What Is the Added Value of Negative Links in Online Social Networks?

Jérôme Kunegis
University of Koblenz–Landau
Universitätsstr. 1
56070 Koblenz, Germany
kunegis@uni-koblenz.de

Julia Preusse
University of Koblenz–Landau
Universitätsstr. 1
56070 Koblenz, Germany
jpreusse@uni-koblenz.de

Felix Schwagereit
University of Koblenz–Landau
Universitätsstr. 1
56070 Koblenz, Germany
schwagereit@uni-koblenz.de

ABSTRACT

We investigate the "negative link" feature of social networks that allows users to tag other users as *foes* or as *distrusted* in addition to the usual *friend* and *trusted* links. To answer the question whether negative links have an added value for an online social network, we investigate the machine learning problem of predicting the negative links of such a network using only the positive links as a basis, with the idea that if this problem can be solved with high accuracy, then the "negative link" feature is redundant. In doing so, we also present a general methodology for assessing the added value of any new link type in online social networks. Our evaluation is performed on two social networks that allow negative links: The technology news website Slashdot and the product review site Epinions. In experiments with these two datasets, we come to the conclusion that a combination of centrality-based and proximity-based link prediction functions can be used to predict the negative edges in the networks we analyse. We explain this result by an application of the models of preferential attachment and balance theory to our learning problem, and show that the "negative link" feature has a small but measurable added value for these social networks.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*; H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Platform policies; link prediction; signed network; enmity; distrust

1. INTRODUCTION

Online social networks allow people to connect with each other, forming a network. In most online social networks, only positive links between people are allowed such as *friendship*, *trust* and the *following* relationship. Relationships between people however may also be of a negative type, for instance enmity as opposed to friendship, and distrust as opposed to trust. A very small number of online social networks actually do allow such negative links. Among them is Slashdot, a technology news website that lets its users tag other users as *friends* and *foes*, as well as the product review site Epinions that allows users to *trust* and *distrust* each other. In both cases, the negative link feature results in directed signed links between users that can be interpreted as approval and disapproval links, and that are used in the user interface of the two websites to decide which content is shown to users.

On Slashdot, the posts of users tagged as foes are given a lower score, and may thus be hidden. On Epinions, the trust and distrust information is used to determine the reviews shown, using an undisclosed algorithm. The negative links are thus used on both sites to enhance the site's content, and a *negative link* feature could similarly enhance he content shown on many websites. Since many online social networks are however reluctant to implement a negative link feature, as shown by the very small number of sites featuring them, the question arises whether negative links have an added value for the network or whether their purpose can be replaced by a prediction algorithm that determines the negative social links automatically from the known, positive links. Such an algorithm could be applied to any online social network that does not want to allow explicit negative links, and would increase the accuracy of news streams, content filters and recommender systems embedded in these online social networking sites.

Based on these premises, this paper investigates the following research question: Can the negative links allowed in Slashdot and Epinions be inferred from the positive links only? To answer this question, we make the following contributions:

- We introduce a general methodology for how to evaluate the redundancy of additional link types in online social networks in addition to the regular positive links, under the assumption that a link type is redundant if it can be inferred by links of another type.
- We define and evaluate the machine learning problem of learning negative from positive links in a signed network. This problem is related to the link prediction problem consisting of predicting future edges in an unweighted network.
- In the two social networks with negative links Slashdot and Epinions, we study to what accuracy negative links can be inferred by positive links.
- We propose a function that has high accuracy at the

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

described link prediction problem, based on the observation that link prediction functions can be grouped into centrality-based and proximity-based functions, depending on their behavior for the negative link prediction task using the data of Slashdot and Epinions.

The paper is structured as follows. Section 2 describes the problem settings studied in this paper, in Section 3 we perform an initial analysis of standard link prediction functions at the task of predicting negative links. Then, Section 4 introduces our methodology, Section 5 gives the results of our experiments using Slashdot and Epinions data, and Section 6 concludes the paper.

2. NEGATIVE LINKS IN SOCIAL NETWORKS

Social networks provide their users with a variety of functionality for connecting with other users. Examples of these features are friends on Facebook, circles on Google Plus and followers on Twitter. Explicitly created connections in social networks can be displayed in the user profile and some users might want to boost their status by collecting as many visible contacts as possible. Besides consequences for the status of these users, these explicit social connections deeply influence the user experience within the social networking platform and the ability to interact with other users. The nature of an explicit link between two users is therefore dependent on its platform-specific implementation.

In this paper, we limit our investigation to links between users that are intended to be permanent and therefore describe a long-lasting connection. This excludes links between users and other entities that form bipartite networks, e.g., ratings of movies, articles, comments, etc. Ratings of persons in dating sites [8] fall in this category too, since the rating and rated users have different roles. The same holds for one-time events such as elections, e.g., the elections of administrators in Wikipedia [12].

Permanent social links between two users can be divided into two types according to their functionality, that can be described as positive and negative. It can be observed that large social networks such as Facebook and Google Plus provide positively connotated linking functionality called *friend*, *contact*, or multiple *circles* with user-defined labels. These links are the defining concept for social networks, and they are crucial for them since they determine the visibility of user-generated content for the creator and for potential readers. It is this functionality that makes the platform social since the user is supported in his interaction with selected other users. In the following, we define links that increase the visibility of users and content or which increase the ability to interact as positive links. Consequently, the links that decrease visibility of content or which decrease the ability to interact are called negative links. Negative links are associated with disapproval for another user. Labels for explicit negative links in social networks are for instance *enemy*, *foe*, *distrust*, *ignore*, *hide* and *block*. As negative aspects of a community are rarely advertised, these negative links are much less used and known than positive links. This might be one reason why only few social networks with negative links are publicly available for study and research. However, two available social networks that contain positive and negative links are Slashdot and Epinions.

Table 1: The two signed social network datasets used in our evaluation. In both networks, all edges are directed.

Dataset	Vertices	Edges (pos. + neg.)
Slashdot Zoo [9]	79,120	515,581 (392,326 + 123,255)
Epinions trust [17]	131,828	841,372 (717,667 + 123,705)

Slashdot[1] is a technology news platform where users can post and read other users' news articles and comments [9]. On Slashdot, users can create two types of explicit and directed social links between themselves and other users. These are labeled *friend* and *foe*. Both link types allow the user to change the visibility of the content the linked user has created. Although the effect of a link is not predetermined but user configurable the convention is that the *friend* link increases the content visibility, the *foe* link decreases content visibility of the target user. Therefore the *friend* link is a positive link, while the *foe* link is a negative link. The *friend* and *foe* link types are also called *fan* and *freak* from the point of view of the targeted user. The signed social network of Slashdot is called the *Slashdot Zoo* on Slashdot itself, and can be considered an extension to Slashdot's sophisticated moderation system [10].

Epinions[2] is a site that collects community-created product reviews [17]. Two types of links can be created by one user to a target user. One link is labeled *trust* the other link is labeled *block* (or formerly, *distrust*). These links influence the visibility of product reviews that are authored by the target user. The user who has created the *trust* link sees the reviews of the trusted user at a higher position in the list of all relevant reviews. Therefore this link is considered to be a positive link. Reviews by users that are *blocked* are not presented to the user, making it a negative link. The positive and negative links on Epinions are also used to predict a global trust score for individual users.

Table 1 summarizes the two datasets. Both datasets are available on the authors' website[3][4]. Both networks have both positive and negative links between users, forming a directed, asymmetric signed network. Although the functionality that lies behind the link types is not fully identical between Slashdot and Epinions, it is very similar according to our definition of positive and negative links. Based on this similar functionality we assume similar properties of the two networks, and will use both datasets for experiments in the rest of the paper.

3. PREFERENTIAL ATTACHMENT AND BALANCE THEORY

A major model of network analysis is preferential attachment, i.e., the rule that new edges are more likely to be attached to nodes with high degree [2]. Another important model is that of a high clustering coefficient, i.e., the rule that typical networks contain a much higher number of triangles than predicted by a random graph model, and thus edges tend to connect nodes that have a high number of common neighbors [20]. A high clustering coefficient is one

[1]slashdot.org
[2]epinions.com
[3]konect.uni-koblenz.de/networks/slashdot-zoo
[4]konect.uni-koblenz.de/networks/epinions

component of the *small-world* network model, and can be generalized to signed graphs to give balance theory, stating that triangles are likely to be balanced, i.e., to contain an even number of negative edges [4].

The preferential attachment model can be used to derive link prediction functions based on centrality measures, such as the degree of nodes and PageRank, whereas the clustering model leads to link prediction functions that compare the immediate proximity of two nodes, such as the number of common neighbors and the cosine similarity. In the case where negative edges are allowed in a network, the problem of predicting the sign of new edges, given the known positive and negative edges is called the link sign prediction problem, and has been extensively studied [9, 13, 14]. In the link sign prediction problem, the known network contains both positive and negative edges, and thus sign information can be used for prediction. For instance, the multiplication rule *the enemy of my enemy is my friend* can be used [9], a comparison of the number of adjacent edges of any positive/negative pattern [13], or a diffusion process [3]. These types of methods can however not be applied in the problem studied in this paper, since in our case only positive edges are known.

A related problem is that of predicting the sign of new links, given both positive and negative links in a network [21]. In addition to the network itself, the method described in that work uses interaction information to achieve its prediction, as well as a small sample of signed edges. Thus, the method cannot be applied to our scenario, since we assume no negative links are possible in the network.

Let $G = (V, E, w)$ be a social network (Slashdot or Epinions) with V the set of users, E the set of directed links between users, and $w : E \to \pm 1$ the edge sign function, with $w((i, j)) = +1$ denoting that user i approves of user j and $w((i, j)) = -1$ denoting that user i disapproves of user j. The fact that two nodes $i, j \in V$ are connected (in either direction) will be denoted by $i \sim j$, and the fact that i and j are connected by a directed edge (i, j) by $i \to j$. The degree of vertex $i \in V$, i.e., the number of vertices connected to i (in either direction) will be written as $d(i)$. The outdegree of node i, i.e., the number of nodes pointed to by i is denoted as $d_{\mathrm{O}}(i)$.

At the task of ordinary link prediction, in which future positive links must be predicted from current positive links, both centrality and proximity measures are used. A link prediction function is defined to take as input a node pair (i, j), and returns a numerical score indicating how likely a new edge is to appear between i and j. The returned scores need not be probabilities, and are higher when the probability of edge formation is higher. Examples of link prediction functions are the PageRank product and the cosine similarity.

PageRank [19] is a centrality measure in a directed network defined as the solution $\mathrm{PR}(i), i \in V$ of

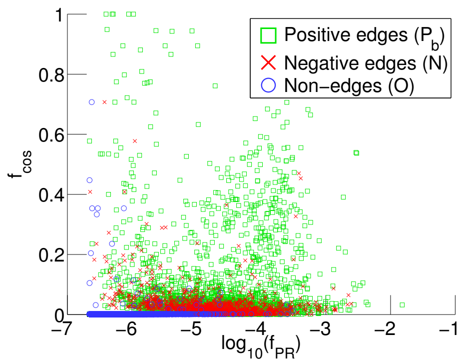
$$\mathrm{PR}(i) = \frac{1-\alpha}{n} + \alpha \sum_{j \to i} \frac{\mathrm{PR}(j)}{d_{\mathrm{O}}(j)}$$

where α is a parameter set to 0.85 [11]. The PageRank values are all positive by construction. The PageRank product link prediction function is then defined as the product of the two nodes' PageRanks.

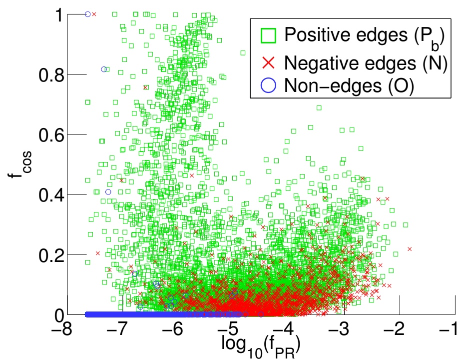
$$f_{\mathrm{PR}}(i, j) = \mathrm{PR}(i)\mathrm{PR}(j) \tag{1}$$

The cosine similarity is defined as the cosine between the two adjacency vectors of i and j, where the adjacency vector of a vertex is the 0/1 vertex-vector indicating to which vertices a given vertex is connected. The cosine similarity can be expressed in the following manner:

$$f_{\cos}(i, j) = \frac{|\{k \mid i \sim k \wedge k \sim j\}|}{\sqrt{d(i)d(j)}} \tag{2}$$



(a) Slashdot



(b) Epinions

Figure 1: Scatter plots of the cosine similarity and the PageRank product with points colored according to their inclusion in the set of unknown positive edges P_{b}, the set of unknown negative edges N and the set of non-edges O. The plots are best viewed in color.

Let P be the set of positive edges and N the set of negative edges, i.e., $E = P \cup N$. To perform an initial analysis of the datasets, we split[5] the set of positive edges P randomly into two sets P_{a} and P_{b} such that $|P_{\mathrm{a}}| = 3|P_{\mathrm{b}}|$. We then consider P_{a} the set of known edges (all positive), P_{b} the set of unknown positive edges, N the set of negative edges to predict, and finally a randomly sampled set O of node pairs not in E with size $|O| = |P_{\mathrm{b}}|$.

We can now compute the PageRank product and the cosine similarity for all node pairs in the sets N, P_{b} and O, based on the known edges P_{a}. Figure 1 shows the scatter plot of the nodes pairs of the three unknown sets plotted in function of their PageRank product and cosine similarity values.

Two observations can be made:

[5]The factor 3 is motivated by the fact that in the link prediction literature, the standard size of the test set is 25% of the total set of edges.

- Most node pairs in the non-edge set O have a cosine similarity of zero, and a small value of the PageRank product.
- Node pairs in the positive edge set P_{b} have high cosine similarity and high PageRank product values (compared to non-edges).
- Node pairs in the negative edge set N have low but mostly nonzero cosine similarity values and high PageRank product values (compared to non-edges).

These observations are true for both the Slashdot and Epinions datasets. We can conclude that negative edges can be identified from the cosine similarity and PageRank product in the following way:

- Negative edges connect nodes with high PageRank product values.
- Negative edges connect nodes with low but nonzero cosine similarity values.

Thus, we expect a combination of a positively weighted centrality measure with a negated proximity measure to solve our problem of predicting negative links, giving a combined prediction measure that takes into account both preferential attachment and balance theory.

4. METHODOLOGY

In this section, we describe our method for testing whether the negative links of a signed social network can be predicted from its positive links. This is a variant of the link prediction problem in which two link types exist. We will review the link prediction problem itself, give suitable link prediction functions adapted to the problem at hand, and will describe two experiments, one for measuring the achievable accuracy of the prediction problem, and one for computing an upper bound on that accuracy.

As defined in the previous section, the set of edges E can be divided into the set of positive edges P and the set of negative edges N. The problem can then be rephrased as the problem of evaluating whether the negative links N can be predicted from the positive links P. The general methodology we introduce for this kind of problem consists in predicting links of one type using only links of another type in the network. This problem extends the ordinary link prediction problem in which only a single link type is present.

4.1 The Link Prediction Problem

Given the set of links E_{a} present at a particular time, how can the new links in the set E_{b} that emerge later be predicted accurately? This problem is called the link prediction problem [15]. Typically, the link prediction problem is solved by *link prediction functions*, i.e., functions that map node pairs to numerical scores, based on the known edges in the set E_{a}. Examples of link prediction functions are the number of common neighbors and the product of node degrees.

To compare the prediction accuracy of multiple link prediction functions, the set of links in a network is split into two sets: the links in the training set E_{training} that are assumed to be known, and the links in the true test set E_{true} set that must be predicted. Additionally, a set of node pairs is randomly generated to be the false test set, denoted by E_{false}, consisting of links not belonging to any of E_{training} or E_{true}. We formalize a link prediction problem $\mathcal{P}$ as

$$\mathcal{P} : E_{\mathrm{training}} \to E_{\mathrm{true}} \mid E_{\mathrm{false}}, \tag{3}$$

which denotes that links from the training set E_{training} are used to distinguish links from the true test set E_{true} against those from the false test set E_{false}. This notation is novel; we choose to use it due to the variety of the encountered link prediction problems in our problem setting.

Using this notation, we are able to formulate the general link prediction problem of using edges of a given type X to predict edges of another type Y:

$$E_{\mathrm{X}} \to E_{\mathrm{Y}} \mid O,$$

where O is a randomly sampled set of node pairs disjoint from E_{X} and E_{Y} of the same size as E_{Y}. If X and Y are two link types supported by a social networking site, this problem can be used to predict whether the link type Y is necessary, or whether it can be predicted from the edges of type X.

To solve a link prediction problem, one uses link prediction functions. A link prediction function is a function

$$f : V \times V \to \mathbb{R}$$

that, when applied to the training set, gives edges in the true test set higher values than edges in the false test set. The result of a link prediction function will be called the link prediction score. Multiple link prediction functions can then be compared to find a function that solves the link prediction problem to a satisfying accuracy, for instance using the area under the curve as described later.

4.2 Link Prediction Functions

Link prediction functions can be divided into proximity-based and centrality-based functions, based on whether they include only vertex-based features of vertex-pair-based features. In the following, we list the link prediction functions used in our experiments, which correspond to the most common general link prediction functions used in the literature, and can be found for instance in [15] and [16]. The two nodes for which a link prediction score is to be computed will be called i and j.

4.2.1 Proximity-based Functions

These link prediction functions are based on comparing the neighboring nodes of i and j. In addition to the cosine similarity defined in Equation (2), we use the following proximity-based link prediction functions.

The number of paths of length two between i and j is defined as

$$f_{\mathrm{P2}}(i, j) = |\{k \mid i \sim k \wedge k \sim j\}|, \tag{4}$$

which equals the number of nodes that are adjacent to both i and j, i.e., the number of common neighbors.

Analogously, the number of paths of length three between i and j is defined as

$$f_{\mathrm{P3}}(i, j) = |\{(k, l) \mid i \sim k \wedge k \sim l \wedge l \sim j\}|, \tag{5}$$

where the sequence (i, k, l, j) forms a path of length three from node i to node j.

The Jaccard coefficient measures the amount of common neighbors divided by the number of neighbors of either vertex [15]:

$$f_{\mathrm{Jac}}(i,j) = \frac{|\{k \mid k \sim i \wedge k \sim j\}|}{|\{k \mid k \sim i \vee k \sim j\}|} \quad (6)$$

The measure of Adamic and Adar counts the numbers of common neighbors, weighted by the inverse logarithm of each neighbor k's degree [1]:

$$f_{\mathrm{AA}}(i,j) = \sum_{k \sim i \wedge k \sim j} \frac{1}{\log(d(k))} \quad (7)$$

The final two common proximity-based link prediction methods are graph kernels. They can be either defined as functions of the adjacency matrix $\mathbf{A}$ of the network, or as sums over all paths from i to j. The symmetric adjacency matrix $\mathbf{A}$ of the graph $G = (V, E)$ is defined as the $|V| \times |V|$ 0/1 matrix defined using $\mathbf{A}_{ij} = 1$ when $i \sim j$ and $\mathbf{A}_{ij} = 0$ otherwise. Both graph kernels have a parameter α, which we set to the value $0.85/\|\mathbf{A}\|_2$, i.e., slightly less than the inverted spectral norm of the adjacency matrix.

The exponential graph kernel is defined as the exponential function of the adjacency matrix [7]

$$f_{\mathrm{EXP}}(i,j) = \left[e^{\alpha \mathbf{A}}\right]_{ij} = \sum_{p \in P_*(i,j)} \frac{\alpha^{|p|}}{|p|!}. \quad (8)$$

The Neumann graph kernel is defined using matrix inversion [6]

$$f_{\mathrm{NEU}}(i,j) = \left[(\mathbf{I} - \alpha \mathbf{A})^{-1}\right]_{ij} = \sum_{p \in P_*(i,j)} \alpha^{|p|}. \quad (9)$$

These expressions make use of the notation $P_*(i,j)$ for the (generally infinite) set of all paths in the network from node i to node j, and of the notation $|p|$ for the length of a path $p \in P_*(i,j)$.

4.2.2 *Centrality-based functions*

Centrality-based link prediction functions are defined as products of centrality measures of the two vertices i and j; different choices of centrality measures lead to different link prediction functions. In addition to the PageRank product defined in Equation (1), we use the preferential attachment value.

The preferential attachment model states that the likelihood of a new node i to connect to node j is proportional to the degree of node j [2]. Thus, the preferential attachment score is defined as

$$f_{\mathrm{PA}}(i,j) = d(i)d(j). \quad (10)$$

4.3 Evaluation Measures

To measure the accuracy of a link prediction function, we use the area under the curve (AUC), defined as the area under the receiver operating characteristic (ROC) curve.

The ROC curve reflects the accuracy of a link prediction function and is constructed as follows. Let f be a link prediction function. All node pairs in the combined true and false test set $E_{\mathrm{true}} \cup E_{\mathrm{false}}$ as defined in Equation 3 are sorted by their value of f. Starting from the best-ranked position, for every position in the ranking the false positive rate is plotted against the true positive rate. The true positive rate equals the number of observed node pairs from the true test set divided by the overall number of node pairs in the true test set. Analogously, the false positive rate is computed as the number of observed node pairs of the false test set divided by the overall number of node pairs in the false test set. The ROC curve is always contained in the square $[0,1] \times [0,1]$. The AUC is then defined as the area under the ROC curve, and is thus a value in the interval $[0,1]$. For a random predictor, the ROC curve approximates the diagonal connecting the points $(0,0)$ and $(1,1)$, giving an AUC value of 0.5, whereas a perfect predictor would yield an AUC value of one. When a link prediction function f is inverted to give $-f$, its AUC value x is replaced by $1 - x$. and its new ROC curve is obtained by rotating the ROC curve of f by 180 degrees.

Alternative measures of accuracy, which we do not use in this paper, are the mean average precision [18] and the normalized discounted cumulative gain [5]. We choose the AUC in this paper since it is robust with respect to changes in the size of the split.

In the following, we describe two experiments to measure how well the negative links can be predicted from the positive links in a signed social network. The purpose of the first experiment is to find good link prediction functions at that task, and to compute their accuracy. The second experiment consists in comparing this link prediction problem to the task of predicting negative links in networks where both positive and negative links are known. Since this task includes more information in the training set (i.e., negative links), the achieved accuracy of that problem is higher and gives an upper bound on the accuracy that can realistically be attained at the problem of predicting negative links when only positive links are known.

4.4 Experiment 1

The goal of this experiment is to measure the accuracy of link prediction functions at the task of predicting negative links in social networks containing only positive links, and to observe which particular functions are well suited for that task.

In our scenario, we want to predict negative links from known positive links. Since we want to compare the scores of link predictions functions applied to node pairs connected by a negative link with the scores of node pairs that are unconnected or connected by a positive link, we split the set of positive edges P randomly into two sets P_{a} and P_{b}. We use the sizes $|P_{\mathrm{a}}| = 3|P_{\mathrm{b}}|$, corresponding to a test set containing 75% of all edges.

The training set is thus P_{a} and the true test set is N. The false test set can be chosen in three different ways to emphasize different features of the tested link prediction functions:

- (P_{b}) Other known positive links in the false test set force a good distinction capability between negative and positive links.

- (O) Only including non-edges in the false test set will emphasize the ability of a link prediction function to distinguish negative edges from non-edges.

- ($P_{\mathrm{b}} \cup O$) Using both positive and non-edges in the false test set evaluates a link prediction function at the task of distinguishing negative edges from both positive edges and non-edges.

The three cases result in the following link prediction problems:

$$P_{\mathrm{a}} \to N \mid P_{\mathrm{b}} \tag{11}$$

$$P_{\mathrm{a}} \to N \mid O \tag{12}$$

$$P_{\mathrm{a}} \to N \mid P_{\mathrm{b}} O \tag{13}$$

Although it may seem sufficient to use the third, combined false test set, our experiments will show that the relative accuracy of individual link prediction functions at the three problems may be radically different, and thus it is a requirement that a good link prediction method is good at all three problems.

4.5 Experiment 2

To assess whether the accuracy of link prediction achieved in Experiment 1 can be considered accurate enough to recommend against the introduction of explicit negative links in online social networks, we compare the results with the results of the related link prediction problem in which negative links are known. This related link prediction function gives an upper bound for the accuracy attainable using the previous methods, and the difference in accuracy between both problems will thus characterize the added value that the *negative link* feature brings to a social networking platform.

We will assume that a part of the negative links in the social network are already known, and include them in the training set. We thus compare the two following link prediction problems:

$$P_{\mathrm{a}} \to N_{\mathrm{b}} \mid P_{\mathrm{b}} O \tag{14}$$

$$P_{\mathrm{a}}, N_{\mathrm{a}} \to N_{\mathrm{b}} \mid P_{\mathrm{b}} O \tag{15}$$

The set of negative edges N is thus split into the two sets $N = N_{\mathrm{a}} \cup N_{\mathrm{b}}$. The split of N is made in the same proportion as the split of P, i.e., $|N_{\mathrm{a}}| = 3|N_{\mathrm{b}}|$.

The first link prediction problem is the same as link prediction problem (11) in Experiment 1 up to the necessary replacement of N by N_{b}; the second one includes additional negative edges N_{a} in the training set. Note that any link prediction function that has a high accuracy in the first problem can be transformed into an accurate link prediction function for the second problem by simply ignoring the negative edges. Thus, the accuracy of link prediction functions at the second problem are upper bounds for the accuracy of link prediction functions at the first problem. The tightness of this bound can then be interpreted in terms of the added value of the negative edges. If the difference is high, negative edges contain information that is not recoverable using only the positive edges, and a *negative link* feature will increase the accuracy of news stream filters and recommender systems based on the social network. If the difference is small, negative links do not give such an added value.

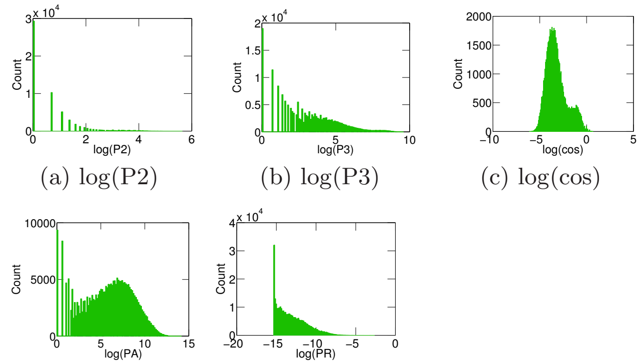
For the second problem, the link prediction methods must be modified to work on signed edges. We follow the methods described in [9], which define the degree $d(i)$ as not depending on edge signs, and essentially replace the number of common neighbors

$$|\{k \mid i \sim k \wedge k \sim j\}|$$

with the difference of positive and negative paths

$$\sum_{i \sim k, k \sim j} w(i,k) w(k,j),$$

which reduces to the number of common neighbors in the unsigned case.



(a) log(P2) (b) log(P3) (c) log(cos)

(d) log(PA) (e) log(PR)

Figure 2: The distribution of feature values for a subset of five link prediction functions, applied to the Slashdot dataset. We use the logarithm of the functions as features for learning an ensemble link prediction function since the logarithmic values are nearer to a normal distribution. (a)–(c) proximity-based functions, (d)–(e) centrality-based functions.

4.6 Ensemble Link Prediction Functions

As shown in Section 3, neither centrality-based link prediction functions such as the PageRank product, nor proximity-based functions such as the cosine similarity are expected to predict negative links from positive links well. Instead, combinations of them are needed. Therefore, we propose a method for combining centrality-based and proximity-based link prediction functions into an ensemble.

To combine several link prediction functions, we use logistic regression applied to the logarithms of individual prediction functions. The reason to take the logarithm is that the values of the computed functions are all distributed in a logarithmic way, i.e., the logarithm of the values are nearer to a normal distribution than the values themselves. Some functions such as the number of common neighbors f_{P2} may be zero, and thus their logarithm is not defined; in this case we use the logarithm of the lowest possible value instead. As illustrations, the distribution of feature values is shown for the main link prediction functions, applied to the Slashdot dataset in Figure 2.

Also, since the behavior of the PageRank product is different when the cosine similarity is exactly zero (as illustrated in Figure 1), we include as a feature the PageRank product multiplied by the indicator function of the cosine similarity being zero. We call this feature the conditional PageRank. Table 2 summarizes all features used in the evaluation.

We propose two ensemble link prediction functions, based on the basic link prediction functions of Table 2. Since the basic features $f_1, \dots, f_9$ correlate among each other (for instance, the Adamic–Adar measure and the common neighbor count have Pearson correlation $p = 99\%$ for the Slashdot dataset), we restrict regression to the five best-performing individual link prediction functions.

- Logistic regression based on the five logarithmic features $f_1, \dots, f_5$.

Table 2: The features used for learning a link prediction function.

Feature	Name	Ref.
$f_1 = \log(\mathrm{P2})$	Common neighbors	Eq. (4)
$f_2 = \log(\mathrm{P3})$	Paths of length three	Eq. (5)
$f_3 = \log(\cos)$	Cosine similarity	Eq. (2)
$f_4 = \log(\mathrm{Jac})$	Jaccard coefficient	Eq. (6)
$f_5 = \log(\mathrm{AA})$	Adamic–Adar	Eq. (7)
$f_6 = \log(\mathrm{Exp})$	Exponential kernel	Eq. (8)
$f_7 = \log(\mathrm{Neu})$	Neumann kernel	Eq. (9)
$f_8 = \log(\mathrm{PA})$	Preferential attachment	Eq. (10)
$f_9 = \log(\mathrm{PR})$	PageRank product	Eq. (1)
$f_{10} = \begin{cases} \log(\mathrm{PR}) & \text{if} \cos = 0, \\ \min(\log(\mathrm{PR})) & \text{otherwise.} \end{cases}$	Conditional PageRank	–

Table 3: The regression link prediction functions used in our evaluation.

Regression	Used features
f_{all}	$f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9$
$f_{\text{PR-cos}}$	f_3, f_{10}

- Logistic regression based on the conditional PageRank f_{10} and the cosine $f_3 = \log(\cos)$.

The two logistic regression-based functions must be trained on the training set. Given a set of features $f_1, \ldots, f_n$, logistic regression finds weights $\alpha_1, \ldots, \alpha_n$ and β such that

$$f_{\mathrm{reg}} = [1 + \exp\{-(\beta + \alpha_1 f_1 + \ldots + \alpha_n f_n)\}]^{-1}$$

approximates the value 1 when a node pair is in the true test set and 0 when a node pair is in the false test set. Regression is performed using the least squares optimization function.

Table 3 shows the two regression features.

The ensemble link prediction methods are only used for Experiment 1, as using them in Experiment 2 to derive an upper bound on link prediction accuracy will skew the results, i.e., we expect that other, more complex link prediction methods may perform better for some datasets.

5. EVALUATION

In this section we apply the methodology described in the previous section to the two signed social networks of Slashdot and Epinions. We perform the two experiments (Experiment 1 and Experiment 2) described in the previous section.

5.1 Experiment 1

We perform Experiment 1 as described in Section 4.4. The AUC values for all link prediction functions for all three link prediction problems are shown in Figure 3. The corresponding ROC curves for the link problem using $P_{\mathrm{b}} \cup O$ as the false test set are shown in Figure 4. The weights learned for logistic regression are given in Table 4.

Observations.

A first observation from Figure 3 is that all individual link prediction functions f_1 to f_9 perform well (AUC > 0.5) at the problem $P_{\mathrm{a}} \to N \mid O$, while their inverses (AUC < 0.5) perform well at the task $P_{\mathrm{a}} \to N \mid P_{\mathrm{b}}$. Thus, none of these

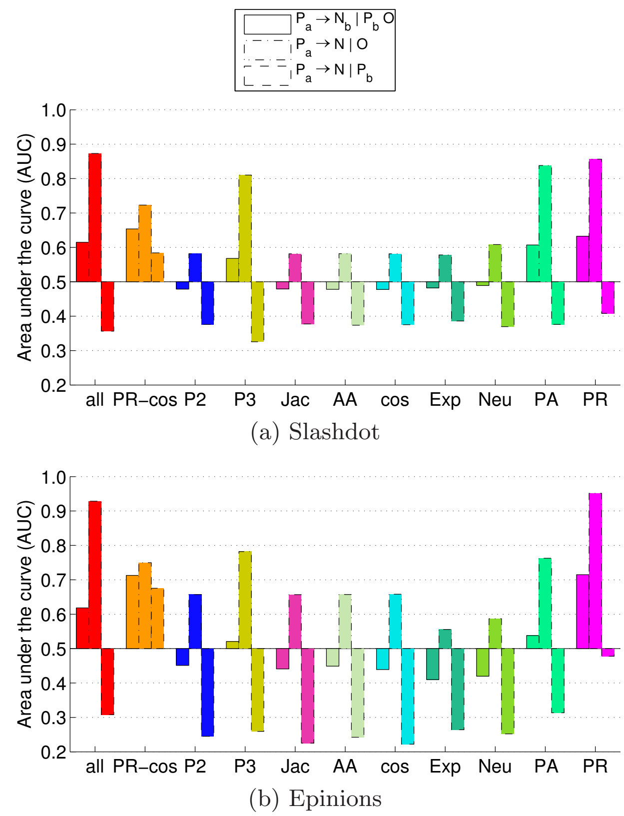
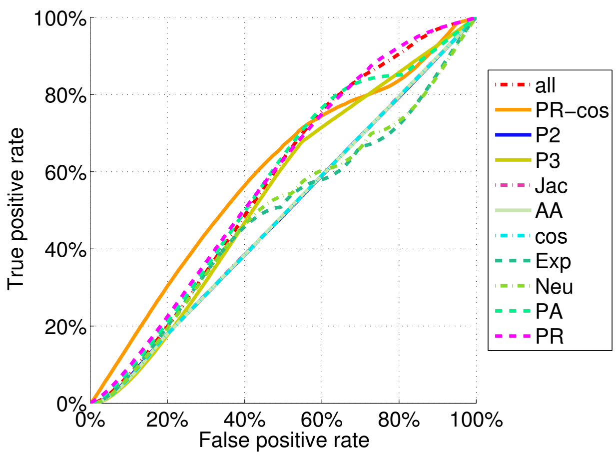


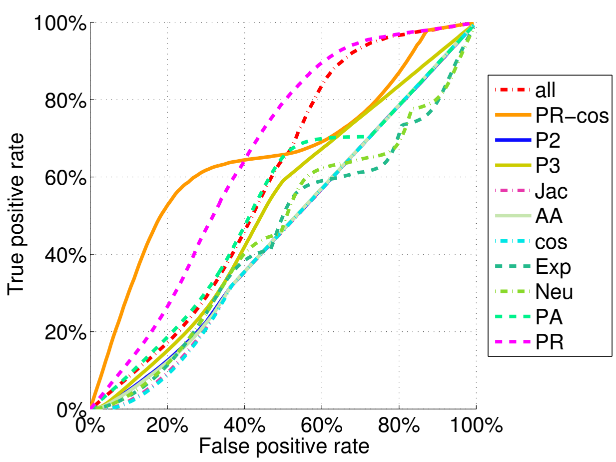
Figure 3: The AUC of the link prediction functions at the three link prediction problems of Experiment 1. The two leftmost functions are ensemble functions; the other functions are the basic link prediction functions. A suitable link prediction function at the task of predicting negative links must have an AUC larger than 0.5 for all three link prediction problem.

Table 4: Learned weights of logistic regression. Weights marked as (–) denote functions that are not used in the respective regression type.

Dataset	Regression	log(P2)	log(P3)	log(cos)	log(PA)	log(PR)	f_{10}
Slashdot	f_{all}	−0.5411	−0.4866	−3.9434	0.2502	0.2321	–
	$f_{\text{PR-cos}}$	–	–	−6.113	–	–	0.2386
Epinions	f_{all}	−0.8587	−0.3827	−5.0360	−0.0105	0.8498	–
	$f_{\text{PR-cos}}$	–	–	−1.5103	–	–	0.5111



(a) Slashdot



(b) Epinions

Figure 4: The ROC curves of all link prediction functions at the link prediction problem $P_{\text{a}} \to N \mid P_{\text{b}}O$ for both datasets. Well performing methods in this experiment have a ROC curve that is higher on the plot than other curves. A high steepness of the curve at the point $(0, 0)$ indicates a high precision for the top-k items, implying a good performance at recommendation tasks.

functions taken by itself is suited to solving our problem. Instead, ensemble methods must be used. Our tests show that the only set of functions that perform well (AUC > 0.5) when combined include the conditional PageRank f_{10}, i.e., regression type $f_{\text{PR-cos}}$. Note that the regression weights in Table 4 cannot be interpreted individually. The regression weights learned for $f_{\text{PR-cos}}$ for both datasets have the same signed and relative weights, and suggest the prediction function

$$f = \alpha \left(\begin{cases} \log(\text{PR}) \text{ if} \cos = 0, \\ \min(\log(\text{PR})) \text{ otherwise.} \end{cases} \right) - \beta \log(\cos),$$

in which the weights $\alpha, \beta > 0$ must be determined experimentally. Figure 4 also shows that this method ($f_{\text{PR-cos}}$) also has the steepest ROC curve at the point $(0, 0)$, implying that this method is best at predicting the top-k unknown negative links for small k. This property is important for the application of recommender systems, in which only the top-k results are used and the rest ignored.

5.2 Experiment 2

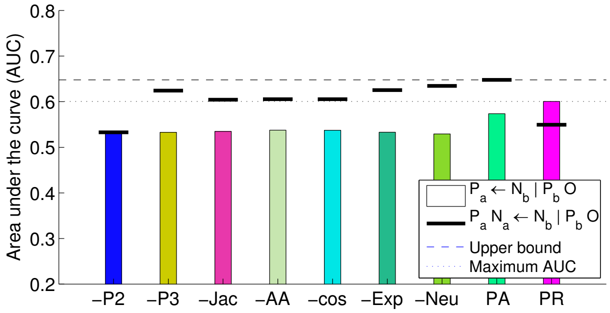
We perform Experiment 2 as described in Section 4.5. In this experiment, the performance of algorithms at the problem $P_{\text{a}}N_{\text{a}} \to N_{\text{b}} \mid P_{\text{b}}O$ serve as an upper bound for the performance of methods at the problem $P_{\text{a}} \to N_{\text{b}} \mid P_{\text{b}}O$. Thus, the results of this experiment can be used to assess how much information is lost when negative links are not recorded in a social network. The results of the experiment for both datasets are shown in Figure 5.

Observations.

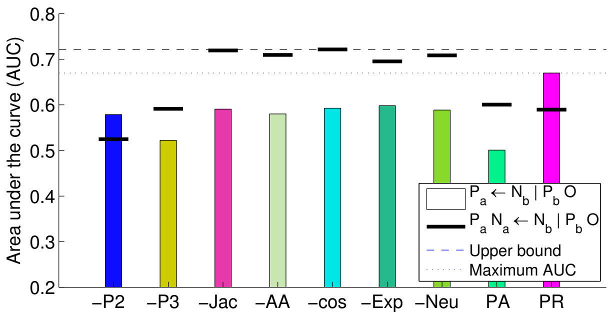
The experimental results show that the best method when negative links are known performs by about 0.05 AUC points better than the best method when no negative links are known. Thus, allowing negative links in an online social network does have an added value for the network, although that added value is small, because the difference in AUC values from one link prediction function to the next are larger than the observed difference of 0.05, suggesting that specific functions adapted to any dataset may be able to close that gap.

5.3 Discussion

The experimental results derived in the two experiments show that the problem of predicting negative links in a social network, using only positive links is a variant of the link prediction problem that can only be solved by combining both centrality-based and proximity-based functions, using positive weights for centrality-based functions and negative weights for proximity-based functions. This result is congruent with the intuition that the existence of an edge (regardless of its sign) correlates positively with centrality-based



(a) Slashdot



(b) Epinions

Figure 5: Comparison of the accuracy of link prediction with and without N_a in the training set. The bars show the AUC values of the link prediction problem in which no negative edges are known. The thick black lines represent the AUC values at the task in which some negative links are known. For the proximity-based prediction functions, the plot shows the AUC values of the inverted prediction functions, since they have AUC values of over 0.5 and are better suited to predict negative links. As expected, the best method when negative edges are known performs better than the best method when negative edges are not known. The observed difference, of about 0.05 AUC points, suggests that allowing negative edges gives an added value to a social network, but much less than expected, as that difference is smaller than the difference from one link prediction function to the next.

functions, showing that models such as preferential attachment, which predict a higher probability of edge attachment for nodes with high degree centrality, is valid independently of edge sign in networks where negative links are allowed. On the other hand, signed networks follow balance theory in that triangles in them tend to have an even number of negative edges, explaining why the proximity-based methods correlate negatively with the presence of negative edges.

6. CONCLUSION

We have shown that in the online social networks Slashdot and Epinions, the *foe* and *distrust* feature is used by users in a way that can be predicted to high accuracy from the *friend* and *trust* links. Thus, with regards maximizing the utility of news stream filtering and social recommendation, the negative link features of these two sites are redundant to a large extent. However, it does not follow that these features are useless. Quite the contrary is true; the *foe* feature of Slashdot is used as a personal organization tool (remembering who is considered a *troll*), or simply to let another user know one's disapproval of them. In Epinions, the *distrust* feature is likewise central to the Epinions's Web of Trust.

As a solution to the generic learning problem of predicting one link type from another one, we showed that the usual link prediction methodology can be applied, but only with the caveat that individual link prediction function may have inverted performance, e.g., the cosine similarity measure in the example of disapproval links.

Finally, as an application of our methods to online social networks that do not allow *foe* or *distrust* links, we propose that a link prediction function learned using regression with Slashdot and Epinions data may be applied. The only way however to ascertain the accuracy of these predictions is to perform the evaluation described in this paper, which by nature of the problem is only possible when negative edges are known.

Acknowledgments

We thank Paolo Massa for providing us with the Epinions dataset. The research leading to these results has received funding from the European Community's Seventh Frame Programme under grant agreement n° 257859, ROBUST.

7. REFERENCES

[1] L. Adamic and E. Adar. Friends and neighbors on the Web. *Social Networks*, 25:211–230, 2001.

[2] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proc. Int. World Wide Web Conf.*, pages 403–412, 2004.

[4] F. Harary. On the notion of balance of a signed graph. *Michigan Math. J.*, 2(2):143–146, 1953.

[5] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[6] J. Kandola, J. Shawe-Taylor, and N. Cristianini. Learning semantic similarity. In *Advances in Neural Information Processing Systems*, pages 657–664, 2002.

[7] R. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proc. Int. Conf. on Machine Learning*, pages 315–322, 2002.

[8] J. Kunegis, G. Gröner, and T. Gottron. Online dating recommender systems: The split-complex number approach. In *Proc. Workshop on Recommender Systems and the Social Web*, pages 37–44, 2012.

[9] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The Slashdot Zoo: Mining a social network with negative edges. In *Proc. Int. World Wide Web Conf.*, pages 741–750, 2009.

[10] C. Lampe and P. Resnick. Slash(dot) and burn: Distributed moderation in a large online conversation space. In *Proc. Int. Conf. on Human Factors in Computing Systems*, pages 543–550, 2004.

[11] A. N. Langville and C. D. Meyer. *Google's PageRank and Beyond*. Princeton University Press, 2006.

[12] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Governance in social media: A case study of the

Wikipedia promotion process. In *Proc. Int. Conf. on Weblogs and Social Media*, pages 98–105, 2010.

[13] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proc. Int. Conf. on World Wide Web*, pages 641–650, 2010.

[14] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *Proc. Int. Conf. on Human Factors in Computing Systems*, pages 1361–1370, 2010.

[15] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proc. Int. Conf. on Information and Knowledge Management*, pages 556–559, 2003.

[16] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A*, 390(6):1150–1170, 2011.

[17] P. Massa and P. Avesani. Controversial users demand local trust metrics: an experimental study on epinions.com community. In *Proc. American Association for Artificial Intelligence Conf.*, pages 121–126, 2005.

[18] M. A. Najork, H. Zaragoza, and M. J. Taylor. Hits on the Web: How does it compare? In *Proc. Int. Conf. on Research and Development in Information Retrieval*, pages 471–478, 2007.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[20] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(1):440–442, June 1998.

[21] S. H. Yang, A. Smola, B. Long, H. Zha, and Y. Chang. Friend or frenemy? predicting signed ties in social networks. In *Proc. Int. Conf. on Research and Development in Information Retrieval*, pages 555–564, 2012.