1	0.3067	0.2612
0.5	$f(x) = 1 - \frac{1}{\log(1+x)}$	0	0.3084	0.2686
1	$f(x) = 1$	0	0.2992	0.2342

Table 5: Component Analysis of NeLP in Slashdot.

C_n	c_j	C_b	F1-measure	Precision
0.7	$f(x) = 1 - \frac{1}{\log(1+x)}$	0.01	0.2441	0.2139
1	$f(x) = 1 - \frac{1}{\log(1+x)}$	0.01	0.2403	0.2094
0.7	$f(x) = 1$	0.01	0.2287	0.1902
0.7	$f(x) = 1 - \frac{1}{\log(1+x)}$	0	0.2347	0.1972
1	$f(x) = 1$	0	0.2213	0.1722

performance of NeLP. The results of component analysis are shown in Tables 4 and 5 for Epinions and Slashdot, respectively.

The first row in each table represents the performance of NeLP with all three components. We make the following observations about different variations of NeLP in other rows of the table:

- in the second row, we set $C_n = 1$, which gives equal weights to positive and negative samples. This approach effectively eliminates the *differential* importance given to errors from negative samples. The performance degrades, which suggests that the errors of negative and positive samples should be treated differently;

- in the third row, we set $c_j = 1$ instead of the reliability weight related to the number of negative interactions to eliminate the component controlling the error in the negative sample x_j. The performance reduces a lot. For example, the precision reduces by 8.70% and 11.08% in Epinions and Slashdot, respectively. These results support the importance of the number of negative interactions to indicate the reliability of negative samples;

- in the fourth row, we set $C_b = 0$ to eliminate the contribution from the balance theory regularization. and the performance is consistently worse than that with

the balance theory regularization. This illustrates the importance of the balance theory regularization in the proposed NeLP framework; and

- in the fifth row, we eliminate all these three components and the performance further degrades. These results suggest that the three components contain complementary information.

5.4 Impact of Balance Theory Regularization

The analysis of the previous subsection shows the importance of balance theory regularization. In this subsection, we perform a more detailed analysis of the impact of the balance theory regularization on NeLP by showing how the performance varies with the changes in the value of C_b. This parameter controls the contribution from the balance theory regularization. We vary the values of C_b as $\{0, 0.001, 0.01, 0.05, 0.1, 0.5, 1\}$ and the results are shown in Figures 5(a) and 5(b) for Epinions and Slashdot, respectively. In general, with increase in C_b, the performance first improves, peaks, and then drops dramatically.

In particular, with the increase of C_b, we make the following observations:

- By increasing the value of C_b from 0 to 0.001, the performance increases significantly. For example, NeLP gains 4.02% and 4.41% in terms of precision in Epinions and Slashdot, respectively. These results further support the importance of the balance theory regularization in the NeLP framework;

- Within certain parameter ranges, such as from 0.001 to 0.1 in Epinions, the performance is relatively stable. This property is practically useful because it makes it easier to set C_b; and

- After certain values such as 0.05 in Slashdot, the performance decreases dramatically. A large value C_b results in balance theory regularization dominating the learning process at the expense of other factors.

In summary, the parameter analysis on C_b suggests that balance theory regularization is important for the NeLP framework.

6. RELATED WORK

In this section, we briefly review work which is related to different variants of the link prediction problem.
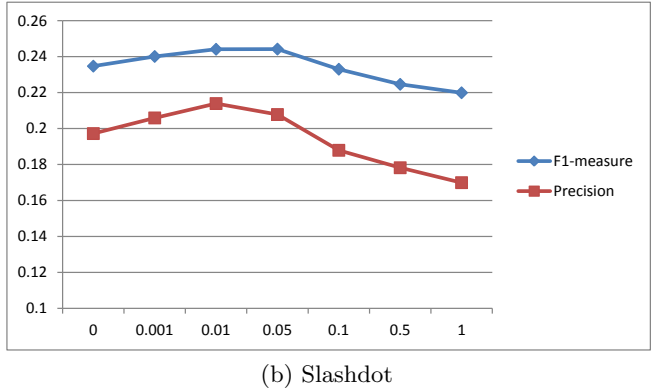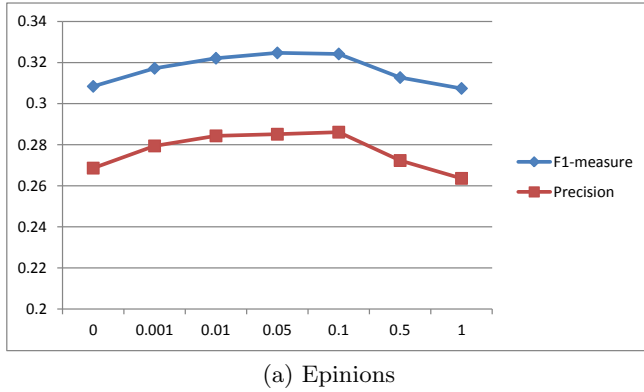
(a) Epinions

(b) Slashdot

Figure 5: The Impact of the Balance Theory Regularization on NeLP

6.1 Positive Link Prediction

Positive link prediction infers new positive links in the near future based on a snapshot of a positive network. Existing methods can be roughly divided into unsupervised methods and supervised methods. Unsupervised methods are usually based on the topological structure of the given positive network. In [15], several unsupervised link prediction algorithms are proposed, such as Katz, Jaccard's coefficient and Adamic/Adar. In [19], several unsupervised algorithms based on low-rank matrix factorization are proposed. There are usually two steps for supervised methods. First, they extract features from available sources to represent each pair of users and consider the existence of positive links as labels. Second, they train a binary classifier based on the representation with extracted features and labels. In [17], the authors show several advantages of supervised link prediction algorithms such as superior performance, adaptation to different domains and variance reduction. In [25], the features extracted from human mobility have very strong predictive power and can significantly improve the positive link prediction performance.

6.2 Positive and Negative Link Prediction

Positive and negative link prediction infers new positive and negative links by giving a snapshot of a signed network, which has attracted increasing attention in recent years [?]. In [7], an algorithm based on trust and distrust propagation is proposed to predict trust and distrust relations. In [13], local-topology-based features based on balance theory are extracted to improve the performance of a logistic regression classifier in signed relation prediction. Features derived from longer cycles in signed networks can be used to improve the positive and negative link prediction performance [4]. In [10], a low-rank matrix factorization approach with generalized loss functions is proposed to predict trust and distrust relations.

6.3 Sign Prediction

Sign prediction infers the sign of a given link. In [26], user behavior of decision making can be used to predict signs of a given unsigned network accurately. The authors also show the importance of modeling balance theory and status theory in the sign prediction problem. Tang et al. proposed a framework to incorporate social theories such as status theory into a machine learning model and infer the signs of links in a target network by borrowing knowledge from

a different source network [23]. In [27], the authors use the transfer learning approach to leverage sign information from an existing and mature signed network to predict signs for a newly formed signed social network.

7. CONCLUSION

Research in signed network analysis suggests that negative links have added value over positive links and they can potentially help various social media services such as recommender systems. However, most social media sites do not enable their users to specify negative links. This makes the problem of negative link prediction more challenging. In this paper, we investigate the problem of negative link prediction. To preserve the generality of our approach, we use positive links and content-centric interactions as sources to predict negative links because these two sources are pervasively available in social media. We first analyze the impact of various social theories, such as balance theory and status theory, on negative links. Then we leverage these insights to provide a principled way to exploit positive links and content-centric interactions. Finally, we propose the NeLP framework for negative link prediction. Experimental results on two real-world social media datasets demonstrate the effectiveness and generalization of the proposed framework. Further experiments illustrate the impact of various model components.

There are several interesting directions needing further investigation. First, negative links predicted by the proposed framework may benefit various social media applications such as positive link prediction and recommender systems. Therefore, we plan to investigate how to incorporate the proposed framework into these applications to improve their performance. Second, in addition to positive links and content-centric interactions, the user-generated content is also pervasively available in social media. We would like to investigate whether user generated content is useful and how to exploit it for the negative link prediction problem. Finally the constructed labels in the studied problem may be noisy; hence we choose a noise-tolerant support vector machine as the basic algorithm for the proposed framework. Learning with noisy labels has been extensively studied in the machine learning community [6] and we will experiment with the use of other noise-tolerant algorithms as basic algorithms for the problem of negative link prediction.

Acknowledgments

8. REFERENCES

[1] M. Belkin, P. Niyogi, and V. Sindhwani. On manifold regularization. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005)*, pages 17–24, 2005.

[2] D. Cartwright and F. Harary. Structural balance: a generalization of heider's theory. *Psychological Review*, 63(5):277, 1956.

[3] K.-Y. Chiang, C.-J. Hsieh, N. Natarajan, A. Tewari, and I. S. Dhillon. Prediction and clustering in signed networks: A local to global perspective. *arXiv preprint arXiv:1302.5145*, 2013.

[4] K.-Y. Chiang, N. Natarajan, A. Tewari, and I. S. Dhillon. Exploiting longer cycles for link prediction in signed networks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1157–1162. ACM, 2011.

[5] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[6] B. Frénay, M. Verleysen, et al. Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, pages 25–17, 2013.

[7] 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.

[8] R. Hardin. Distrust: Manifestations and management. *Distrust*, 8:3–33, 2004.

[9] F. Heider. Attitudes and cognitive organization. *The Journal of psychology*, 21(1):107–112, 1946.

[10] C.-J. Hsieh, K.-Y. Chiang, and I. S. Dhillon. Low rank modeling of signed networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 507–515. ACM, 2012.

[11] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*, pages 741–750. ACM, 2009.

[12] 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.

[13] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, 2010.

[14] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1361–1370. ACM, 2010.

[15] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of 12th International Conference on Information and Knowledge Management*, 2003.

[16] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[17] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 243–252. ACM, 2010.

[18] H. Ma, M. R. Lyu, and I. King. Learning to recommend with trust and distrust relationships. In *Proceedings of the third ACM conference on Recommender systems*, pages 189–196. ACM, 2009.

[19] A. K. Menon and C. Elkan. Link prediction via matrix factorization. In *Machine Learning and Knowledge Discovery in Databases*, pages 437–452. Springer, 2011.

[20] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3):515–554, 2012.

[21] J. Tang, X. Hu, and H. Liu. Social recommendation: a review. *Social Network Analysis and Mining*, 3(4):1113–1133, 2013.

[22] J. Tang and H. Liu. Feature selection with linked data in social media. In *SDM*, pages 118–128. SIAM, 2012.

[23] J. Tang, T. Lou, and J. Kleinberg. Inferring social ties across heterogenous networks. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 743–752. ACM, 2012.

[24] 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.

[25] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A. Barabási. Human mobility, social ties, and link prediction. In *KDD*, pages 1100–1108. ACM, 2011.

[26] S.-H. Yang, A. J. Smola, B. Long, H. Zha, and Y. Chang. Friend or frenemy?: predicting signed ties in social networks. In *SIGIR*, pages 555–564. ACM, 2012.

[27] J. Ye, H. Cheng, Z. Zhu, and M. Chen. Predicting positive and negative links in signed social networks by transfer learning. In *WWW*, pages 1477–1488, 2013.

[28] X. Zhu, Z. Ghahramani, J. Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.