Using Social Data to Predict Trust on Web Communities: A Case Study with the Epinions.com Website

Denis D. Mauá[1] and Fabio G. Cozman[1]

[1]Escola Politécnica – Universidade de São Paulo (USP)
Av. Prof. Mello Moraes 2231
Cidade Universitária
São Paulo, SP, Brazil

{denis.maua,fgcozman}@usp.br

Abstract. *In this paper we analyze the performance of state-of-the-art machine learning techniques in trust prediction. We use two propositionalization methods together with the Naive and Tree-Augmented Naive Bayesian Classifiers, and the C4.5 algorithm. We compare those results with classifiers defined through Markov Logic, using data from the Epinions.com website, a well-known product review community. The experiments show that predicting trust relationships is a difficult task, in which Markov Logic models outperform other methods in accuracy but are able to recover only a relatively small fraction of the existing relationships in the dataset.*

1. Introduction

Datasets reporting on relationships among people are now ubiquitous in the world wide web. The most prominent examples are the social-networking websites that allow users to manage their friendships and to relate to similar people through web-based interfaces. In such sites users can classify other people, exchange messages, join communities and perform many other "social" acts. Other examples include e-commerce systems, where people can be related by the products they buy; collaborative production systems, where people participate in the production of shared knowledge; web logs, where people can comment on a given author posts; newsgroups, where people exchange messages within a topic.

One of the many kinds of relationship data collected by web-based systems is *trust*. Trust can be seen as a social control mechanism, used to minimize the complexity of the decisions one has to make in a highly uncertain environment [Buskens 1998]. Trust statements found in web sites have already been used to boost the performance of recommender systems [Massa and Avesani 2007] and of anti-spam techniques [Golbeck and Hendler 2004], to evaluate the potential of viral marketing campaigns [Richardson and Domingos 2002], or simply to provide trust metric recommendations upon which people can make better decisions [Golbeck et al. 2003, Richardson et al. 2003]. A central component of all these applications is a *web-of-trust*, a compound of trust statements made by users concerning how much they trust other users. An example of a web-of-trust visualized as a digraph is depicted on Figure 1. According to the graph, John has stated that he has trust on Anna and on Mary, Anna has stated that she trusts Mary and Bob, Mary by its turn has stated that she trusts only Bob, who has not provided any trust statement within the system.

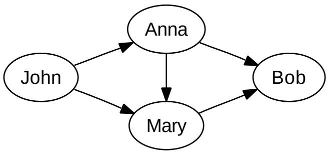

Figure 1. A web-of-trust example with four users.

There is often much more data collected about the relationships among users than direct trust statements. Traditional approaches to the use of web-based trust data have mostly discarded such data on relationships, benefiting only from the trust statements structure, commonly to provide trust recommendations to pairs of users that have not directly stated the amount of trust they have one on each other [Golbeck et al. 2003, Richardson et al. 2003].

The present work explores a different direction, exploiting the existing information of users interaction to *estimate* a web-of-trust. We are thus concerned with the estimation of trust relationships among users of a web-based system, based solely on the aggregation of past experiences recorded with in the system. In particular, we use data from a well-known product-review website regarding evaluation of users' reviews to predict a web-of-trust. A brief summary of the dataset we use here is in Section 2. First, following work by [Sen et al. 2008, Karamon et al. 2007], we first convert our relational data into a form suitable to traditional machine learning techniques [Hastie et al. 2001], in special Naive Bayes, Tree-Augmented Naive Bayes and C4.5 classifiers. Second, we experiment with a recently developed statistical relational learning language, Markov Logic [Richardson and Domingos 2006], trying to directly model the relational characteristics of data. Section 3 presents our proposals for applying traditional classifiers as well as our models developed in Markov Logic. Working with relational data present some difficulties to common machine learning methodology such as splitting data for creating training and test instances. Section 4 discuss the methodology we follow in this work. The results obtained through the use of the cited techniques under the specific methodology are described and discussed in Section 5. Related work is discussed in Section 6, and conclusions and future work are presented in Section 7.

Trust prediction can be very important, because users often refute to make statements, what makes their neighborhood (i.e. the set of users to whom they stated having a relationship) very small. The work presented in here can be used to augment the web-of-trust in real environments by adding predicted relations. Such a procedure can increase the performance of existing trust recommendations methods. Another possible application of this work is the generation of web-of-trust when users are not allowed to provide trust statements, for instance to extract a web-of-trust and use it to make recommendations (indeed, this is similar to what recommender systems do).

2. The Epinions.com System and Dataset

Epinions.com is a peer-review web site where users are encouraged to rate other users' posts in order to improve system usability. Besides being able to rate contents, users can also state the users they trust, and such statements are used to filter information when visualizing reviews on the site. The dataset we use was obtained from the trustlet repository (http://www.trustlet.org) and consists of a digraph of $114,467$ vertices and $717,129$ arcs representing users and trust statements, respectively. The trust network has an average degree of 12.53, a transitivity coefficient of 0.0921 and an average clustering index of 0.078. The dataset also consists of $13, 668, 320$ ratings made by users to $1, 560, 144$ contents. These numbers demonstrate the information overload in the system, and give credit to development of information retrieval systems that enable users to benefit from such a high volume of data.

3. Trust Prediction Approaches

In this section we present the techniques we have used for trust prediction. We pay special attention to the model we have developed in Markov Logic [Richardson and Domingos 2006], as the latter is a new relational language that aims to be a unifying tool in statistical relational learning. Such a language allows us to smoothly do collective classification, which has been shown to perform better in relational datasets [Macskassy and Provost 2007, Sen et al. 2008].

3.1. Data Propositionalization and Propositional Classifiers

Standard machine learning procedures work with individuals represented by a set of attributes and a class label. But the relational data we have contains information on the connection between individuals: individual information is not relevant. Thus, in order to make data suitable to classification techniques we need to transform the relationship information in attribute information.

As we want to classify links, we take our class variable to be the class of a relationship among two arbitrary individuals. So, for every ordered pair of users (u, v) in the system we create an instance $T_{u,v}$ representing a possible trust relationship. $T_{u,v} = 1$ denotes that user u has trust on user v, and $T_{u,v} = 0$ denotes that we can not infer such a trust relationship from data. We call an *encounter* any interaction between a pair of users. In the Epinions.com dataset, for example, an encounter of user u with user v happens when u rates a v content. For every ordered pair (u, v) of users there can be zero or more encounters.

We propositionalize data by aggregating encounters and making them the attributes. Each type of encounter is transformed into an attribute variable and each occurrence of such a kind of encounter is converted into a corresponding attribute instance. In Epinions.com there are five types of encounter, given by the possible rates a user can make on other user content.

We tested two aggregation procedures that we refer to as *counting aggregation* and *binary aggregation*. Counting aggregation is performed by using the total number of encounters of type c from user u to v as the value of cth attribute variable in the $T_{u,v}$ instance. Binary aggregation uses variables as indicators only, placing a 1 in the cth variable if any encounter of type c has occurred from user u to user v in the $T_{u,v}$ instance row, and 0 otherwise.

There are many other aggregation procedures that could be used. We have tested some of them, but obtained low-accuracy classifiers (for instance, aggregation by the mean average). Others classification schemes remain as work for the future. An in-depth study of the aggregation techniques is found in [Karamon et al. 2007].

There is a large number of propositional classifiers in the literature. We have tested out three state-of-the-art classification techniques [Hastie et al. 2001]. The first two are based on Bayesian Networks, a graphical probabilistic language. The third one is based on decision trees.

- *Naive Bayesian Classifiers* (Naive Bayes) are based on the assumption that attribute variables are independent of each other given the class. For the propositionalization procedure described in Subsection 3.1, this means that a given type of encounter has no influence in another type of encounter when estimating the existence of a trust relationship. Although the independence assumption is often incorrect, Naive Bayes Classifiers have been reported to perform well in a variety of tasks, being easy to implement and efficient in large datasets.
- *Tree-Augmented Naive Bayesian Classifiers* (TAN) are an attempt to soften the hard attribute independence assumptions made by Naive Bayes classifiers. TANs allow dependence connections between attributes (an attribute may have a single attribute as parent) in a way that learning and inference are still efficient. In many domains TAN has been shown to outperform Naive Bayes, keeping the algorithm simplicity low and its efficiency high.
- *C4.5* is a learning algorithm that uses a decision tree as a predictive model. It is somewhat related to TAN in that it allows dependence among attribute variables to be taken into account and uses information entropy to “prefer” simpler models to complex ones.

3.2. Markov Logic Classifiers

Markov Logic is a probabilistic relational language that can be used to create complex Markov Networks from simple and concise templates [Richardson and Domingos 2006]. Its syntax is given by a set of weighted first-order logic formulae and its semantics is defined by equation (1) that assigns a probability value to every possible world defined by the syntax. In the equation, Z is a normalization factor given by $\sum_x \exp(\sum_i w_i f_i)$, i ranges over all grounded formulae in the knowledge base, w_i are the weights of the ith grounded formula and f_i is a feature that takes value 1 if the ith formula is true, and 0 otherwise.

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i f_i\right) \qquad (1)$$

Models in Markov Logic can been seen as an aggregation procedure where data is combined by means of first-order logic. The main benefit of Markov Logic is that we can easily define complex probabilistic models, and learning/inferences is relatively straightforward. As one can see from our following discussion, modifying a model in Markov Logic is simple (corresponding to add, remove or change a formula in its first-order logic base knowledge).

We develop two models to the trust prediction task. The first one only takes into account the encounters between users to predict trust relationships. We create rules stating that each type of encounter has a particular influence on the final estimate of a trust relationship.

w_1 `rates(u,v,1)` ⇒ `trusts(u,v)`,
w_2 `rates(u,v,2)` ⇒ `trusts(u,v)`,

$$
\begin{array}{l}
w_3 \ \texttt{rates(u,v,3)} \Rightarrow \texttt{trusts(u,v),} \\
w_4 \ \texttt{rates(u,v,4)} \Rightarrow \texttt{trusts(u,v),} \\
w_5 \ \texttt{rates(u,v,5)} \Rightarrow \texttt{trusts(u,v).}
\end{array}
$$

In this model, each possible rating a user can make on another user content may have a different influence in the probability of a trust relationship between them. The weights are to be estimated by applying a gradient-ascent method over the pseudo-likelihood of the joint probability [Richardson and Domingos 2006].

Our second model includes the fact that most social networks present the small world phenomenon [Wasserman and Faust 1994], where users tend to group into small clusters with similar properties. In our case, this would mean that if a user x trusts user y and user y trusts user z then it is likely that user x will trust user z as well. We call this *trust propagation* and model it in Markov Logic by adding to the previous model the trust propagation formula:

$$w_p \ \texttt{trusts(x,y)} \wedge \texttt{trusts(y,z)} \Rightarrow \texttt{trusts(x,z).}$$

4. Methodology

In this section we describe data preprocessing and criteria to evaluate different methods.

As usual in machine learning methods, we need to sample smaller subsets from the original data in order to create training and test sets (respectively to adjust the parameters and to validate the output of the algorithms). The common way of separating data is to randomly removing a certain amount (e.g. 60%) of the original dataset, forming the training set, and setting the remaining data to be the test set. However, with strongly relational data such as the one used here such a task cannot be done without strongly biasing the data, because randomly sampling individuals do not take into account their ties (the actual data that matters). Besides, using a random sample strategy makes the sample dataset not disjoint, because there often are ties between the individuals in the training part and those in the test part. Sampling relationships rather than individuals incurs in the same problems.

Another problem we face is that currently the available algorithms for learning and inference in Markov Logic do face scalability problems, and they are not capable of dealing with large databases as Epinions. Thus, we need to use smaller datasets so that our experiments can be performed in reasonable time and with limited resources.

We have developed a sampling algorithm based on the one presented by [Sen et al. 2008] to create sample datasets. The algorithm, called *Snowball Sampling*, appears as Algorithm 1. There, $T = (\mathcal{V}_T, \mathcal{E}_T)$ is the original trust statement data structured as a digraph, N is the number of nodes to be sampled, d is the max number of neighbors to jump to for each node in the current iteration seed, and s is the number of nodes in the initial seed. Basically, it starts by randomly sampling a given number of individuals, the seed, and then proceed by adding the neighborhood of each element in the seed and all the existing ties among the sampled individuals. The algorithm continues by making the new aggregated elements to be the next seed, exploring their neighborhood and continuing until a pre-specified number of individuals has been collected. Guarantees concerning the distribution of sampled data have not been obtained so far (we leave this

Algorithm 1: SnowballSampling($\mathcal{V}_T$,$\mathcal{E}_T$,N,d,s)

```
$seed \leftarrow$ randomly sample $s$ nodes from $\mathcal{V}_T$
$\mathcal{V}_S \leftarrow \emptyset$
$\mathcal{E}_S \leftarrow \emptyset$
**while** $|\mathcal{V}_S| < N$ **do**
    $newseed \leftarrow \emptyset$
    **while** $|seed| > 0$ **and** $|\mathcal{V}_S| < N$ **do**
        $n \leftarrow$ get an element from $seed$
        $\mathcal{V}_S \leftarrow \mathcal{V}_S \cup \{n\}$
        $\mathcal{V}_n \leftarrow$ randomly sample $d$ neighbors of $n$
        **while** $|\mathcal{V}_n| > 0$ **and** $|\mathcal{V}_S| < N$ **do**
            $nn \leftarrow$ get an element from $\mathcal{V}_n$
            **if** $nn \notin seed$ **then** $newseed \leftarrow newseed \cup \{nn\}$
            **if** $nn \notin \mathcal{V}_S$ **then** $\mathcal{V}_S \leftarrow \mathcal{V}_S \cup \{nn\}$
            $\mathcal{E}_S \leftarrow \mathcal{E}_S \cup \{(n, nn)\}$
            **for** $u \in \mathcal{V}_S$ **do**
                **if** $(u, nn) \in \mathcal{E}_T$ **then** $\mathcal{E}_S \leftarrow \mathcal{E}_S \cup \{(u, nn)\}$
                **else if** $(nn, u) \in \mathcal{E}_T$ **then** $\mathcal{E}_S \leftarrow \mathcal{E}_S \cup \{(nn, u)\}$
            **end**
        **end**
    **end**
    $seed \leftarrow newseed$
**end**
```

for future work), but our experiments have shown that it indeed produces more homogeneous data than random sampling. We measure data homogeinity by the average degree, that is, the mean average of the number of ties a individual has; the transitivity, that is, the proportion of transitive relationships present in the network; and the average clustering, that is, the mean average of the proportion of transitive relationships in the neighborhood of an individual. A transitive relationship is a triangle in the corresponding graph of the web-of-trust indicating a situation where someone trusts the trustees of their trustees.

We now turn to evaluation metrics. Two common metrics used in information retrieval are the *precision* and the *recall*. The former measures how accurate are the predicted ties, and is calculated as the ratio of the correct instances over the total number of estimated relationships. A high value indicates that predicted trust relationships very often exists in practice. The latter summarizes the proportion of true relationships that the algorithm was able to recover, and is computed as the ratio of correct estimates over the total number of true relationships in the dataset. A high value for this measure indicates that the algorithm was able to retrieve most of the existing relationships. Finally, a balance between precision and recall can be obtained by means of their harmonic mean, denoted *F1-score*.

Let $T = (\mathcal{V}_T, \mathcal{E}_T)$ and $E = (\mathcal{V}_E, \mathcal{E}_E)$ denote the true web-of-trust and the estimated web-of-trust, respectively. The prediction precision p, recall r and F1-score f_1 are thus computed, respectively, by $p = |\mathcal{E}_T \cap \mathcal{E}_E|/|\mathcal{E}_E|$, $r = |\mathcal{E}_T \cap \mathcal{E}_E|/|\mathcal{E}_T|$, $f_1 = 2|\mathcal{E}_T \cap \mathcal{E}_E|/(|\mathcal{E}_T| + |\mathcal{E}_E|)$.

Table 1. Properties of the three datasets used in the experiments.

Dataset	$\|\mathcal{E}\|$	d	t	γ
S_1	613	12.26	0.19	0.23
S_2	500	10.0	0.15	0.18
S_3	416	8.32	0.13	0.16

5. Experiments and Discussion

We used the snowball sampling procedure with $N = 100$, $d = 10$ and $s = 1$ to produce three distinct graph samples from the original dataset. The number of individuals in each sample was chosen to be low so that we have a low probability of high correlation between samples and that the Markov Logic models can be performed. Markov Logic currently suffers for scalability problems (with respect to learning and inference), so the number of samples and the order of each sample have to be a balance between usefulness and feasibility.

We refer to samples by S_1, S_2 and S_3. Table 1 depicts a summary of sample properties. $|\mathcal{E}|$ denotes the number of trust statements, d the average degree of vertices, and t and γ the transitivity and the average clustering coefficients of the related graph.

To evaluate the classifiers on the trust prediction task, we developed six baseline algorithms. The first one, that we denoted ALWAYSY, returns always true when asked if a given user has trust on another one. Thus, its precision is simply the proportion of relationships in the dataset, and its recall is always maximum. Although such a method is of no use in real domains, it is practical to analyze how a strongly biased algorithm would perform. The other five baseline methods are denoted RULEc and work by creating simple rules that infer that if and only if user u has an encounter of type c with other user v than the trust variable $T_{u,v} = 1$. In the Epinions.com dataset there are five possible types of encounter resulting in five different RULE methods.

In order to minimize the possible influence of a particular sample we ran experiments with the six possible permutations of the three datasets as training/test instances. So we used S_1 to train the algorithms and validate them on S_2 and on S_3, then to learn algorithms with S_2 and to validate on S_1 and on S_3; and finally we used S_3 to learn algorithms and S_1 and S_2 as validation sets. We averaged the results of each of the six runs and computed the standard deviation. The results are shown in Table 2, where p denotes the precision, r the recall and f_1 the F1-score of the respective method. The first five rows in the table report on the baseline methods results. MLN1 and MLN2 denotes the first (encounter-based) and the second (propagation augmented) models in Markov Logic. NB, TAN and C4.5 stands for Naive Bayesian Classifiers, Tree-Augmented Naive Bayesian Classifier and C4.5 Decision Tree Classifier, respectively. The suffixes c and b indicates whether counting or binary aggregation was used to propositionalize data. We highlight the best methods under each metric in the baseline and in the classifiers block.

The results, at a first glance at least, are surprising. Simple rule methods like RULE4 and RULE5 were able to achieve good precision or recall values. Indeed, most of the rating data in the Epinions.com dataset is compound by 4 grade ratings, what can explain the high ability to recover trust relationships of the RULE4 method. On the other

Table 2. Experimental results.

Method	p (%)	r (%)	f_1 (%)
ALWAYSY	5.10± 0.99	100.0 ± 0.0	9.69± 1.79
RULE1	0.05± 7.2	0.51± 0.28	0.97± 0.55
RULE2	10.66± 6.27	1.37± 0.55	2.42± 1.03
RULE3	20.02± 0.86	22.06± 4.18	20.9 ± 2.34
RULE4	22.24± 2.5	**84.43± 3.59**	**35.14± 3.1**
RULE5	**45.07±10.73**	6.07± 4.61	10.12± 6.89
MLN1	**51.4 ±11.64**	8.80± 5.33	14.13± 6.86
MLN2	49.75±10.21	8.78± 5.25	14.06± 6.74
NBc	40.25± 4.23	22.70± 7.74	**28.18± 5.88**
NBb	24.85± 1.59	**23.6 ± 3.99**	24.12± 2.71
TANc	28.0 ±22.42	7.58±10.63	10.64±13.46
TANb	13.32±20.74	2.67± 4.49	4.4 ± 7.29
C4.5c	34.34±33.58	4.42± 5.08	7.38± 7.92
C4.5b	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

hand, type 5 encounters were rare and its presence were quite good at predicting a relationship, but it was only able to capture a small portion of the web-of-trust. These results display the fact that the Epinions.com data is very biased. With a more uniform distribution over ratings we expect these simple methods to perform much worse.

Markov Logic models had the best precision over all models with a precision of 51.40% and 49.75% for models 1 and 2. Their recall, on the other hand, was not so good, helding values under 9%. Naive Bayesian models perform the opposite, having the best recall after ALWAYSY and RULE4 baseline methods with values of 22.70% and 23.60% for the counting and the binary aggregation, respectively. Their F1-score were also pretty good, competting only with RULE4 F1-score, whose value was lifted by the high value of its recall. TAN and C4.5 have been shown to perform badly in this scenario, standing among the worse results. In fact, one can see that in the binary propositionalization version of C4.5 the algorithm was not able to predict any relationship.

It is also worth noting that the standard deviation for the precision metric was very high in general, reaching the order of the precision value itself in the case of C4.5, and being greater than it in the case of TAN. This was caused by the great disparity in accuracy through the use of different training/test set pairs. Some methods like TAN and C4.5 were very sensitive to the training data. TAN had a poor performance with S_2 as training set and C4.5 perfomed badly when S_3 was used for learning. Other methods such as MLN and NB were more sensitive to the sample used as test set. Both MLN and NB had their best performance when S_1 was used for validation ($\sim$63–66% for MLN and $\sim$42–47% for NB).

6. Literature Review

Exploiting trust relationships in social networks found on the internet is an active field of research in the computer science community. In [Abdul-Rahman 1997], a *web-of-trust* is first introduced to deal with the issue of credential exchange. Instead of having a few entities working as absolute authorities, the web-of-trust allows any one to be a source of credentials, this way decentralizing the signature emission and checking tasks. The same idea is used in [Richardson et al. 2003] to present a solution to the problem of information reliability in the semantic web. Information veracity is checked by propagating trust metrics among a network of interacting entities. In [Golbeck et al. 2003], this idea is taken to web-based social networks, in order to provide trust quantity recommendations. Such recommendations can be used for example to rank e-mails [Golbeck and Hendler 2004] or filter information [Massa and Avesani 2007, Richardson and Domingos 2002]. All these works presume a fully-observable web-of-trust to exist at first.

The procedures we adopted here resemble those of the link-prediction literature, where the objective is to predict the existence of ties among individuals based on attributes of individuals and links. The work of [Popescul and Ungar 2003] and [Taskar et al. 2003] look for solutions of the problem. Whereas they deal with constrained networks such as citation networks, web pages from university departments and students friendship, our work is based on a more general social structure, that of trust relationships in a product review website.

7. Conclusion and Future Work

In this work we tackled the problem of predicting trust relationships among users of a web community based only in social data such as observed interactions. Particularly, we analysed real data from a well-known product-review website and conducted experiments using different machine learning techniques.

Our results show, first, that trust prediction is a difficult task. This can be seen by the low values of precision and F1-score of all tested methods. Relational classifiers such as our Markov Logic models seems to produce more accurate estimates, but are only able to recover a small fraction of the true relationships present in data. On the other hand, simple rule-of-thumbs such as RULE4 have been shown to be good retrieval algorithms, exhibiting high recall values.

Because the methods had relatively low accuracies, we recommend the use of trust predictors only in collaboration with other techniques such as trust propagation procedures, that may benefit from augmenting their input data with more information [Richardson et al. 2003, Golbeck et al. 2003].

Tasks such as suspicious scoring, however, may benefit from more balanced methods with a higher F1-score such as the case of Naive Bayesian Classifiers, increasing the range of relationships captured without much loss in precision. This is the case where it is cheap to check estimated data for veracity but it may be expensive to produce true estimates.

Data structure has shown to have a great impact in the performance of the different methods tested. It has also shown to impact differently each type of algorithm used. Future research must be done to investigate the causes of such effects in link prediction tasks based solely on link attributes such as the case of this work, and to what extent different methods such as the ones used here are affected by different data properties.

Another task for the future is the study of other aggregation techniques and the enrichment of the Markov Logic models, perhaps with a mix of propositionalized data in order to reduce grounded model complexity.

Acknowledgements

This work has been developed in collaboration with HP.

References

Abdul-Rahman, A. (1997). The pgp trust model. *the Journal of Electronic Commerce*.

Buskens, V. (1998). The social structure of trust. *Social Networks*, 20:265–289(25).

Golbeck, J. and Hendler, J. A. (2004). Reputation network analysis for email filtering. In *CEAS*.

Golbeck, J., Parsia, B., and Hendler, J. A. (2003). Trust networks on the semantic web. In Klusch, M., Ossowski, S., Omicini, A., and Laamanen, H., editors, *CIA*, volume 2782 of *Lecture Notes in Computer Science*, pages 238–249. Springer.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2001). *The Elements of Statistical Learning*. Springer.

Karamon, J., Matsuo, Y., Yamamoto, H., and Ishizuka, M. (2007). Generating social network features for link-based classification. In Kok, J. N., Koronacki, J., de Mántaras, R. L., Matwin, S., Mladenic, D., and Skowron, A., editors, *PKDD*, volume 4702 of *Lecture Notes in Computer Science*, pages 127–139. Springer.

Macskassy, S. A. and Provost, F. (2007). *A Brief Survey of Machine Learning Methods for Classification in Networked Data and an Application to Suspicion Scoring*, volume 4503, pages 172–175. Springer Berlin / Heidelberg.

Massa, P. and Avesani, P. (2007). Trust-aware recommender systems. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24, New York, NY, USA. ACM.

Popescul, A. and Ungar, L. H. (2003). Statistical relational learning for link prediction. In *IJCAI03 Workshop on Learning Statistical Models from Relational Data*.

Richardson, M., Agrawal, R., and Domingos, P. (2003). Trust management for the semantic web. In *Proceedings of the Second International Semantic Web Conference*, Sanibel Island, Florida.

Richardson, M. and Domingos, P. (2002). Mining knowledge-sharing sites for viral marketing. In *KDD*, pages 61–70. ACM.

Richardson, M. and Domingos, P. (2006). Markov logic networks. *Mach. Learn.*, 62(1-2):107–136.

Sen, P., Namata, G. M., Bilgic, M., Getoor, L., Gallagher, B., and Eliassi-Rad, T. (2008). Collective classification in network data. Technical Report CS-TR-4905, University of Maryland, College Park.

Taskar, B., Wong, M. F., Abbeel, P., and Koller, D. (2003). Link prediction in relational data. In Thrun, S., Saul, L. K., and Schölkopf, B., editors, *NIPS*. MIT Press.

Wasserman, S. and Faust, K. (1994). *Social network analysis*. Cambridge University Press, Cambridge.