The Troll-Trust Model for Ranking in Signed Networks

Zhaoming Wu
Georgia Institute of Technology
zhaoming@gatech.edu

Charu C. Aggarwal
IBM T. J. Watson Research Center
charu@us.ibm.com

Jimeng Sun
Georgia Institute of Technology
jsun@cc.gatech.edu

ABSTRACT

Signed social networks have become increasingly important in recent years because of the ability to model trust-based relationships in review sites like Slashdot, Epinions, and Wikipedia. As a result, many traditional network mining problems have been re-visited in the context of networks in which signs are associated with the links. Examples of such problems include community detection, link prediction, and low rank approximation. In this paper, we will examine the problem of ranking nodes in signed networks. In particular, we will design a ranking model, which has a clear physical interpretation in terms of the sign of the edges in the network. Specifically, we propose the *Troll-Trust* model that models the probability of trustworthiness of individual data sources as an interpretation for the underlying ranking values. We will show the advantages of this approach over a variety of baselines.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-Data Mining

Keywords

Signed Networks; Ranking; Data Mining

1. INTRODUCTION

Many social network sites, such[1] as Epinions, Slashdot, and Wikipedia, allow users to contribute opinions, feedback, and the ability to indicate their trust or distrust in each other's opinions. Such networks are useful in the context of a wide variety of tasks, such as recommendations, clustering, and classification because they tell us important aspects of the relations between nodes. The trust and distrust relationships in such networks are often represented in the form of *signed networks*. Some examples of such signed networks, are as follows:

1. *Epinions:* Epinions is an online review site, in which users can provide ratings to items. However, users can provide positive and negative ratings not just to items but to other raters. Such positive and negative opinions are represented as directed links between different users [5]. The sign of the link indicates whether the relationship is positive or negative.

2. *Slashdot:* Slashdot is a technology blog, where the users have the option of designating each other as friends or foes. Therefore, friends can be viewed as trusted parties (positive links), whereas foes can be viewed as distrusted parties (negative links).

3. *Wikipedia:* In Wikipedia, users are able to cast positive or negative votes for other admin users. Such votes can be viewed as positive and negative links.

As evident from the discussion above, the vast majority of signed networks arise in the context of trust networks. Such networks are very useful in improving the effectiveness of many applications such as collaborative filtering [7, 19], community detection [4, 11], and link prediction [4, 13, 14, 20]. In many of these applications, human-centric preferences play an important role; therefore, the trust component is relatively important in having an impact on the final results.

An important problem in network analytics is that of ranking nodes based on their reputation. A classical algorithm in this context is known as the *PageRank* method [3]. Ranking has traditionally been used for Web page ranking based on citation mechanisms. *PageRank* is viewed as a measure of reputation (or prestige) of a node in a network because it assumes that only reputable nodes are in-linked to by other nodes. In other words, each in-linking node is a vote of confidence in that node. Furthermore, the prestige of the in-linking node is also accounted for in the *PageRank* equations. However, the traditional *PageRank* equations are based on only the use of positive links. Developing methods for signed versions of *PageRank* is very useful in the context of a variety of applications such as global trust quantification of entities [23].

Over the last few years, a number of methods have been designed for signed network analysis with both positive and negative links [8, 9, 18, 21]. Most of these methods are based on simple modifications of the *PageRank* equations to account for negative weights on the links. However, many of these methods do not have natural interpretations in terms of the reputation values on the nodes. One of the reasons that the *PageRank* algorithm works so effectively is that it can naturally be interpreted in the form of a random surfer model with a clear physical interpretation for the importance of the nodes. While one can heuristically extend these equations with negative values on the links, we will show that the results can sometimes be unintended. In this paper,

[1] `http://snap.stanford.edu/data/index.html#signnets`

WSDM2016, 2016, San Francisco, 2016.

we will propose a *Troll-Trust* model with clear assumptions on what negative links mean in terms of the trust that users have on one another.

One of the reasons that trustworthy computing has become so important is the open nature of online forums in which users can intentionally post misleading information. Such users are referred to as *trolls*. Troll users might either have malicious intent, profit motives, or simply be inclined to behave in a disruptive way. Such users often have a serious impact on the quality of interaction in online forums. Therefore, the ability to identify users in signed networks based on their trustworthiness is crucial in a myriad of applications. Troll users can often be identified in signed networks, because they often have inbound negative links from other trustworthy users. This is similar to the *PageRank* reputation framework, except that the sign of the link also plays a critical role in reputation determination.

In our model, we use a Bernoulli distribution to characterize each user as either being trustworthy, or being a troll. This is used to construct a probabilistic model in terms of the links between various users. We show the advantages of using such an approach over heuristic extensions of *PageRank*. We also extend the approach to other variations such as personalized *Troll-Trust* ranking model. We show improved results of our techniques over competing methods in the literature.

This paper is organized as follows. The remainder of this section discusses the related work. The motivations of the *Troll-Trust* model are discussed in section 2. The iterative algorithm for the *Troll-Trust* model is illustrated in section 3. Methods for personalized *PageRank* in signed networks are discussed in section 4. The experiments are discussed in section 5. The conclusions are presented in section 6.

1.1 Related Work

Many data mining applications have been designed in the context of a wide variety of social networking applications [1]. In recent years, signed networks have gained increasing attention because of the ability of many social networking sites, such as Epinions, Slashdot, and Wikipedia, to specify trust and distrust relationships between users. Such relationships are useful in discerning between interactions of varying trustworthiness. The use of such networks can be helpful in applications such as collaborative filtering [7, 19], community detection [4, 11], and link prediction [4, 13, 14, 20].

Ranking is a well known problem in the context of unsigned networks [3, 10]. The *PageRank* problem has also been recently explored in the context of signed networks [8, 9, 18, 21]. Methods for finding node bias and prestige are discussed in [16], and this approach is based on the *HITS* algorithm. *PageRank* is useful for assessing the reputation of nodes, and it is also used in other applications such as sign prediction [18, 23]. One of the problems with existing variants of *PageRank* methods is that they heuristically try to modify the *PageRank* equations with the use of negative links. Such heuristic modifications often lose their interpretability, and they do not necessarily reflect the level of trust a given user has in another. In this paper, we will use a model, which is firmly grounded in probabilistic understanding of trust, which is the semantic interpretation in most signed networks. We show that the resulting approach is more robust than the traditional methods for signed *PageRank*.

2. MOTIVATION OF THE TROLL-TRUST MODEL

Before discussing the *Troll-Trust* model in detail, we will first revisit the *PageRank* method, and discuss how it is used in the context of a variety of applications. The traditional *PageRank* model was defined in the context of random surfer model on the Web. The *PageRank* model calculates the probability that a random surfer on the Web visits a given Web page by traversing links outgoing from Web pages in random fashion. In addition, in order to avoid the effect of dead-ends and smooth the ranking values, a restart probability is imposed on nodes. Let $G = (N, A)$ be a directed network with the node set N and the edge set A. Let us assume that the number of nodes in the set N is denoted by n. The nodes are indexed from $1 \ldots n$. The weights of the edges between nodes are stored in an $n \times n$ weight matrix W. Note that an edge weight is non-zero in W only when it is present in A. For traditional networks, all the weights in W are non-negative. In signed networks, however, some of these weights might be negative. The semantic interpretation of the sign in these cases is often one of trust or distrust.

In some signed networks, all weights in W are always drawn from $+1$ or -1, although this is not always the case. In some applications [20], positive or negative weights can be associated with the links, depending on the strength of the interactions. In the traditional random walk model, each node is associated with a probability of visit $\pi(i)$. Furthermore, the transition probability p_{ij} of each edge $(i, j) \in A$ is obtained by dividing each row in W by the sum of the corresponding entries in that row. This can be represented in matrix form as follows. Let D be the diagonal matrix in which the (i, i)th entry is given by the sum of the entries of W in the ith row.

$$D_{ii} = \sum_{j=1}^{n} w_{ij} \tag{1}$$

Then, the *PageRank* transition matrix is given by the following matrix $P = [p_{ij}]$:

$$P = D^{-1}W \tag{2}$$

Then, the *PageRank* equations may be stated in the form of a random surfer having transition probabilities p_{ij}. At each step, the surfer may restart to a random node with probability α. The steady state probability of visit of the random surfer is denoted by $\pi(i)$, and it represents the *PageRank* value. The steady state probability can be related to the transition probabilities by setting $\pi(i)$ to be the probability that the random surfer transitions into node i at a given step:

$$\pi(i) = (1 - \alpha) \cdot \sum_{j=1}^{n} \pi(j) \cdot p_{ji} + \alpha/n \tag{3}$$

Note that the values $\pi(i)$ intuitively represent transition probabilities, and they sum up to 1 over all the different nodes. Larger values of $\pi(i)$ indicate a better reputation for the node.

It is noteworthy that the random surfer model is intuitively interpretable only when the edge weights are non-negative and can be interpreted as transition probabilities.

The *PageRank* equations yield stochastic probabilities precisely because of this non-negativity. If one naturally tries to modify the *PageRank* equations with negative weights, it leads to negative ranking values. Unfortunately, this has the impact of making the equations less easily interpretable. In some cases, the resulting reputation values of $\pi(i)$ also become misleading. We will discuss this issue later in this section.

2.1 Problems with Existing Methods

In order to handle the problem of signed networks, a number of possible modifications have been made to the *PageRank* methodology. For example, the work in [9] multiplies the right-hand side of Equation 3 with a heuristic correction factor so that nodes with many incoming negative edges automatically have a lower reputation. The exponential ranking approach [21] uses an exponential variation of the *PageRank* equations in order to model the trust values of nodes. This approach still models the node-specific reputation in a heuristic as a (positive and negative) linear combination of the node-specific trust probabilities. The coefficients of the linear combination correspond to the edge weights. Although the approach provides a *relative* ranking of a heuristic nature, it is often hard to assign an absolute interpretability to the reputation values in terms of how much one should trust a particular user.

Recently, a very simple modification of *PageRank* [18] computes the *PageRank* separately on the positive and negative subgraphs. If the corresponding ranks for the node i are r_i^+ and r_i^-, respectively, then the overall reputation is computed as $r_i^+ - r_i^-$. Unfortunately, this approach does not account for the relative density of the positive and negative links in the network. For example, if a network contains a large number of positive links relative to the negative links, and vice versa, it should affect the reputation values. Unfortunately, the value of $r_i^+ - r_i^-$ is relatively insensitive to the proportion of positive and negative links. For example, if either the positive or the negative links are down-sampled in isolation, it will make only modest differences to the reputation values, as computed by $r_i^+ - r_i^-$. This scenario is intuitively undesirable. When the relative proportion of negative links in a network increases, it is likely that the network contains a larger number of trolls. This should change not only the absolute reputation values, but it should also have an impact on the relative ranking and the nature of the interaction between the various users.

A number of modifications of the *HITS* algorithm are also proposed in [18]. In all these cases, again the basic approach is divide the graph into positive and negative subgraphs to compute the hub and authority scores separately for each subgraph. The authority scores for the positive and negative subgraphs are subtracted from one another to provide the final score. However, this approach has similar issues to the modified *PageRank* method in that it is relatively insensitive to the relative proportion of positive and negative links.

2.2 The Troll-Trust Model

The key of the *Troll-Trust* model is to replace the random surfer model with a new set of probabilities associated with the nodes, which have a clear physical interpretation in terms of the global trustworthiness of the participants. Unlike the random surfer model, in which the steady-state probabilities define random walk probabilities, the probability $\pi(i)$ in this model defines the likelihood that a participant is either trustworthy or a troll. Let E_i be the event that the ith person is trustworthy. Therefore, we have:

$$E_i = \begin{cases} 1 & \text{Person } i \text{ is trustworthy with probability } \pi(i) \\ 0 & \text{Person } i \text{ is a troll with probability } 1-\pi(i) \end{cases} \tag{4}$$

Therefore, E_i is a Bernoulli random variable with the parameter $\pi(i)$, and our goal is to determine $P(E_i) = \pi(i)$, which gives us a measure of the reputation of individual i. Intuitively, one can also interpret this probability as the likelihood of a user providing misleading information in a trust-centric network where negative links have the semantic interpretation of untrustworthiness. Note that this interpretation of $\pi(i)$ is different from what is normally used in a random surfer model. The default (or *a priori*) value of $\pi(i)$, in a network is set to the global parameter β. Thus, if a network has no positive or negative interactions between users, then the value of $\pi(i)$ for all nodes would be β. The parameter β is useful for settings where the network is very sparse, although its impact is often somewhat limited.

Next, we define the significance of the weights $W = [w_{ij}]$ in the probabilistic modeling process. Intuitively, these weights represents statements of the level to which user i feels that user j is trustworthy. Therefore, we define the event U_{ij} as follows:

$$U_{ij} = \begin{cases} 1 & \text{Person } i \text{ views } j \text{ as trustworthy} \\ 0 & \text{Person } i \text{ views } j \text{ as a troll} \end{cases} \tag{5}$$

Then, we model the probability distribution of the Bernoulli variable U_{ij} as a logistic function of the weights w_{ij}. Large positive values of w_{ij} means that the expected value of U_{ij} should be close to 1. On the other hand, when the weight w_{ij} is negative, the expected value of U_{ij} should be close to 0. In the event that w_{ij} is 0, the expected value of U_{ij} should be β, which is the default value of the trust between any pair of participants. We *model* U_{ij} as Bernoulli random variable whose parameters are defined as logistic function of the weights w_{ij}. Therefore, we have:

$$U_{ij} = \begin{cases} 1 & \text{with probability } \frac{1}{1+e^{-\lambda_0-\lambda_1 w_{ij}}} \\ 0 & \text{with probability } \frac{1}{1+e^{\lambda_0+\lambda_1 w_{ij}}} \end{cases} \tag{6}$$

Here λ_0 and λ_1 are user-defined parameters. The value of λ_1 is always set to a nonnegative value. Therefore, increasing the weight w_{ij} also increases the trust probability $P(U_{ij} = 1)$. The value of λ_0 can be derived in terms of the default trustworthiness probability β. Note that when $w_{ij} = 0$, it is necessary for $P(U_{ij} = 1)$ to be β because a zero weight edge corresponds to a neutral opinion and it does not change the default belief of user i in user j. We can use this relationship to derive the value of λ_0 in terms of β. Therefore, we have:

$$P(U_{ij} = 1)_{w_{ij}=0} = \beta = \frac{1}{1+e^{-\lambda_0-\lambda_1(0)}} \tag{7}$$

By simplifying the aforementioned expression, we obtain:

$$\lambda_0 = \ln\left(\frac{\beta}{1-\beta}\right) \tag{8}$$

How can these probability values be used to model the steady-state probability of trustworthiness of an individual? Trust modeling is often used in fact-finding applications. An important point to keep in mind is that if an individual is a

troll, it is not necessary that the opposite of their stated facts are always correct. Rather, a lower value of the trust simply lowers the *probability* of their stated facts, included their stated trust in other individuals, to be correct. Therefore, it makes sense to model the reputation of each individual by using the probability of their trustworthiness. In other words, we want to create a model in which the feedback of each individual about other individuals (in terms of the sign and weight of links) is weighted by the probability of their trustworthiness. Just as the *PageRank* model simulates a random walk through the network, the *Troll-Trust* model simulates the following probabilistic process:

Sample a node j selected from the incoming neighbors of a node i, such that the probability of the node j being selected is proportional to its modeled trustworthiness probability $P(E_j = 1)$. Then the trustworthiness probability $P(E_i = 1)$ is modeled as the expected value of $P(U_{ji} = 1)$ over this sampling.

In other words, we use the sampled opinion of the neighbors in a node in order to model its trustworthiness. Furthermore, because the neighbors are weighted by their trust probabilities, this definition is recursive, just like the *PageRank* equations. Therefore, we have the following recursive relationship:

$$P(E_i = 1) = \frac{\sum_{j;(j,i)\in A} P(E_j = 1) \cdot P(U_{ji} = 1)}{\sum_{j;(j,i)\in A} P(E_j = 1)} \quad (9)$$

One can express this relationship directly in terms of the aforementioned variables $\pi(i)$, and the edge weights:

$$\pi(i) = \frac{\sum_{j;(j,i)\in A} \pi(j) \cdot \frac{1}{1+e^{-\lambda_0 - \lambda_1 w_{ji}}}}{\sum_{j:(j,i)\in A} \pi(j)} = \frac{\sum_{j;(j,i)\in A} \pi(j) \cdot \frac{1}{1+e^{-\ln[\beta/(1-\beta)] - \lambda_1 w_{ji}}}}{\sum_{j;(j,i)\in A} \pi(j)}$$

Of course, this definition does not yet account for the prior probability β, which is particularly useful in cases where the node i has no incoming nodes, which are also presumably trustworthy. The probability that none of the incoming nodes is trustworthy is given[2] by $\prod_{j:(j,i)\in A}(1 - \pi(j))$. In that case, the trustworthiness of node i is set to the default value β. Therefore, the aforementioned sampling needs to include an additional case in which the trustworthiness of the user i needs to be set to β when the case with probability $\prod_{j:(j,i)\in A}(1-\pi(j))$ occurs. This default scenario can also be viewed as a form of Laplacian smoothing, and it serves a similar goal to the restart step in the *PageRank* algorithm. Therefore, accounting for this default scenario, we need to add the term $\beta \prod_{j:(j,i)\in A}(1 - \pi(j))$ to the numerator, and the term $\prod_{j:(j,i)\in A}(1-\pi(j))$ to the denominator. Therefore, we obtain the following:

$$\pi(i) = \frac{\sum_{j;(j,i)\in A} \frac{\pi(j)}{1+e^{-\ln[\beta/(1-\beta)] - \lambda_1 w_{ji}}} + \beta \prod_{j:(j,i)\in A}(1 - \pi(j))}{\sum_{j;(j,i)\in A} \pi(j) + \prod_{j:(j,i)\in A}(1 - \pi(j))} \quad (10)$$

The aforementioned condition needs to hold for each node $i \in \{1 \ldots n\}$. Furthermore, when the node i has no incoming edges, it is easy to verify that the value of $\pi(i)$ will be set to β by the above equation. The values of β and λ_1 are two user-defined parameters in this algorithm. The selection of these parameters will be described later.

[2]We make the naive assumption that the trustworthiness of nodes is independent of one another.

3. ITERATIVE ALGORITHM

The condition for $\pi(i)$ in the previous section defines a *system* of equations. In this particular case, the system of equation is nonlinear. Unlike *PageRank*, such a nonlinear system of equations is hard to solve in closed form. Such non-linear systems of equations are often solved using iterative methods by starting with an initial set of default values and cycling through the system of equations and updating each value of $\pi(i)$.

Correspondingly, we use an iterative approach to update the probabilities of all the nodes. Let $\pi^t(i)$ be the value of the trust probability $\pi(i)$ of the ith node in the tth iteration.

1. Initialize iteration index $t \Leftarrow 0$.
2. Initialize $\pi^t(i) = \beta$ for each i.
3. For each i update from the probability values in the tth iteration to $(t+1)$th iteration as follows:

$$\pi^{t+1}(i) \Leftarrow \frac{\sum_{j\in I(i)} \frac{\pi^t(j)}{1+e^{-\ln[\beta/(1-\beta)] - \lambda_1 w_{ji}}} + \beta \prod_{j\in I(i)} (1 - \pi^t(j))}{\sum_{j\in I(i)} \pi^t(j) + \prod_{j\in I(i)} (1 - \pi^t(j))} \quad (11)$$

   , where $I(i) = \{j|(j,i) \in A\}$ is an index set.
4. Update $t \Leftarrow t + 1$
5. If converged then report the converged values $\pi^t(1) \ldots \pi^t(n)$ and terminate; else go to step 3

This approach is continued to convergence. In the next section, we will discuss the convergence behavior of this approach.

3.1 Convergence Behavior

We provide the convergence proof of the simplified *Troll-Trust* algorithm. First, we rewrite Eq. 9 in matrix form as follows:

$$\pi^{t+1} = ((A \odot P)^T \pi^t) \oslash (A^T \pi^t)$$

Here, A is the adjacency matrix, and P is a matrix with entries $P_{ij} = \frac{1}{1+\exp\{-\ln \beta/(1-\beta) - \lambda_1 w_{ij}\}}$. $\odot$ and $\oslash$ denotes element-wise multiplication and division respectively. Let us focus on the ith element $\pi^{t+1}(i)$ of the probability vector in the $(t+1)$th iteration,

$$\pi^{t+1}(i) = \frac{(A_i \odot P_i)^T \pi^t}{A_i^T \pi^t} = \frac{A_{ii} P_{ii} \pi^t(i) + B}{A_{ii} \pi^t(i) + D} \quad (12)$$

Here, A_i and P_i are the ith columns of A and P, respectively. We further denote $B = \sum_{j\neq i} A_{ij} P_{ij} \pi^t(j)$, and $D = \sum_{j\neq i} A_{ij} \pi^t(j)$. Since $A_{ii} \in \{0, 1\}$, $P_{ij} \in (0, 1)$, we immediately have $B < D$ and $\pi^{t+1}(i) < 1, \forall i$, and thus $D \in [0, n-1)$.

To prove convergence of Eq. 12, we use Banach Fixed Point theorem [2], which we introduce below.

DEFINITION 1. *Let (X, d) be a metric space. Then, a map $T : X \to X$ is called a contraction map on X if there exists $q \in [0, 1)$ such that*

$$d(T(x), T(y)) \leq qd(x, y), \forall\ x, y\ in\ X$$

THEOREM 1 (BANACH FIXED POINT THEOREM). *Consider a non-empty complete metric space (X, d) with a contraction mapping $T : X \to X$. Then, T admits a unique fixed-point x^* in X (i.e., $T(x^*) = x^*$).*

Furthermore, x^* can be found by using the following approach. We start with an arbitrary element x_0 in X and define a sequence x_n by $x_n = T(x_{n-1})$. Then it can be shown that $x_n \to x^*$.

THEOREM 2. *The* Troll-Trust *algorithm converges to a fixed point.*

PROOF. Let us map the problem within the terminology of the Banach fixed point theorem. Here, in our case $T(x) = \frac{A_{ii}P_{ii}x+B}{A_{ii}x+D}$, and we use the common Euclidean space, with L_2-norm as the associated distance function. The case where $A_{ii} = 0$ (q can be any value in $[0, 1)$) is trivial and we focus on the case where $A_{ii} = 1$, and now $T(x) = \frac{P_{ii}x+B}{x+D}, x \in [0, 1)$

$$T(x) - T(y) = \frac{P_{ii}D - B}{(x + D)(y + D)}(x - y)$$

According to the aforementioned theorem, the convergence proof only requires us to show that

$$q = \frac{|P_{ii}D - B|}{(x + D)(y + D)} < 1$$

. We consider three cases:

Case 0. $P_{ii}D - B = 0$

In this case $q = 0$, which satisfies the condition trivially.

Case 1. $P_{ii}D - B < 0$

$$q = \frac{B - P_{ii}D}{(x + D)(y + D)} < 1 \Leftrightarrow (x + D)(y + D) > B - P_{ii}D$$

$$\Leftarrow D^2 + (x + y + P_{ii} - 1)D + xy > 0\ [D > B] \quad (13)$$

The equation above contains the quadratic function $\varphi(D) = D^2 + (x + y + P_{ii} - 1)D + xy > 0$ with $D \in [0, n - 1)$. Notice when $x + y + P_{ii} - 1 > 0$, the minimum of φ in $[0, n - 1)$ is $\varphi(0) = xy > 0$, and we can choose $P_{ii} = \frac{1}{1+\exp\{-\ln\beta/(1-\beta)-\lambda_1 w_{ii}\}}$ to satisfy such a condition. Therefore, convergence is guaranteed.

Case 2. $P_{ii}D - B > 0$

When $P_{ii}D - B > 0$, similarly, we have

$$q < 1 \Leftrightarrow D^2 + (x + y - P_{ii})D + xy + B > 0 \quad (14)$$

We can now choose P_{ii} so that $(x + y - P_{ii}) > 0$, and thus $\varphi = D^2+(x+y-P_{ii})D+xy > \varphi(0) = xy > 0$. Since $B \geq 0$, Eq. 14 holds, we also have convergence in Case 2. □

3.2 Selecting the Parameters

An important aspect is the choice of the parameters β and λ_1. Much like *PageRank* this is a heuristic choice. However, one can choose the parameters by using the principle of self consistency with the network structure. The basic idea is to compute the trust values on a network in which around 90% of the links are retained, and the remaining 10% are held out. The computed trust values values are then used to perform sign prediction on the remaining 10% of the links. Various choices of β and λ_1 are tested on these links. After selecting these values of β and λ_1, the approach is then applied to the full data set. A more effective approach for selecting the parameters is to use cross-validation in which the data is divided into various folds and the parameter choices are decided by averaging the performance over various folds.

4. EXTENSIONS TO PERSONALIZED RANKING

In many scenarios, it is desired to determine the personalized ranking values for individuals. For a given user i, one may wish to determine the most similar users to i in the signed network. In other cases, a *subset* of nodes may be selected and it may be desirable to determine the most similar nodes to this subset. For generality, let us consider the case, where it is desired to personalize the ranking with respect to a subset S of nodes. There are two types of personalization which are possible:

- *Weighted personalization:* In this case, prior probabilities are provided for each node in S. These can also be viewed as weights specifying the importance of each node $i_r \in S$. For each node $i_r \in S$, its prior probability is specified as $\beta(i_r)$, as part of the input to the problem. For each node $i_r \notin S$, it is assumed that its prior probability is a small default value of β, which is less than that the prior probabilities of the nodes in S. This case is a very straightforward modification of the global *Troll-Trust* method. Specifically, the prior probabilities in Equation 10 are modified to replace the prior probabilities as follows:

$$\pi(i) = \frac{\sum_{j \in I(i)} \frac{\pi(j)}{1+e^{-\ln[\beta(i)/(1-\beta(i))]-\lambda_1 w_{ji}}} + \beta(i) \prod_{j \in I(i)} (1 - \pi(j))}{\sum_{j \in I(i)} \pi(j) + \prod_{j \in I(i)} (1 - \pi(j))}$$

  Note that $\beta(i)$ is set to a small default value for nodes which are not in S. The iterative update is based on the aforementioned relationship.

- *Hard personalization:* Unlike weighted personalization, it is assumed that the nodes in S are completely trusted, and therefore the values of *both* $\pi(i)$ and $\beta(i)$ are set to 1 for those nodes. Therefore, the Equation 10 is not applied to those nodes, which are in S. For nodes that are not in S, a small default value of β is assumed. For each node i, which is not in S, the following relationship is used to compute $\pi(i)$:

$$\pi(i) = \frac{\sum_{j \in I(i)} \frac{\pi(j)}{1+e^{-\ln[\beta/(1-\beta)]-\lambda_1 w_{ji}}} + \beta \prod_{j \in I(i)} (1 - \pi(j))}{\sum_{j \in I(i)} \pi(j) + \prod_{j \in I(i)} (1 - \pi(j))}$$

The main difference between weighted personalization and hard personalization is the level of trust placed in elements in set S. In the case of hard personalization, it is assumed that the elements of set S are completely trusted, irrespective of how many other users might have pointed to these nodes with negative links. On the other hand, in the case of weighted personalization, even though a prior bias is provided, the final trust probability of these nodes may be different from the prior values. Therefore, the weighted form

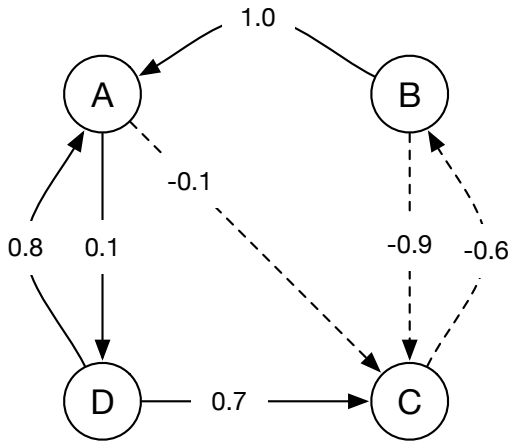

Figure 1: A four-node signed network with 4 positive links and 3 negative (dashed) links. Edge weights are labeled on each edge.

of personalization may be viewed as a softer way of biasing the process.

4.1 Interesting Special Cases and Applications

Personalized ranking has many applications in information search and recommender systems [22, 17]. The *Troll-Trust* model can be applied to various applications such as link prediction problem. It is well known that similar nodes are more likely to form connections [6]. This is the reason that personalized ranking methods were among the earliest methods used for unsigned link prediction [15]. This general principle can also be used in the case of signed networks. Using the same intuition as in the case of unsigned networks, we can make the assumption that an individual node in a signed network is more likely to initiate positive/negative edges to nodes with higher/lower personalized ranking scores with respect to that node. In general, since personalized ranking methods are used in a host of other network-centric applications, such as collective classification, it is conceivable that these methods can also be extended to those cases.

5. EXPERIMENTAL RESULTS

In this section, we will show the effectiveness of the *Troll-Trust* model with respect to baseline methods. We will first use an illustrative example to show that the ranking values obtained from the *Troll-Trust* model are semantically interpretable compared to other straightforward adaptations of ranking algorithms to the signed network scenario. We also apply the approach to an application-centric scenario to show that the underlying ranking approach is more robust. We use three different real datasets from the signed network domain, which are Wikipedia, Slashdot, and Epinions.

5.1 Comparative methods

To study the effectiveness of the *Troll-Trust* algorithm, we compare it to the following baselines. Each of these baselines can be applied to compute global ranking scores for all the nodes in the network:

- *Prestige* [24]

  Prestige is a simple algorithm that considers only positive and negative incoming links. It assumes that a node receiving more positive than negative incoming links is more likely to be trusted.

- *PageRank* [3]

  PageRank was originally designed for unsigned networks. Here, we apply *PageRank* on G^+, the subgraph of a signed network where all the negative links are removed, and positive links are preserved to obtain global trust values.

- *Exponential ranking*(*Exp*) [21]

  Exponential ranking was designed for ranking nodes in signed networks by heuristically using an exponential variation of the *PageRank* equations to deal with negative links.

- *Modified PageRank*(*MPR*) [18]

  Modified PageRank applies *PageRank* separately on both G^+, the positive subgraph, and G^-, the negative subgraph. Suppose the corresponding ranks for the node i are r_i^+ and r_i^- respectively. The overall reputation is computed as $r_i^+ - r_i^-$.

- *PageTrust* [9]

  PageTrust is a modified version of *PageRank*, which multiplies the right-hand side of the *PageRank* equation with a heuristic correction factor in an effort to account for negative links.

- *Bias and Prestige*(BAP) [16]

  Bias and Prestige models both bias, which shows the expected weight of an outgoing edge of a node, and prestige, which reflects the expected weight of an incoming edge of a node, recursively in signed networks. The assumption is that the opinions of biased users, even with high prestige, should not be discounted.

- *HITS* [10]

  HITS was originally proposed to analyze the link structure in the *World Wide Web* (*WWW*). To apply *HITS* in signed networks, *HITS* is run separately on both G^+ and G^-. Similar to *Modified PageRank*, the overall authority value is computed as $a_i^+ - a_i^-$, where a_i^+ and a_i^- denote the corresponding authority values for the node i respectively.

5.2 Synthetic Case Study

Before providing the results of ranking algorithms in application-centric scenarios, we provide a synthetic case study on a toy data set, to provide an intuitive understanding of *why* the results from a *Troll-Trust* model may make semantic sense over competing methods. This also explains the later results on why its use in application settings provides more accurate results.

As shown in Figure 1, a signed network is constructed with 4 nodes, 4 positive links, and 3 negative links. The trust/distrust weights on the edges are also indicated. It is intuitively evident that node A is the most trustworthy, given the positive incoming links from other nodes and no negative incoming link. The trustworthiness of node C is more controversial, because it is distrusted by node A and B but partially trusted by node D. We run the *Troll-Trust*

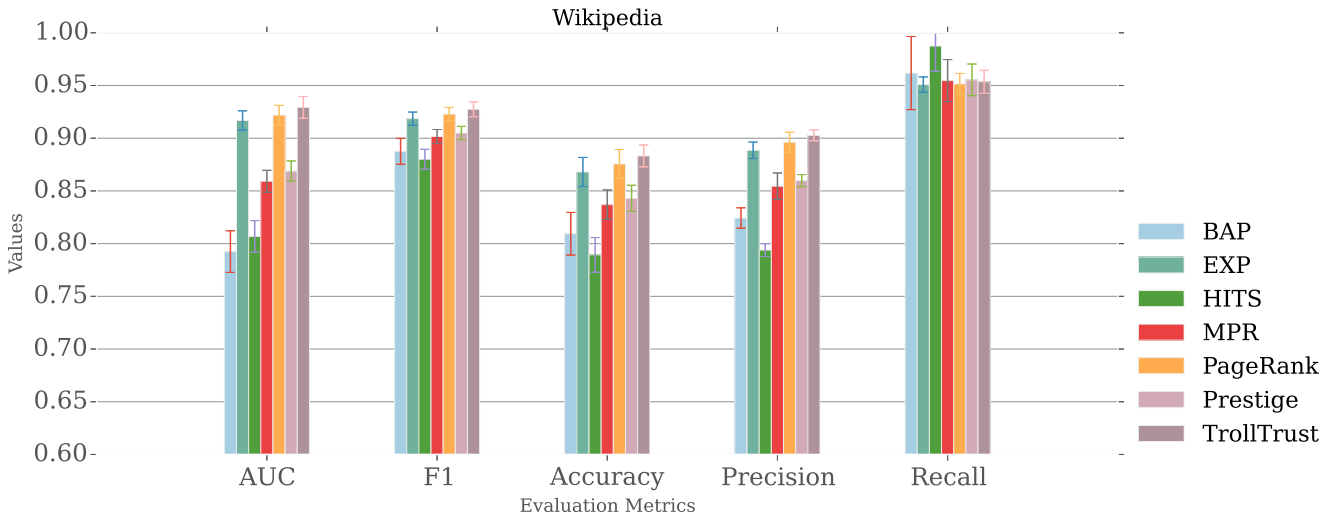

Figure 2: Predictive performance of *Rep* (Eq 15) and *Opt* (Eq 16) based on different ranking algorithms, including *Troll-Trust*, *Exp*, *BAP*, *PageRank*, *Prestige*, *MPR* and *HITS*. Various evaluation measures, such as the *AUC*, *F1 Score*, *Accuracy*, *Precision* and *Recall*, are shown on the Wikipedia network.

Table 1: Global trust values from different models on the signed network in Figure 1

Model	A	B	C	D
Prestige	1.0	0.5	0.3333	1.0
PageRank	0.2025	0.1650	0.5001	0.1324
Exp	0.7074	0.5226	0.4591	0.5168
MPR	0.5687	0.4182	0.5719	0.4412
PageTrust	0.1723	0.1220	0.2154	0.1204
BAP	0.9201	0.4965	0.4568	0.5496
HITS	0.7366	0.4348	0.2473	0.7508
Troll-Trust	0.6708	0.5094	0.4790	0.5168

algorithm together with 8 other baselines on this signed network. The obtained trust values are shown in Table 1. All results are linearly scaled into $[0, 1]$.

It can be observed from Table 1 that some of the baseline algorithms yield poorly interpretable results. *Prestige* is too focused on the number of positive and negative incoming links without taking into account the network structure. *PageRank* on the positive subgraph doesn't take into account negative links at all, therefore node C has the the highest trust value, which is not desirable given the negative links from node A and node B. *PageTrust* tries to incorporate negative links into the *PageRank* model but node C still receives the highest trust value in the modified model. *MPR* and *HITS*, respectively, apply the original *PageRank* and *HITS* separately on the positive and negative subgraph and obtain the difference as the global trust value. However, this type of decomposed approach fails to take the interaction among the two types of links into account.

The results of *BAP*, *Exp* and *Troll-Trust* are more in accordance with intuitive expectations. Although the results of *Exp* and *Troll-Trust* are very similar in this case, further experiments on real datasets will show their difference in prediction tasks.

We should notice that each trust score in the results of the *Troll-Trust* model represents the probability of a node being trustworthy. On the other hand, other baselines, although whose results are able to be scaled into $[0, 1]$, actually lack a probabilistic explanation in their models. Therefore their results are only an intuitive analogy to the trust values from the hidden probabilistic model.

5.3 Data Set Descriptions

We test our proposed work on three real signed network datasets, Epinions, Slashdot and Wikipedia, where each edge is labeled explicitly with either a positive $(+1)$ or a negative (-1) sign. All three datasets are available online.

Table 2: Dataset statistics

Dataset	Node	Edge	+Edge(%)	−Edge(%)
Epinions	131,828	841,372	85.0	15.0
Slashdot	82,144	549,202	77.4	22.6
Wikipedia	10,835	159,388	78.7	21.2

- *Epinions:* Epinions is an online review site, in which users are connected by directed links with positive $(+1)$ or negative (-1) signs. Epinions can be viewed as a directed graph with $131,828$ nodes and $841,372$ edges, of which 85% are labeled positive.
- *Slashdot:* Slashdot is a technology blog, where the users can tag each other as 'friends' or 'foes'. The 'Friends' tag is indicative of a positive link, whereas the 'foes' tag is indicative of a negative (-1) link. Therefore, Slashdot can represent a signed network with $82,144$ nodes and $549,202$ edges, of which 77.4% are labeled positive.
- *Wikipedia:* In Wikipedia, users are able to cast supporting or opposing votes for other admin users. Votes are extracted as signed links $(+1$ for positive votes and -1 for negative votes). This dataset contains $10,835$ nodes and $159,388$ edges, of which 78.7% are labeled positive.

In all these networks, the proportion of positive links is around 80%. Topological details including the number of nodes, the number of edges, and the proportion of positive

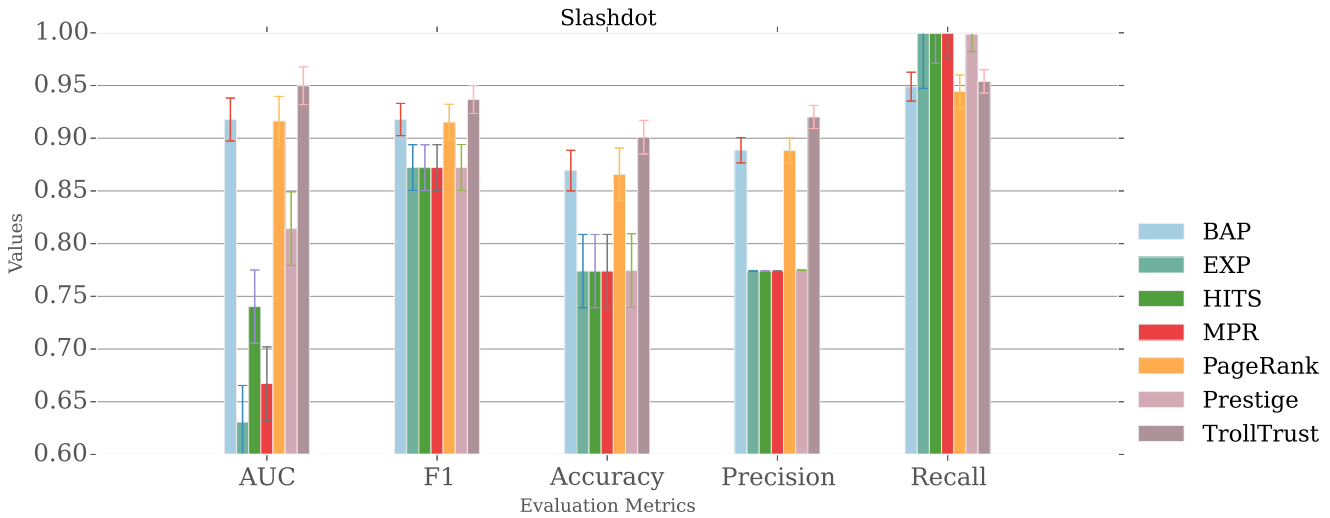

Figure 3: Predictive performance for Slashdot dataset. Other designations are the same as Fig 2

and negative links of all three datasets are summarized in Table 2.

5.4 Testing Methodology

How can we show that one ranking method is better than another? After all, there is no ground-truth availability for the ranks of the nodes in any of the data sets. Therefore, we use an indirect approach, which conforms to the methodology generally used in the literature [18, 21]. The key idea is that the node rankings are often used as a component of various applications, whose effectiveness can be measured more easily. For example, the rank (or trust) of a node is an integral input as a feature to many sign prediction methods. The quality of the latter can be measured easily against a concrete ground-truth. We emphasize that our goal is not to provide new sign prediction methods, but to test the effectiveness of incorporated ranking methods with this indirect approach. Therefore, we will use sign prediction methods to test our approach.

5.4.1 *Description of Sign prediction Task*

Consider a signed network in which the signs on some edges are not recorded. The sign prediction problem is to predict those hidden signs based on the information obtained from the rest of the network, such as the structure of the network and the signs of other links [13]. To accomplish the prediction task [18], the nodes are ranked in terms of global trust values, which are further used to calculate the *reputation* (*Rep*) and *optimism* (*Opt*) scores. With *Rep* and *Opt* used as features, logistic regression is used as the predictive model. The accuracy is evaluated with the use of measures such as the *Accuracy*, *F1-Score*, *AUC*, *Precision* and *Recall*.

Consider a signed network in which $s(i,j) = \pm 1$ denotes the sign of the edge from from node i to node j. If no edge exists from i to j, then the sign is 0. In the sign prediction problem, we predict the signs of edges for which the sign is unknown.

We adopt the two measures in [18], denoted by *reputation* and *optimism*, as features for sign prediction tasks. Both these features are expressed in terms of ranking scores, and therefore a good performance by the sign prediction task with these features is indicative of a ranking of good quality. The *reputation* feature quantifies the popularity of a node in the network, whereas *optimism* quantifies the pattern of votes a node make in the network. Let $\pi(i)$ denote the global trust score we obtain for node i. Then, the *reputation* (Rep_i) and the *optimism* (Opt_i) features of node i are expressed in terms of $\pi(i)$ (or any other baseline ranking mechanism) as follows:

$$Rep_i = \frac{\sum_{j;(j,i)\in A^+} \pi_j - \sum_{j;(j,i)\in A^-} \pi_j}{\sum_{j;(j,i)\in A^+} \pi_j + \sum_{j;(j,i)\in A^-} \pi_j} \tag{15}$$

$$Opt_i = \frac{\sum_{j;(i,j)\in A^+} \pi_j - \sum_{j;(i,j)\in A^-} \pi_j}{\sum_{j;(i,j)\in A^+} \pi_j + \sum_{j;(i,j)\in A^-} \pi_j} \tag{16}$$

Here, $A+$ and $A-$ denote the adjacency matrices for the positive and negative subgraphs, respectively. For any given edge, these two features are extracted from its endpoints in order to perform the learning process.

5.5 Evaluation metrics

Next, we briefly introduce our evaluation metrics. In the succeeding discussion, we denote y as the ground truth and $\hat{y}$ as our prediction.

- *Accuracy*

  The accuracy measures the proportion of the successfully predicted instances, which can be biased in the case of unbalanced datasets. Therefore we also use metrics discussed below.

- *Precision and Recall*

  Sign prediction is a binary classification task, and thus we have the following possible outcomes as in Table 3

Table 3: Possible Outcomes in Link Prediction Tasks

Total population	+ Link	− Link
Predicted + Link	True Positive(tp)	False Positive(fp)
Predicted − Link	False Negative(fn)	True Negative(tn)

Based on counts in each categories, precision and recall

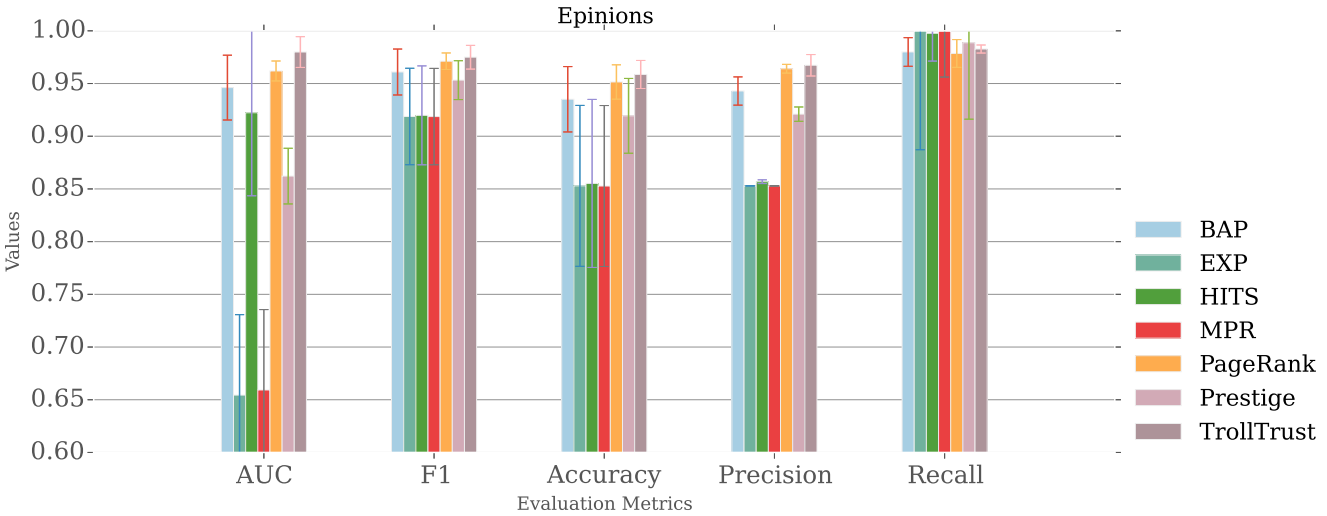

Figure 4: Predictive performance for Epinions dataset. Other designations are the same as Fig 2

is defined as follows:

$$precision = \frac{tp}{tp + fp}$$

$$recall = \frac{tp}{tp + fn}$$

Typically, there is a trade-off between precision and recall.

- *F1-Score*

  To account for the trade-off between precision and recall, we also use F1-Score, the harmonic mean of precision and recall

  $$F1 = 2\,\frac{precision \times recall}{precision + recall}$$

  F1-Score falls in the range of $[0, 1]$ and higher value indicates higher predictive power.

- *Area Under ROC Curve (AUC)*

  Another measure accounts for such trade-off is the area under receiver operating characteristics (ROC) curve, which measures the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one; a higher AUC would indicate a better predictive performance.

5.6 Experimental setup

For each data set, 10-fold cross-validation is performed to evaluate the predictive model. In each round of cross-validation, for any node i in the signed network G from the dataset, we first run the ranking algorithms to obtain the global trust value $\pi(i)$. Then for any edge $e(u, v)$, four features, Rep_u, Opt_u, Rep_v, Opt_v, are extracted based on $\pi(u)$ and $\pi(v)$ to construct a feature vector. We use logistic regression with $L1$ regularization as the classifier and evaluate the predictive performance by *Accuracy*, *F1-Score*, *AUC*, *Precision* and *Recall*. For clarity, here we use the names of different ranking algorithms to denote different classifiers respectively.

In our *Troll-Trust* model, the parameters are selected via cross-validation on the training data, with the aim of achieving the highest *Accuracy*. As generally assumed, the damping factor in *PageRank* is set to 0.85 and both the initial *Authority* and *Hub* values in *HITS* are set to 1. To guarantee convergence, the μ in *Exp* is set to 1 [21]. In the *PageTrust* algorithm, multiplication of large dense matrices is inevitable, and therefore this algorithm is excluded in our experiments due to the memory limit.

All experiments were conducted on machines with Intel(R) Xeon(R), CPUs @ 2.60GHz and 125GB RAM.

5.7 Results

The predictive performances of the sign prediction methods, based on features derived from different ranking algorithms on Wikipedia, Slashdot and Epinions, are shown in Fig 2, Fig 3 and Fig 4 respectively.

We can observe that on all three datasets our *Troll-Trust* model significantly outperforms other baselines in terms of comprehensive evaluation metrics *AUC*, *F1 Score* and *Accuracy*. High *Recall* scores achieved by baselines indicate that they can be overly aggressive in the prediction task, which yields extremely poor *Precision* scores and mediocre overall performance.

With the *Troll-Trust* model giving the best results, the pattern of the performances of other algorithms seems to vary on different datasets. Surprisingly, we found the unsigned *PageRank* to be reasonably stable on all the datasets, even though it was not the best performer. There can be multiple reasons behind it. First of all, although the information about negative links is completely ignored in *PageRank* algorithm, it is taken into account into the computation of the *Rep* and *Opt* during the feature extraction process. Furthermore, all three datasets are highly imbalanced with about 80% positive links. As a result, the the *PageRank* model is able to perform well on at least the positive parts of the network. Nevertheless, it is still outperformed by the *Troll-Trust* model.

Another model that is relatively stable is *Prestige*, while the results are not desirable, which is expected since it does not differentiate the trustworthiness of the incoming neighbors. The performance of other algorithms including *Exp*, *BAP*, *MPR* and *HITS* fluctuates according to the datasets. The heuristic modifications in these model can account for such instability. Although *Exp* performed well in the case

study presented in Section 5.2, it does not perform as well in the sign prediction tasks.

6. CONCLUSIONS AND SUMMARY

This paper presents a *Troll-Trust* model for ranking in signed networks. One of the interesting aspects of the model is that there is a clear semantic interpretation of the ranking values in terms of the trust probabilities. Such a semantic interpretation provides a model which is more clearly rooted in the dynamics of such signed networks. This is also reflected in the fact that our model provides predictions of high quality. We also develop a personalized version of the ranks, which can be used for other applications such as link prediction. Our experimental results show that the *Troll-Trust* model provides more accurate results than competing methods for the sign prediction tasks.

7. ACKNOWLEDGEMENT

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

8. REFERENCES

[1] C. Aggarwal Social network data analytics. *Springer*, 2011.

[2] S. Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fund. Math*, 3(1):133–181, 1922.

[3] S. Brin, and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1), pp. 107–117, 1998.

[4] K. Y. Chiang, C. J. Hsieh, N. Natarajan, I. S., Dhillon, and A. Tewari. Prediction and clustering in signed networks: a local to global perspective. *The Journal of Machine Learning Research*, 15(1), pp. 1177–1213, 2014.

[5] Ramanthan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, pages 403–412. ACM, 2004.

[6] W. Cukierski, B. Hamner, and B. Yang. Graph-based features for supervised link prediction. In *International Joint Conference on Neural Networks (IJCNN)*, pp. 1237–1244, 2011.

[7] M. Jamali and M. Ester. TrustWalker: A random-walk model for combining trust-based and item-based recommendation. *ACM KDD Conference*, pp. 397–406, 2009.

[8] S. Kamvar, M. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. *World Wide Web Conference*, pp. 640–651, 2003.

[9] C. de Kerchove and P. V. Dooren. The PageTrust algorithm: how to rank web pages when negative links are allowed? *SIAM Conference on Data Mining*, pp. 346–352, 2008.

[10] J. M. Kleinberg, Authoritative Sources in a Hyperlinked Environment, *Journal of the ACM*, 46(5), pp. 604–632, 1999.

[11] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. *SIAM Conference on Data Mining*, 2010.

[12] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: mining a social network with negative edges. *World Wide Web Conference*, pp. 741–750, 2009.

[13] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. *World Wide Web Conference*, pp. 641–650, 2010.

[14] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. *SIGCHI Conference on Human Factors in Computing Systems*, pp. 1361–1370, 2010.

[15] 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), pp. 1019–1031, 2007.

[16] A. Mishra and A. Bhattacharya. Finding the bias and prestige of nodes in networks based on trust scores. *Proceedings of the 20th international conference on World Wide Web*, pp. 567–576, 2011.

[17] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Uncertainty in Artificial Intelligence (UAI)*, pp. 452–461, 2009.

[18] M. Shahriari and M. Jalili. Ranking nodes in signed social networks, *Social Network Analysis and Mining*, 4:172, 2014.

[19] P. Symeonidis, E. Tiakas, and Y. Manolopoulos. Transitive node similarity for link prediction in social networks with positive and negative links. *ACM Conference on Recommender Systems*, pp. 183–190, 2010.

[20] J. Tang, S. Chang, C. Aggarwal, and H. Liu. Negative link prediction in social media, *WSDM Conference*, 2015.

[21] V. Traag, Y. Nesterov, P. van Dooren. Exponential Ranking: taking into account negative links. *Social Informatics*, 6430, pp. 192–202, 2010.

[22] H. Wang, X. He, M.-W. Chang, Y. Song, R. W. White, and W. Chu. Personalized ranking model adaptation for web search. In *ACM SIGIR Conference*, pp. 323–332, 2013.

[23] T. Zhang, H. Jiang, Z. Bao, and Y. Zhang. Characterization and edge sign prediction in signed networks. *Journal of Industrial and Intelligent Information* Vol, 1(1), 2013.

[24] K. Zolfaghar and A. Aghaie. Mining trust and distrust relationships in social web applications. In *Intelligent Computer Communication and Processing Conference(ICCP)*, pp. 73–80, 2010.