Trust Prediction Models using PSL

Niharika Srivastav
UC Santa Cruz
nsrivast@ucsc.edu

Abstract

Trust prediction in social media networks is a very important aspect in understanding the probable relations between social media users. To do a proper study of a social media network, for recommender systems, advertising, or prediction systems, trust analysis between users proves to be very useful. This project mainly focuses on understanding and comparing different PSL trust models (or combinations of psl trust models) to predict trust between individuals in a social network. The trust models in this project are based on the attributes of people, similar interests, and/or existing trust information. Also, in this project we use PSL (a statistical relational learning framework) to measure trust as a float value between 0 and 1 and not a binary value. This helps in giving a more accurate understanding of the relationship between users. In real life situations, this makes more sense, since we don't have a binary 0/1 trust between people. PSL helps to recognize the relational information in a social network, and the soft truth values help to quantize the trust values to varying degrees of trust.

Introduction

Trust can be defined as the 'willingness of a party to be vulnerable to actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party' (Mayer, Davis, and Schoorman 1995). Almost all decisions a person makes are subconsciously dictated by trust. It plays an important role in almost all human interactions. Trust plays an important role in people's personal and professional relationships, families, and social interactions. Trust prediction in social networks is required to solve the enigmatic problem, which consists of predicting a trust value when the trustor has no direct previous experience with the trustee.

Trust is a complex social concept and an important component of almost all social interactions/ relations. This phenomenon can be calculated between friends, acquaintances, colleagues, etc., in different social relationships. Thus, trust prediction analysis is very useful in solving a wide range of problems. All such social networks are based on social interactions between people, which are inherently based on a certain level of trust. Trust is a very subjective parameter and it exists in all human interactions. We trust known people and even strangers with important personal information and tasks based on the amount of trust we have in them. Some trust relationships may be very critical or personal (like with doctors, family, friends, close relationships) and some trust relationships might be with unknown people (delivery services, seller-buyer relationship, taxi drivers). The definition of trust and the degree of the trust can be different in different situations, and can also depend on the context of the social interaction. Social media network analysis is useful for measuring trust levels in online social networks, for example, retweet behaviour detection (Bild et al. 2015), (Abdullah et al. 2017), fake news detection (Ghafari et al. 2018), recommender systems (Ma, Lu, and Gan 2015), (Yu et al. 2016) social spammer detection (Li et al. 2015) and influence spread problem (Calio and Tagarelli 2019), (B. Abu-Salih and Albahlal 2020).

The past models have used various kinds of features and parameters (like user reputation, social context) and various algorithms (like neural networks, supervised and unsupervised learning) to predict the trust values between people in social networks (Liu, Zhang, and Yan 2018), (W. Sherchan and Paris 2013), (Tang and Liu 2015), (A. Josang and Boyd 2007). Many of these social network analyses consider trust to be a binary valued parameter. In most real life scenarios, this may not be the best form of trust representation. Therefore, in this project we use the concept of real values (between 0 and 1) for trust. This gives a more accurate representation of trust between individuals in a social network, since strict logical rules are not very accurate predictors for a social concept with a lot of variance and forms and degrees of trust between people.

In this paper we used PSL (Probabilistic Soft Logic), a statistical machine learning framework. PSL is particularly effective to solve this problem (Bach et al. 2017). It uses first order rules with soft truth values. This helps to calculate a degree of trust between individuals. This is a more natural and intuitive framework for trust analysis. This project deals with predicting trust relations (edges of the social network graph) in a partially labelled social network of users (nodes). This is done by defining first order logical rules based on different trust models. This project does a comparative study of models from four different papers (Huang et al. 2012), (Huang et al. 2013), (Bach et al. 2015), (Bach et al. 2017). These papers overall model 13 different trust models and study different aspects of trust models in different ways on different datasets. This project does a holistic analysis of all the models. We apply these models on both datasets studied in these papers.

The rest of this report is organized as follows: The next section - Related Work presents with different kinds of trust prediction features and algorithms used in previous papers. The following section - Methodology, describes how the trust is represented in this paper and how we use PSL to implement different trust prediction models. After explaining methodology, the next section - Empirical Evaluation, explains the dataset, the trust models and the experimental results seen in this project. Lastly, Conclusion summarizes the results and the key findings of this work and explains some possible future work. The Model Appendix at the end of the paper, details the different rules within the different trust models.

Related Work

With the development of scientific knowledge and technology, different branches of science that focused on human behavioural analysis and human interaction analysis started to study the concept of trust. Trust prediction in a social network has been researched in many different ways. Various different papers suggest different algorithms, techniques, features or latent models. The most common method has been to use the most basic form of machine learning models using existing trust relationships and similarity information between people based on a known aspect and predicting other trust relationships using the existing information (Liu, Zhang, and Yan 2018).

Social media websites are one of the most popular online collaborations, where people share their experiences with a large number of unknown people, as well as their friends. The social interactions among users in such online communities are constructed based on trust that is established from each user's subjective perspective on the limited interaction experiences within the community. This paper (Kim and Phalak 2012) measures a degree of trust based on users' expertise and preferences regarding various categories, using users feedback rating data which are available and much denser than a web of trust. Another way to predict trust between users is by calculating the reputation of users (Nuñez-Gonzalez and Manuel Graña 2015). In this method reputation is feature chosen subjectively based on background information of users and existing (known) trust relationships. The known trust relationships and reputation of users (in one area of expertise) is used as features to train machine learning models. Below we discuss some specific aspects of trust prediction research and the different kinds of algorithms used, in the previous papers.

Context-Dependent Trust Modelling:

In this method of trust prediction models uses context information(Ghafari et al. 2018), (Tang and Liu 2015). This model is based on the logic that trusting someone in one context does not assure trusting them in another context (Tang and Liu 2015). This (Tang, Gao, and Liu 2012) paper talks about the context dependency of trust in the dataset from a real-world product review website. The data from the website explicitly gives an option to the users to indicate which users are trustworthy. (Tang et al. 2013) used this information as the truth values for their analysis. The paper considered items' categories (e.g., electronics, sports and entertainment) as the context of trust and reported that: 'less than 1% of users trust their friends in all categories' and 'on average, people trust only 35.4% of their friends in the networks for a specific category'. Hence, people trust each other only in certain contexts. Context is the information about the condition of an entity (Zheng et al. 2014). For example, if we consider an illustration of a single context (focusing on the topic of the trust), consider Alice who is a chef in a big multi cuisine restaurant. She trusts the head chef, Barbara for cooking tips and guidance; however, she does not necessarily trust her in topics related to sports. As a result, predicting pair-wise trust relations with respect to the different context of trust can be a difficult task.

Supervised Learning Methods:

(H. Liu and Kim 2008) proposed a trust prediction model and a classifier that works with a set of users' features and interactions using supervised learning methods. (Matsuo and Yamamoto 2009) focused on an e-commerce website called @cosme, and were the first to describe the concept of community gravity: a two-way effect of rating and trust. This was followed with a model to formulate the trust prediction and rating prediction problems. (N. Ma and Liu 2009) discusses a personalised and cluster-based classification trust prediction model that creates user clusters and then trains a classifier for the users. (M. Graña and Kamińska-Chuchmała 2015) introduced a supervised trust prediction approach: a binary classification that focuses on peoples' perceived reputation. (Wang, Wang, and Sun 2016) proposed a trust-distrust prediction method that also used the Dempster-Shafer theory and neural networks. This paper also analysed the effects of homophily theory, emotion tendency and status theory in trust relations (Wang, Wang, and Sun 2016). (Zhao and Pan 2014) developed another supervised trust prediction approach: a classifier with a feature set that included several trust-related factors. These features could be demographic features (e.g., age and Gender), profile features (e.g., number of followers and followees), numeric representation of textual contents provided by users and etc. (S. M. Ghafari and Orgun 2019). They used the existing trust labels for training their classifier. However, the main shortcoming of these approaches is the fact that because of the sparsity of trust relations in online social networks, they have not enough label data available for their training process.

Unsupervised Learning Methods:

(Tang et al. 2013) developed an unsupervised trust prediction model called hTrust. It utilizes the homophily effect on the trust prediction procedure by focusing on similar users. In this way, Tang et al. identified similar users based on the users' ratings similarity. They considered these aspects for rating similarities: network users who rated similar items, network users who gave similar ratings for similar items and network users who had similar ratings patterns. (Y. Wang and Cai 2015) proposes an unsupervised trust prediction model, sTrust, using social status theory and the PageRank algorithm (Page et al. 1999). In this approach, if a user has a higher social status in an online social network, he or she is more likely to be trusted by other network users.

(Guha et al. 2004) developed a trust prediction model that propagates trust based on users' trust or distrust relations with others. Golbeck (Golbeck 2006b) put forward a website called FilmTrust which used trust to produce movie recommendations. (Wang et al. 2018a) proposed a trust prediction approach that, in addition to learning low-rank representations of users, also learned these sparse components of the trust network (Wang et al. 2018a). (Zheng et al. 2014) suggested an unsupervised trust prediction model based on the concept of trust transference, to transfer trust between different contexts (Zheng et al. 2014). (Y. Wang and Liu 2015) introduced an unsupervised trust prediction model to infer trust among users with an indirect connection. (G. Liu and Zhou 2018) proposed a trust inference model, incorporating factors such as residential location and out degree. (Wang et al. 2018b) proposed a novel trust prediction model, CATrust, for auction websites, using Bayesian inference based on Markov Chain Monte Carlo. More importantly, their model considered the contexts of trust.

Why is Trust Prediction difficult?:

User-specified trust relations are extremely improbable (Wang et al. 2018a). For example, 'the density of a typical trust network in social media is less than 0.01' (Tang and Liu 2015), (Golbeck 2006b). As another example, 'the sparsity of Advogato, Ciao, and Epinions, and Flixster [these are some datasets often used in trust prediction related research], i.e., the ratio of the observed trust relations to all the possible relations, is 0.001%, 0.003%, 0.004%, 0.004% and 0.0035%, respectively (Ghafari et al. 2018), (Tang and Liu 2015), (Chen and Gao 2018), (Wan 2017). It is challenging to predict the trust relations well with so limited observed links' (Wang et al. 2018a). Moreover, trust relations follow the rules of the power law distribution: many trust relations can be accounted for by a small number of users and a large number of users participate in only a few trust relations (Tang et al. 2013). For any trust prediction approach in the online social networks, the number of known user-specified trust relations compared to all possible relations among users is low. This makes the pairwise trust prediction problem in online social networks a difficult task.

Methodology

This section describes the methodology used to implement the PSL trust models in this paper. We start with explaining how the trust relation is represented in the PSL trust models. After that we explain how we build models using logical rules to simulate real life trust relationships. After which we describe the specific trust models used in this project.

Trust Representation

A trust relation between two users is considered a unidirectional relationship between a source user (trustor) and the user who is being trusted (trustee), that indicates that the trustor trusts the trustee. Probabilistic soft logic (PSL) (Broecheler, Mihalkova, and Getoor 2010) is a system for probabilistic modeling using first-order logic rules. PSL uses soft-truth values, and allows the truth values to be in the interval [0, 1] all the while adapting logical connectives accordingly. The soft logic formulation makes the inference problem in PSL a convex optimization problem. Next section gives a short overview of PSL, its usage, and its internal representation.

PSL uses first-order logic (FOL) as its underlying modeling language. In a PSL program, relationships and attributes are modeled by different predicates, and first order rules model dependencies or constraints on these predicates (Huang et al. 2013), (Huang et al. 2012). For example,

$$\text{TRUSTS(X, Y)} \rightarrow \text{KNOWS(X, Y)}$$

reads as "if X trusts Y, then X knows Y", where X and Y are variables referring to arbitrary objects. Here, they refer to the user names in the social media network in question. Replacing these variables (X,Y) with constants (real names) from the domain of the program results in a ground rule. PSL extends the notion of rule to the soft context, i.e., rules can be assigned a weight, indicating at what expense a grounding of the rule can be violated or not true for the dataset. For example, the following rule,

$$0.6\ \text{TRUSTS(X, Y)}\ \ \text{TRUSTS(Y, Z)} \rightarrow \text{TRUSTS(X, Z)}$$

models that the trust link between X, Y and Z is not fully transitive and gets weaker along chains of links, by a factor. 0.6 signifies the strength of the specific rule. Furthermore, a PSL program specifies known truth values for a subset of ground atoms. For example, KNOWS(Alice,Bob) = 1.0 and TRUSTS(Alice, Bob) = 0.6 indicate that Alice knows Bob, but only trusts him somewhat above average. Throughout, the text, we use the convention that predicates are written in small caps (e.g., TRUSTS) and variables are italicized capital letters (e.g., A). To relax Boolean truth values to continuous variables, PSL uses the Lukasiewicz t-norm and its corresponding co-norm as the relaxation of the logical AND and OR, respectively. These relaxations are exact at the extremes, when variables are either true (1.0) or false (0.0), but provide a consistent mapping for values in-between. The formulas for the relaxation of the logical conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$) are as follows:

$$\begin{gathered} a \wedge b = \max\{0, a + b - 1\}, \\ a \vee b = \min\{a + b, 1\}, \\ \neg a = 1 - a, \end{gathered}$$

The logical conjunction and disjunction are relaxation from the Boolean domain. Rules are evaluated using the Lukasiewicz norms by converting the implication operator with the identity

$$X \rightarrow Y \equiv \neg X \vee Y .$$

The probability distribution defined by a PSL program measures the overall distance to satisfaction, that is, the more groundings of rules have high truth values in an interpretation, the more likely it is. More formally, for a PSL program, let G be the set of all groundings for each rule. For any grounding g 2 G, let wg be the weight assigned to the rule, and tg(x) 2 [0, 1] be the grounded rule's truth value under interpretation x. The probability distribution over interpretation x defined by the program is

$$Pr(x; w) = exp(-\sum_{g \in G} w_g(1 - t_g(x)))$$

Considering each grounded rule a factor and each truth value a variable, this probability distribution becomes a log-linear Markov random field over continuous variables. Maximum likelihood inference for the unknown truth values corresponds to solving a linear program, where the truth-value variables are constrained to be consistent with respect to the t-norms and are weighted by rule potentials. Additional details, including a description of a learning algorithm for setting the weights, are provided in (Broecheler, Mihalkova, and Getoor 2010).

Modeling Trust in PSL

Social network theory studies the structural balance of relationships. This section explains the different models used in this paper and the relevant logic behind them. One of the models, for instance, is the same as in Granovetter (Granovetter 1973) social networks tend to exhibit triadic closure, which is loosely the concept that strong relationships are transitive. In the context of trust, this idea translates to how people determine whether to trust others by consulting with those they trust. For example, if Alice strongly trusts Bob, and Bob strongly trusts Chris, then triadic closure implies that Alice will likely trust Chris. Another common idea in analysis of trust is that of reputation, where people who are trusted gain a reputation of being trustworthy, thus garnering more trust (R. Bhattacharya and Pillutla. 1998). Additionally, the qualities of the trustee (i.e.,the person who is trusted) have been identified as important factors for determining trust. For example, whether Alice trusts Bob depends on Bob's beliefs and goals, as well Alice's notions of confidence in Bob. People also have person-specific innate tendencies for trust, which may stem from early-childhood experiences (Castelfranchi and Falcone 2000). Trust is also known to be affected by the similarity in traits of the involved people. In particular, trust exhibits the notion of homophily, a concept from social network theory which suggests that people connect to others with whom they are similar (Cosmides and Tooby 1992). Finally, an important aspect of trust is its context dependency. Trust determines how much individuals value information communicated from each other, so it is natural to consider the level of trust to be a function of the information's topic area. Similarly, trust behavior varies significantly between different relationship types, such as trust between family members, co-workers, or religious group members (Glanville and Paxton 2007). The following section describes the three types of first order logic rules experimented with in this paper and the different types of rules in each of the models, each modeling a different aspect from social theory. The rules fall in three main categories:

- **Triadic rules:** This type of rules have cyclic direction for trust between three users. The figure [Fig.1.] depicts the relationship between A, B and C, as shown in this rule:

  Trusts(A, B) & Trusts(B, C) → Trusts(C, A)

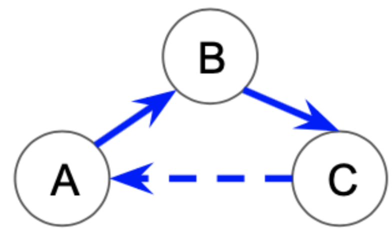

Figure 1: Representation of triadic closure rules

  This can be read as "if A trusts B, and B trusts C, then C trusts A". This model exhibits the triadic closure model explained earlier. The solid lines in the diagram represent a known relationship and the dotted line represents the relationship being predicted.

- **Status / Hierarchical Model:** This model is opposite of the triadic model. This model says "if A trusts B and B trusts C, then C does not trust A" [Fig.2.]. This refers to the hierarchical structure in a professional environment, where C is more senior to A and B, and B is more senior to A. "!" is represent the logical negation symbol ¬ below.

  Trusts(A, B) & Trusts(B, C) → !Trusts(C, A)

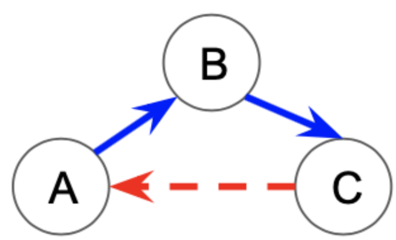

Figure 2: Representation of Status model rules.

- **Non- Cyclic:** This model is non-cyclic in and represents rules like, "if A trusts B and A trust C then C trusts B" [Fig.3]. This rule is given below:

  Trusts(A, B) & Trusts(A, C) → Trusts(C, B)

These rules can be thought of as building blocks which are used individually or in combination for trust modelling in many SRL formalisms. We have used these rules to represent different models of trust in our PSL model. We demonstrate this principle in the context of PSL, allowing us to easily represent degrees of trust and rely on PSL's parameter learning technique to estimate rule weights. We primarily model trust relations with the predicate TRUSTS. A

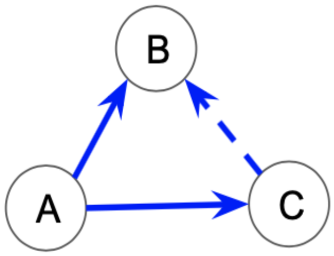

Figure 3: Representation of Non-Cyclic model rules.

soft truth value for TRUSTS(A,B) = 1.0 means that A fully trusts B, while TRUSTS(A,B) = 0.5 means that A somewhat trusts B, and TRUSTS(A,B) = 0 indicates that A does not trust B. Like mentioned in the rules, the first logic we implemented was the triadic closure. The triadic rules were used in four different models, balance5, balance5_recip, balance_extended, balance_extended_recip. The rules in these models are listed in the appendix for reference. All four of these models use the same triadic closure rules but have different numbers of rules in them. The balance5 and balance5_recip models have 5 triadic closure rules whereas the balance_extended and balance_extended_recip models have 16 triadic closure rules. We encode the tendency for transitivity and reciprocity in trust using the triadic rules. The balance5 model has the following rules:

TRUSTS(A,B) & TRUSTS(B,C) → TRUSTS(A,C)
TRUSTS(A,B) & !TRUSTS(B,C) → !TRUSTS(A,C)
!TRUSTS(A,B) & !TRUSTS(B,C) → TRUSTS(A,C)
TRUSTS(A,B) & TRUSTS(A,C) → TRUSTS(B,C)
TRUSTS(A,C) & TRUSTS(B,C) → TRUSTS(A,B)
TRUSTS(A,B) → TRUSTS(B,A) !TRUSTS(A,B) → !TRUSTS(B,A)

These models have a base predicate not expressed in the rules above, called KNOWS (A,B). This is the base blocking predicate for computing the trust value. Thus, in this paper, this predicate depicts that the two users must know each other for being able to predict the trust value. In other statistical relational frameworks, this kind of modelling can get very inefficient with respect to prediction accuracy and/or time. Many frameworks may require discretization of the trust scale, since they might not be able to consider soft values for their predicates. The last two rules in the list shown above are called "prior". These rules are used to model the obvious connection between the nodes (users) or some basic logic on top of which model specific rules can build on. Another natural addition to enforce a consistent status hierarchy suggests the inversion (the models with _inv subscript have the priors shown below) of trust between pairs of individuals. We can represent this with the rules like:

TRUSTS(X, Y) → !TRUSTS(Y,X)
!TRUSTS(X, Y) → TRUSTS(Y,X)

The second social phenomenon we model is basic personality (Huang et al. 2012). This is a latent variable model, which means that the predicate information is derived from the dataset. More specifically, we consider additional predicates TRUSTING and TRUSTWORTHY, modeling whether a person is trusting or trustworthy, respectively. These predicates are not part of the input data, but they correspond to hidden variables that need to be inferred during prediction of trust values. The intuition is that a trusting person is likely to trust more, while a trustworthy person will earn more trust. The model name for this is "personality". The third social phenomenon we model is the effect of similarity on trust. The name for this concept is "similarity". Homophily is the tendency of individuals to associate with others who are similar. The trust ratings people have assigned to one another in our experiments are set in the context of movies (i.e., how much do users trust others' opinions about movies). This makes the movie rating data especially relevant to understanding trust. Previous work has shown that trust in similar social network data is strongly correlated with similarity (Ziegler and Golbeck. 2007). In this PSL model, we consider an additional predicate SAMETRAITS(A,B), which indicates the similarity of A and B according to their personal traits. For example, in our experiments, we measure the similarity of users' survey responses on movie preferences. The intuition here is that people with similar traits tend to trust each other. We additionally consider the idea that people who trust (or do not trust) a particular individual will likely trust (or not trust) those similar to that individual. Conversely, similar people will trust (or not trust) similar sets of trustees.

The fourth type of models are the status models (Huang et al. 2013). This model implements the hierarchical model rules representing a hierarchical structure in the social network. To understand this we can take three users (A,B,C) who are in a triadic trust relationship. For instance, if A, B and C work together in a company and A is junior to B and C is at a higher level to both A and B. In status model rules we would consider a rule like : "If A trusts B and B trusts C, then C does not trust A". This model indirectly uses the "experience" as a parameter for modelling the trust. The status models are the status and status_inv models. The rules from all the models are listed in the model appendix (including the prior rules) for detail reference. The last type of model we modelled were the cyclic_balanced and cyclic_balanced_unbal models. These models use the non-cyclic type of logic rules listed in the previous section. This paper first compared these models and analysed the performance of these two models compared to other baselines (Bach et al. 2015).

Finally, we also study the different combinations of these models like the triad-personality, triad-similarity, personality-similarity, triad-pers-sim. As the name suggests these models are a combination of the basic models like described above. The detailed list of rules in these models are listed in the models appendix. The next section reports on empirical experiments using these models.

Empirical Evaluation

Data sets

For this project, we evaluate the models on two different datasets.

FilmTrust data set: FilmTrust is a web service designed to leverage user-to-user trust values and user-to-movie ratings for movie recommendation (Golbeck 2006a). The dataset consists of a list of anonymized users, their trust values for other users, and their ratings for a set of movies. Since the user trust values are rather sparse, we prune the data to only include the largest connected component of users. Users rate each other on a discrete scale of whole numbers from 1 to 10, which we normalize to [0, 1], making each trust value interpretable as a soft truth value. Similarly, users rate movies with a recommendation rating between 1 and 5. There are 500 users in the largest connected component, among which there are 1574 total user-to-user trust values. The trust values are directed and thus not symmetric. For each pair of users within a two-hop distance, we compute their soft similarity SAMETRAITS via a normalized inner product of their overlapping rated-entry vectors.

Epinions: Average of 8 splits from JMLR-2017 (with Quadratic rules)

Model	Average MAE	Average Spearman Correlation	Average Kendall Correlation	Average AUC-ROC	Average AU-PRC Positive Class	Average AU-PRC Negative Class
status_inv	0.2433	0.1222	0.1034	0.6278	0.9428	0.3097
status	0.1919	0.1612	0.133	0.6708	0.9523	0.3443
balance_extended	0.1298	0.2715	0.2228	0.789	0.9737	0.375
balance5	0.1305	0.2774	0.2267	0.7954	0.9748	0.3774
cyclic_bal_unbal	0.2773	0.15	0.1226	0.6593	0.9396	0.3779
cyclic_balanced	0.1327	0.2711	0.2215	0.7886	0.9741	0.388
balance5_recip	0.1245	0.299	0.2499	0.8161	0.9762	0.3912
balance_extended_recip	0.125	0.2869	0.239	0.804	0.9744	0.3917
triad-personality	0.1219	0.3775	0.3084	0.9017	0.989	0.5647
personality	0.1284	0.3902	0.3188	0.915	0.9894	0.6866

Figure 4: Epinions dataset results (Quadratic Rules)

Epinions: Average of 8 splits from JMLR-2017 (with Linear rules)

Model	Average MAE	Average Spearman Correlation	Average Kendall Correlation	Average AUC-ROC	Average AU-PRC Positive Class	Average AU-PRC Negative Class
status_inv	0.1721	0.1856	0.1714	0.6857	0.9507	0.169
status	0.1675	0.2059	0.1903	0.7059	0.9545	0.178
cyclic_bal_unbal	0.1236	0.1744	0.1573	0.6764	0.9541	0.2196
balance_extended	0.1042	0.22	0.2099	0.706	0.953	0.2501
balance_extended_recip	0.0985	0.2595	0.249	0.7315	0.9563	0.2657
balance5	0.1034	0.2385	0.2262	0.7257	0.9565	0.2689
balance5_recip	0.0953	0.262	0.2522	0.7356	0.9577	0.2707
cyclic_balanced	0.1019	0.2355	0.2232	0.7246	0.957	0.2852
triad-personality	0.0737	0.3265	0.3089	0.753	0.9598	0.4005
personality	0.0791	0.3736	0.3467	0.8211	0.9734	0.416

Figure 5: Epinions dataset results (Linear Rules)

Epinions Dataset: In the Epinions dataset the trust values are directed and thus not symmetric, same as FilmTrust. This dataset is created by snowball sampling a network of 2,000 users from the Epinions data, which contains 8,675 discrete [1, 1] trust scores between users, which we treat as false and true TRUSTS predicate values. The task we consider is collective prediction of trust values given the fully-observed social network. We generate eight folds where, in each fold, 1/8 of the trust values are hidden at random. The prediction algorithm can use the remaining 7/8 of the trust values and the full structure of the social network to learn parameters for a model and perform inference of the unknown

FilmTrust: Average of 8 splits from SBP-2013 (with Quadratic rules)

Model	Average MAE	Average Spearman Correlation	Average Kendall Correlation	Average AUC-ROC	Average AU-PRC Positive Class	Average AU-PRC Negative Class
balance5	0.2103	0.1272	0.0983	0.5554	0.7513	0.3381
balance5_recip	0.2197	0.1142	0.0846	0.5446	0.7435	0.3237
balance_extended	0.2108	0.1504	0.1145	0.5724	0.7592	0.3531
balance_extended_recip	0.2187	0.1427	0.1048	0.5598	0.7525	0.3343
status	0.2258	0.1435	0.1117	0.5324	0.7361	0.3232
status_inv	0.2775	0.0256	0.0206	0.5122	0.7237	0.31
personality	0.1992	0.3148	0.2319	0.6549	0.8164	0.4365
cyclic_balanced	0.2086	0.1486	0.1147	0.5654	0.7558	0.3542
cyclic_bal_unbal	0.2342	0.0606	0.0469	0.5029	0.72	0.2986
triad-personality	0.2023	0.2783	0.2031	0.6416	0.8054	0.4206
similarity	0.2961	-0.0113	-0.0082	0.4852	0.7083	0.2831
triad-similarity	0.2729	0.0248	0.0182	0.5107	0.7358	0.2859
personality-similarity	0.2497	0.1547	0.1129	0.5679	0.7599	0.343
triad-pers-sim	0.2436	0.1445	0.1048	0.5708	0.7655	0.3444

Figure 6: FilmTrust dataset results (Quadratic Rules)

FilmTrust: Average of 8 splits from SBP-2013 (with Linear rules)

Model	Average MAE	Average Spearman Correlation	Average Kendall Correlation	Average AUC-ROC	Average AU-PRC Positive Class	Average AU-PRC Negative Class
balance5	0.2143	0.1159	0.0897	0.5487	0.7429	0.3355
balance5_recip	0.229	0.0999	0.0744	0.5419	0.7449	0.3182
balance_extended	0.2116	0.1532	0.1168	0.5671	0.7546	0.3478
balance_extended_recip	0.2288	0.1232	0.0916	0.5498	0.7444	0.3282
status	0.2278	0.1456	0.1138	0.5342	0.7387	0.3231
status_inv	0.3204	0.0228	0.0174	0.5137	0.7292	0.2955
personality	0.2024	0.1996	0.1442	0.5974	0.7863	0.3706
cyclic_balanced	0.2109	0.1414	0.109	0.5615	0.7486	0.3469
cyclic_bal_unbal	0.2367	0.041	0.0317	0.496	0.7188	0.2931
triad-personality	0.2037	0.1857	0.1337	0.5877	0.7806	0.3679
similarity	0.2098	-0.0074	-0.0048	0.4854	0.7102	0.2937
triad-similarity	0.2152	0.0989	0.0753	0.5386	0.7387	0.3217
personality-similarity	0.2104	0.1961	0.1414	0.6041	0.7864	0.3879
triad-pers-sim	0.2112	0.2153	0.1555	0.6218	0.7924	0.4032

Figure 7: FilmTrust dataset results (Linear Rules)

trust values. For example, PSL learns weights for the rules in each given model from these observed trust values.

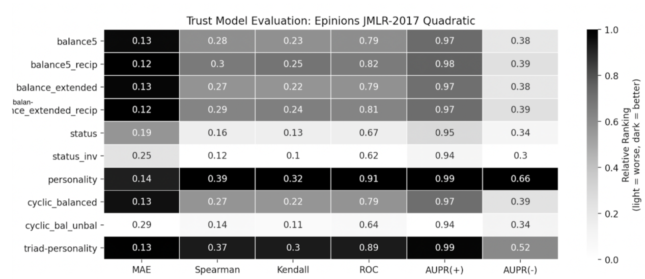

Figure 8: Epinions dataset heatmap results (Quadratic Rules)

Results and Discussion

On both the datasets, we measure for each algorithm the average score over the eight folds for three metrics: mean average error (MAE), Kendall's (tau) statistic, and Spearman's rank correlation , average AU-ROC, average AU-PR (both positive and negative class). MAE measures the absolute error on the soft truth-values, while and measure ranking performance. The ROC and AUPR values give an idea of the classification accuracy score. The average scores are listed in Table 1-4. Since the Epinions dataset does not have the similarity information the model set for both the datasets is different. The result tables show the difference in the models

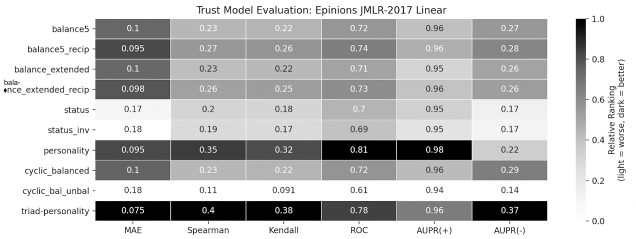

Figure 9: Epinions dataset heatmap results (Linear Rules)

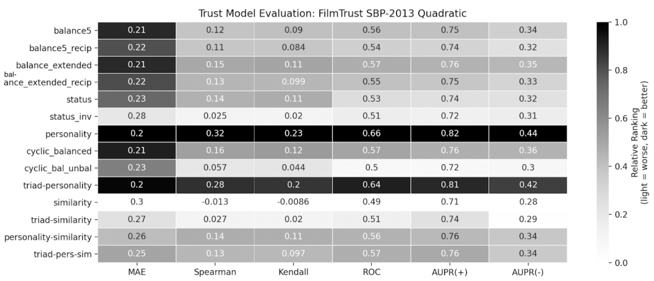

Figure 10: FilmTrust dataset heatmap results (Quadratic Rules)

used for the two datasets. The standard deviation has been calculated for all the metrics, to get a sense of the error graph across the dataset along with knowing the averages.

The runtimes for the Epinions datasets, including all 8 splits and all 10 models, is 10.01 mins. The runtime for the FilmTrust dataset, including all 8 dataset splits and for 14 different models, is 25.03 mins.

In PSL, any weighted rule can be chosen to square their hinge-loss functions. Squaring the hinge-loss (or "squared potentials") can result in better performance, as seen in the experiments in this paper as well. Non-squared potentials tend to encourage a "winner take all" optimization, while squared potentials encourage more trading off. The results with squared rules is better compared to the linear rules. The weights for the learned rule models are more accurate and multifarious when squared as opposed to the linear rule models.

The big table with so many models and evaluation parameters is difficult to read and analyse. So we use heatmaps to understand which model(s) performs the best in all (or most) categories. The heat maps are shown in figures (put numbers here). The models are relatively ranked in the heatmaps, where for each column (evaluation parameter) is colored light to dark, representing the worst to best evaluation parameter for that column. All the evaluation parameters (except MAE) are higher the better. The cell values represent the evaluation parameter values same as in the tables.

These heatmaps give us a better understanding of the comparison between the different models and helps us see which models perform better than the others. The four heatmaps show that the triad-pers, triad-pers-sim and the personality models are overall the best performing models and have dark rows in all four heatmaps. The personality

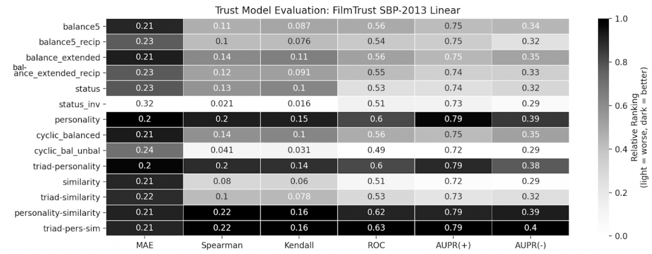

Figure 11: FilmTrust dataset heatmap results (Linear Rules)

model seems to perform the best amongst these top three. Since personality model is a latent model and using only latent rules does not give us as much information about the model as compared to the basic models, we believe that the best model is the triad-personality model for the Epinions dataset and triad-pers-sim model for the filmTrust dataset.

Conclusions and Future Work

This paper explores different PSL models and compares the evaluation metrics of these models, to be able to model trust in social networks. PSL framework allows for easy exploration of trust models based on different assumptions about social phenomena. To demonstrate the effectiveness of PSL models for this task we compare the effectiveness of these models on two different dataset and across a variety of social phenomena and rule models. We also explore additional models by adding different kinds of base models to build composite models. We explore the results over multiple data splits and analyse both the average and the standard deviation of the evaluation parameters to be able to get a holistic view of the model performance. We also explore other PSL parameters like using weight learning and choosing to square their hinge-loss functions of PSL weighted rules. This gives us a richer analysis of the weights of different rules in the models and helps us to remove unnecessary rules from a model.

A lot of more study is possible in this field. These models can be used on other datasets with different data types. If more information is available in the dataset with different kinds of social phenomenon more rules can be added to existing models to give better results. Same models can be applied to different datasets with different contexts too. Since trust can be morphed as friendships, professional relationships, etc. these models can be really useful in recommender systems for social media platforms with different kinds of user networks.

Appendix

The list of rules for each of the models in the paper is given below.

- **balance5:**
  Trusts(A, B) Trusts(B, C) → Trusts(A, C)
  Trusts(A, B) & !Trusts(B, C) → !Trusts(A, C)
  !Trusts(A, B) & !Trusts(B, C) → Trusts(A, C)
  Trusts(A, B) & Trusts(C, B) → Trusts(A, C)

Trusts(B, A) & Trusts(B, C) → Trusts(A, C)
two-sided prior →
Knows(A, B) & Prior('0') → Trusts(A, B)
Knows(A, B) & Trusts(A, B) → Prior('0')

- **balance5_recip:**
Trusts(A, B) & Trusts(B, C) → Trusts(A, C)
Trusts(A, B) & !Trusts(B, C) → !Trusts(A, C)
!Trusts(A, B) & !Trusts(B, C) → Trusts(A, C)
Trusts(A, B) & Trusts(C, B) → Trusts(A, C)
Trusts(B, A) & Trusts(B, C) → Trusts(A, C)
two-sided prior:
Knows(A, B) & Prior('0') → Trusts(A, B)
Knows(A, B) & Trusts(A, B) → Prior('0')
Trusts(A, B) → Trusts(B, A)
!Trusts(A, B) → !Trusts(B, A)

- **balance_extended :**
Trusts(A, B) & Trusts(B, C) → Trusts(A, C)
Trusts(A, B) & !Trusts(B, C) → !Trusts(A, C)
!Trusts(A, B) & !Trusts(B, C) → Trusts(A, C)
Trusts(A, B) & Trusts(C, B) → Trusts(A, C)
Trusts(B, A) & Trusts(B, C) → Trusts(A, C)
two-sided prior
Knows(A, B) & Prior('0') → Trusts(A, B)
Knows(A, B) & Trusts(A, B) → Prior('0')
!Trusts(A, B) & Trusts(B, C) → !Trusts(A, C)
Trusts(A, B) & !Trusts(C, B) → !Trusts(A, C)
!Trusts(A, B) & Trusts(C, B) → !Trusts(A, C)
!Trusts(A, B) & !Trusts(C, B) → Trusts(A, C)
Trusts(B, A) & !Trusts(B, C) → !Trusts(A, C)
!Trusts(B, A) & Trusts(B, C) → !Trusts(A, C)
!Trusts(B, A) & !Trusts(B, C) → Trusts(A, C)
Trusts(B, A) & Trusts(C, B) → Trusts(A, C)
Trusts(B, A) & !Trusts(C, B) → !Trusts(A, C)
!Trusts(B, A) & Trusts(C, B) → !Trusts(A, C)
!Trusts(B, A) & !Trusts(C, B) → Trusts(A, C)

- **balance_extended_recip:**
Trusts(A, B) & Trusts(B, C) → Trusts(A, C)
Trusts(A, B) & !Trusts(B, C) → !Trusts(A, C)
!Trusts(A, B) & !Trusts(B, C) → Trusts(A, C)
Trusts(A, B) & Trusts(C, B) → Trusts(A, C)
Trusts(B, A) & Trusts(B, C) → Trusts(A, C)
!Trusts(A, B) & Trusts(B, C) → !Trusts(A, C)
Trusts(A, B) & !Trusts(C, B) → !Trusts(A, C)
!Trusts(A, B) & Trusts(C, B) → !Trusts(A, C)
!Trusts(A, B) & !Trusts(C, B) → Trusts(A, C)
Trusts(B, A) & !Trusts(B, C) → !Trusts(A, C)
!Trusts(B, A) & Trusts(B, C) → !Trusts(A, C)
!Trusts(B, A) & !Trusts(B, C) → Trusts(A, C)
Trusts(B, A) & Trusts(C, B) → Trusts(A, C)
Trusts(B, A) & !Trusts(C, B) → !Trusts(A, C)
!Trusts(B, A) & Trusts(C, B) → !Trusts(A, C)
!Trusts(B, A) & !Trusts(C, B) → Trusts(A, C)
two-sided prior
Knows(A, B) & Prior('0') → Trusts(A, B)
Knows(A, B) & Trusts(A, B) → Prior('0')
Trusts(A, B) → Trusts(B, A)
!Trusts(A, B) → !Trusts(B, A)

- **status :**
Trusts(A, B) & Trusts(B, C) → Trusts(A, C)
!Trusts(A, B) & !Trusts(B, C) → !Trusts(A, C)
Trusts(A, B) & !Trusts(C, B) → Trusts(A, C)
!Trusts(A, B) & Trusts(C, B) → !Trusts(A, C)
Trusts(B, A) & !Trusts(B, C) → !Trusts(A, C)
!Trusts(B, A) & Trusts(B, C) → Trusts(A, C)
Trusts(B, A) & Trusts(C, B) → !Trusts(A, C)
!Trusts(B, A) & !Trusts(C, B) → Trusts(A, C)
two-sided prior
Knows(A, B) & Prior('0') → Trusts(A, B)
Knows(A, B) & Trusts(A, B) → Prior('0')

- **status_inv :**
Trusts(A, B) & Trusts(B, C) → Trusts(A, C)
!Trusts(A, B) & !Trusts(B, C) → !Trusts(A, C)
Trusts(A, B) & !Trusts(C, B) → Trusts(A, C)
!Trusts(A, B) & Trusts(C, B) → !Trusts(A, C)
Trusts(B, A) & !Trusts(B, C) → !Trusts(A, C)
!Trusts(B, A) & Trusts(B, C) → Trusts(A, C)
Trusts(B, A) & Trusts(C, B) → !Trusts(A, C)
!Trusts(B, A) & !Trusts(C, B) → Trusts(A, C)
two-sided prior
Knows(A, B) Prior('0') → Trusts(A, B)
Knows(A, B) Trusts(A, B) → Prior('0')
Inverse rules
Trusts(A, B) → Trusts(B, A)
!Trusts(A, B) → !Trusts(B, A)

- **personality :**
Trusts(A, B) → TrustWorthy(B)
Trusts(A, B) → Trusting(A)
!Trusts(A, B) → !TrustWorthy(B)
!Trusts(A, B) → !Trusting(A)
Trusting(A) & TrustWorthy(B) → Trusts(A, B)
!Trusting(A) & !TrustWorthy(B) → !Trusts(A, B)
two-sided prior
Knows(A, B) & Prior('0') -¿ Trusts(A, B)
Knows(A, B) & Trusts(A, B) -¿ Prior('0')

- **cyclic_balanced**
Trusts(A, B) & Trusts(B, C) → Trusts(C, A)
!Trusts(A, B) & !Trusts(B, C) → Trusts(C, A)
Trusts(A,B) & Trusts(A,C) → Trusts(C,B)
Trusts(A,B) & !Trusts(A,C) → !Trusts(C,B)

!Trusts(A,B) & Trusts(A,C) → !Trusts(C,B)
!Trusts(A,B) & !Trusts(A,C) → Trusts(C,B)
two-sided prior
Knows(A, B) & Prior('0') → Trusts(A, B)
Knows(A, B) & Trusts(A, B) → Prior('0')

- **cyclic_bal_unbal :**
  Trusts(A, B) & Trusts(B, C) → Trusts(C, A)
  !Trusts(A, B) & !Trusts(B, C) → Trusts(C, A)
  Trusts(A,B) & Trusts(A,C) → Trusts(C,B)
  Trusts(A,B) & !Trusts(A,C) → !Trusts(C,B)
  !Trusts(A,B) & Trusts(A,C) → !Trusts(C,B)
  !Trusts(A,B) & !Trusts(A,C) → Trusts(C,B)
  Trusts(A, B) & !Trusts(B, C) → Trusts(C, A)
  Trusts(A, B) & Trusts(B, C) → !Trusts(C, A)
  Trusts(A,B) & Trusts(A,C) → !Trusts(C,B)
  Trusts(A,B) & !Trusts(A,C) → Trusts(C,B)
  !Trusts(A,B) & Trusts(A,C) → Trusts(C,B)
  !Trusts(A,B) & !Trusts(A,C) → !Trusts(C,B)
  two-sided prior
  Knows(A, B) & Prior('0') → Trusts(A, B)
  Knows(A, B) & Trusts(A, B) → Prior('0')
- **triad-personality :**
  Trusts(A, B) & Trusts(B, C) → Trusts(A, C)
  Trusts(A, B) & !Trusts(B, C) → !Trusts(A, C)
  !Trusts(A, B) & !Trusts(B, C) → Trusts(A, C)
  Trusts(A, B) & Trusts(C, B) → Trusts(A, C)
  Trusts(B, A) & Trusts(B, C) → Trusts(A, C)
  two-sided prior
  Knows(A, B) & Prior('0') → Trusts(A, B)
  Knows(A, B) & Trusts(A, B) → Prior('0')
  Trusts(A, B) → TrustWorthy(B)
  Trusts(A, B) → Trusting(A)
  !Trusts(A, B) → !TrustWorthy(B)
  !Trusts(A, B) → !Trusting(A)
  Trusting(A) & TrustWorthy(B) → Trusts(A, B)
  !Trusting(A) & !TrustWorthy(B) → !Trusts(A, B)
- **similarity:**
  SameTastes(A, B) → Trusts(A, B)
  !SameTastes(A, B) → !Trusts(A, B)
  Trusts(A, B) & SameTastes(B, C) → Trusts(A, C)
  !Trusts(A, B) & SameTastes(B, C) → !Trusts(A, C)
  Trusts(A, C) & SameTastes(A, B) → Trusts(B, C)
  !Trusts(A, C) & SameTastes(A, B) → !Trusts(B, C)
- **triad-similarity:**
  Trusts(A, B) & Trusts(B, C) → Trusts(A, C)
  Trusts(A, B) & !Trusts(B, C) → !Trusts(A, C)
  !Trusts(A, B) & !Trusts(B, C) → Trusts(A, C)
  Trusts(A, B) & Trusts(C, B) → Trusts(A, C)
  Trusts(B, A) & Trusts(B, C) → Trusts(A, C)
  two-sided prior
  Knows(A, B) & Prior('0') → Trusts(A, B)
  Knows(A, B) & Trusts(A, B) → Prior('0')
  SameTastes(A, B) → Trusts(A, B)
  !SameTastes(A, B) → !Trusts(A, B)
  Trusts(A, B) & SameTastes(B, C) → Trusts(A, C)
  !Trusts(A, B) & SameTastes(B, C) → !Trusts(A, C)
  Trusts(A, C) & SameTastes(A, B) → Trusts(B, C)
  !Trusts(A, C) & SameTastes(A, B) → !Trusts(B, C)
- **personality-similarity:**
  Trusts(A, B) → TrustWorthy(B)
  Trusts(A, B) → Trusting(A)
  !Trusts(A, B) → !TrustWorthy(B)
  !Trusts(A, B) → !Trusting(A)
  Trusting(A) & TrustWorthy(B) → Trusts(A, B)
  !Trusting(A) & !TrustWorthy(B) → !Trusts(A, B)
  two-sided prior
  Knows(A, B) & Prior('0') → Trusts(A, B)
  Knows(A, B) & Trusts(A, B) → Prior('0')
  SameTastes(A, B) → Trusts(A, B)
  !SameTastes(A, B) → !Trusts(A, B)
  Trusts(A, B) & SameTastes(B, C) → Trusts(A, C)
  !Trusts(A, B) & SameTastes(B, C) → !Trusts(A, C)
  Trusts(A, C) & SameTastes(A, B) → Trusts(B, C)
  !Trusts(A, C) & SameTastes(A, B) → !Trusts(B, C)
- **triad-pers-sim:**
  Trusts(A, B) & Trusts(B, C) → Trusts(A, C)
  Trusts(A, B) & !Trusts(B, C) → !Trusts(A, C)
  !Trusts(A, B) & !Trusts(B, C) → Trusts(A, C)
  Trusts(A, B) & Trusts(C, B) → Trusts(A, C)
  Trusts(B, A) & Trusts(B, C) → Trusts(A, C)
  Trusts(A, B) → TrustWorthy(B)
  Trusts(A, B) → Trusting(A)
  !Trusts(A, B) → !TrustWorthy(B)
  !Trusts(A, B) → !Trusting(A)
  Trusting(A) & TrustWorthy(B) → Trusts(A, B)
  !Trusting(A) & !TrustWorthy(B) → !Trusts(A, B)
  two-sided prior
  Knows(A, B) & Prior('0') → Trusts(A, B)
  Knows(A, B) & Trusts(A, B) → Prior('0')
  SameTastes(A, B) → Trusts(A, B)
  !SameTastes(A, B) → !Trusts(A, B)
  Trusts(A, B) & SameTastes(B, C) → Trusts(A, C)
  !Trusts(A, B) & SameTastes(B, C) → !Trusts(A, C)
  Trusts(A, C) & SameTastes(A, B) → Trusts(B, C)
  !Trusts(A, C) & SameTastes(A, B) → !Trusts(B, C)

All rules have a "Knows(X,Y) blocking predicate for every any predicate with two nodes). For example, Trusts(X,Y). The extended result tables with the standard deviations is shown below.

The extended result tables (including the standard deviation) are given below :

Epinions: Average of 8 splits from JMLR-2017 (with Quadratic rules)

Model	Average MAE	MAE (STD)	Average Spearman Correlation	Spearman Correlation (STD)	Average Kendall Correlation	Kendall Correlation (STD)	Average AUC-ROC	AUC-ROC (STD)	Average AU-PRC Positive Class	AU-PRC Positive Class (STD)	Average AU-PRC Negative Class	AU-PRC Negative Class (STD)
status_inv	0.2433	0.0062	0.1222	0.0262	0.1034	0.0229	0.6278	0.0222	0.9428	0.01	0.3097	0.034
status	0.1919	0.0078	0.1612	0.0288	0.133	0.0229	0.6708	0.0267	0.9523	0.0093	0.3443	0.0336
balance_extended	0.1298	0.009	0.2715	0.0253	0.2228	0.0202	0.789	0.0249	0.9737	0.0067	0.375	0.049
balance5	0.1305	0.009	0.2774	0.0231	0.2267	0.0189	0.7954	0.0201	0.9748	0.0062	0.3774	0.0434
cyclic_bal_unbal	0.2773	0.0079	0.15	0.0292	0.1226	0.0239	0.6593	0.0279	0.9396	0.0126	0.3779	0.0571
cyclic_balanced	0.1327	0.0085	0.2711	0.0224	0.2215	0.0183	0.7886	0.0187	0.9741	0.0062	0.388	0.0502
balance5_recip	0.1245	0.0094	0.299	0.0275	0.2499	0.0253	0.8161	0.0228	0.9762	0.0056	0.3912	0.0441
balance_extended_recip	0.125	0.0094	0.2869	0.0253	0.239	0.0229	0.804	0.0254	0.9744	0.007	0.3917	0.0441
triad-personality	0.1219	0.0082	0.3775	0.0275	0.3084	0.0224	0.9017	0.0133	0.989	0.0026	0.5647	0.0398
personality	0.1284	0.0073	0.3902	0.031	0.3188	0.0253	0.915	0.0146	0.9894	0.0031	0.6866	0.0383

Figure 12: Epinions dataset heatmap results extended (Quadratic Rules)

Epinions: Average of 8 splits from JMLR-2017 (with Linear rules)

Model	Average MAE	MAE (STD)	Average Spearman Correlation	Spearman Correlation (STD)	Average Kendall Correlation	Kendall Correlation (STD)	Average AUC-ROC	AUC-ROC (STD)	Average AU-PRC Positive Class	AU-PRC Positive Class (STD)	Average AU-PRC Negative Class	AU-PRC Negative Class (STD)
status_inv	0.1721	0.0106	0.1856	0.0235	0.1714	0.0215	0.6857	0.0135	0.9507	0.0059	0.169	0.0249
status	0.1675	0.0092	0.2059	0.0263	0.1903	0.0237	0.7059	0.0158	0.9545	0.0051	0.178	0.0306
cyclic_bal_unbal	0.1236	0.0094	0.1744	0.0413	0.1573	0.0362	0.6764	0.0372	0.9541	0.0067	0.2196	0.0603
balance_extended	0.1042	0.0111	0.22	0.0321	0.2099	0.0307	0.706	0.0288	0.953	0.0094	0.2501	0.0467
balance_extended_recip	0.0985	0.011	0.2595	0.0234	0.249	0.0219	0.7315	0.0181	0.9563	0.0079	0.2657	0.0361
balance5	0.1034	0.0096	0.2385	0.0376	0.2262	0.0338	0.7257	0.0304	0.9565	0.0079	0.2689	0.0322
balance5_recip	0.0953	0.0096	0.262	0.0293	0.2522	0.027	0.7356	0.0205	0.9577	0.007	0.2707	0.0287
cyclic_balanced	0.1019	0.0085	0.2355	0.0298	0.2232	0.0275	0.7246	0.0241	0.957	0.0079	0.2852	0.0538
triad-personality	0.0737	0.0081	0.3265	0.0421	0.3089	0.0393	0.753	0.032	0.9598	0.0085	0.4005	0.0375
personality	0.0791	0.0077	0.3736	0.0363	0.3467	0.0341	0.8211	0.0247	0.9734	0.0077	0.416	0.0623

Figure 13: Epinions dataset heatmap results extended (Linear Rules)

FilmTrust: Average of 8 splits from SBP-2013 (with Quadratic rules)

Model	Average MAE	MAE (STD)	Average Spearman Correlation	Spearman Correlation (STD)	Average Kendall Correlation	Kendall Correlation (STD)	Average AUC-ROC	AUC-ROC (STD)	Average AU-PRC Positive Class	AU-PRC Positive Class (STD)	Average AU-PRC Negative Class	AU-PRC Negative Class (STD)
balance5	0.2103	0.008	0.1272	0.0555	0.0983	0.042	0.5554	0.0251	0.7513	0.0219	0.3381	0.0197
balance5_recip	0.2197	0.0074	0.1142	0.047	0.0846	0.0342	0.5446	0.0458	0.7435	0.043	0.3237	0.0417
balance_extended	0.2108	0.007	0.1504	0.0428	0.1145	0.0323	0.5724	0.0196	0.7592	0.0147	0.3531	0.0285
balance_extended_recip	0.2187	0.0066	0.1427	0.0456	0.1048	0.0325	0.5598	0.0463	0.7525	0.0408	0.3343	0.0442
status	0.2258	0.0092	0.1435	0.0452	0.1117	0.0335	0.5324	0.0404	0.7361	0.0326	0.3232	0.0201
status_inv	0.2775	0.0142	0.0256	0.0536	0.0206	0.0396	0.5122	0.0491	0.7237	0.0384	0.31	0.038
personality	0.1992	0.0089	0.3148	0.0408	0.2319	0.0299	0.6549	0.0178	0.8164	0.0224	0.4365	0.0271
cyclic_balanced	0.2086	0.0077	0.1486	0.057	0.1147	0.0446	0.5654	0.0285	0.7558	0.0093	0.3542	0.0313
cyclic_bal_unbal	0.2342	0.0064	0.0606	0.0484	0.0469	0.0363	0.5029	0.0351	0.72	0.0236	0.2986	0.0244
triad-personality	0.2023	0.0083	0.2783	0.0306	0.2031	0.0218	0.6416	0.0202	0.8054	0.0251	0.4206	0.0211
similarity	0.2961	0.0146	-0.0113	0.0656	-0.0082	0.047	0.4852	0.0542	0.7083	0.0364	0.2831	0.0395
triad-similarity	0.2729	0.013	0.0248	0.0547	0.0182	0.0392	0.5107	0.0396	0.7358	0.0375	0.2859	0.0181
personality-similarity	0.2497	0.0118	0.1547	0.0432	0.1129	0.0315	0.5679	0.0297	0.7599	0.0357	0.343	0.0261
triad-pers-sim	0.2436	0.0116	0.1445	0.0429	0.1048	0.0308	0.5708	0.0257	0.7655	0.0337	0.3444	0.0246

Figure 14: FilmTrust dataset heatmap results extended (Quadratic Rules)

FilmTrust: Average of 8 splits from SBP-2013 (with Linear rules)

Model	Average MAE	MAE (STD)	Average Spearman	Spearman Correlation (STD)	Average Kendall Correlation	Kendall Correlation (STD)	Average AUC-ROC	AUC-ROC (STD)	Average AU-PRC Positive Class	AU-PRC Positive Class (STD)	Average AU-PRC Negative Class	AU-PRC Negative Class (STD)
balance5	0.2143	0.0083	0.1159	0.0435	0.0897	0.0352	0.5487	0.0212	0.7429	0.0213	0.3355	0.022
balance5_recip	0.229	0.0112	0.0999	0.05	0.0744	0.0369	0.5419	0.0458	0.7449	0.0437	0.3182	0.0385
balance_extended	0.2116	0.0083	0.1532	0.0644	0.1168	0.0499	0.5671	0.0316	0.7546	0.016	0.3478	0.0304
balance_extended_recip	0.2288	0.0098	0.1232	0.0542	0.0916	0.0396	0.5498	0.0413	0.7444	0.0392	0.3282	0.0361
status	0.2278	0.0108	0.1456	0.0639	0.1138	0.0489	0.5342	0.0478	0.7387	0.0368	0.3231	0.0184
status_inv	0.3204	0.0169	0.0228	0.0607	0.0174	0.0453	0.5137	0.0565	0.7292	0.0427	0.2955	0.0441
personality	0.2024	0.0091	0.1996	0.049	0.1442	0.0354	0.5974	0.0231	0.7863	0.0314	0.3706	0.043
cyclic_balanced	0.2109	0.0083	0.1414	0.0516	0.109	0.0396	0.5615	0.0253	0.7486	0.019	0.3469	0.0221
cyclic_bal_unbal	0.2367	0.0062	0.041	0.0355	0.0317	0.0261	0.496	0.0372	0.7188	0.0267	0.2931	0.0257
triad-personality	0.2037	0.0085	0.1857	0.0641	0.1337	0.0454	0.5877	0.0326	0.7806	0.0329	0.3679	0.0393
similarity	0.2098	0.0081	-0.0074	0.072	-0.0048	0.0533	0.4854	0.0547	0.7102	0.0347	0.2937	0.0377
triad-similarity	0.2152	0.0076	0.0989	0.0499	0.0753	0.0391	0.5386	0.0259	0.7387	0.0227	0.3217	0.0292
personality-similarity	0.2104	0.0094	0.1961	0.0537	0.1414	0.0412	0.6041	0.0332	0.7864	0.0379	0.3879	0.0413
triad-pers-sim	0.2112	0.008	0.2153	0.0441	0.1555	0.0314	0.6218	0.0289	0.7924	0.0294	0.4032	0.0251

Figure 15: FilmTrust dataset results extended (Linear Rules)

References

A. Josang, R. I., and Boyd, C. 2007. A survey of trust and reputation systems for online service provision. *Decis. Support Syst.* 43(2):618–644.

Abdullah, N. A.; Nishioka, D.; Tanaka, Y.; and Murayama, Y. 2017. Why i retweet? exploring user's perspective on decision-making of information spreading during disasters. *Hawaii Int. Conf. Syst. Sci.* 50:1–10.

B. Abu-Salih, K. Y. Chan, O. A.-K. M. A.-T. P. W. T. I. H. S. M. A.-H. B. B., and Albahlal, A. 2020. Time - aware domain-based social influence prediction. *J. Big Data* 7(1):10–20.

Bach, S.; Huang, B.; Boyd-Graber, J.; and Getoor, L. 2015. Paired-dual learning for fast training of latent variable hinge-loss mrfs. In *International Conference on Machine Learning (ICML)*. Lille, France: PMLR.

Bach, S. H.; Broecheler, M.; Huang, B.; and Getoor, L. 2017. Hinge-loss markov random fields and probabilistic soft logic. volume 18, 1–67.

Bild, D. R.; Liu, Y.; Dick, R. P.; Mao, Z. M.; and Wallach, D. S. 2015. Aggregate characterization of user behavior in twitter and analysis of the retweet graph. *ACM Trans. Internet Technology* 15(1):1–24.

Broecheler, M.; Mihalkova, L.; and Getoor, L. 2010. Probabilistic similarity logic. In *Conference on Uncertainty in Artificial Intelligence (UAI)*. Catalina Island, CA, USA: AUAI.

Calio, A., and Tagarelli, A. 2019. Complex influence propagation based on trust-aware dynamic linear threshold models. *Appl. Netw. Sci.* 4(1):14–24.

Castelfranchi, C., and Falcone, R. 2000. Trust is much more than subjective probability: Mental components and sources of trust. *International Conference on System Sciences (HICSS)*.

Chen, L.-J., and Gao, J. 2018. A trust-based recommendation method using network diffusion processes. *Physica A: Statistical Mechanics and its Applications* 506.

Cosmides, L., and Tooby, J. 1992. Cognitive adaptations for social exchange. the adapted mind: Evolutionary psychology and the generation of culture. *Oxford University Press*.

G. Liu, Y. Liu, A. L. Z. L.-K. Z. Y. W., and Zhou, X. 2018. Context-aware trust network extraction in large-scale trust-oriented social networks. *World Wide Web* 21(3):713–738.

Ghafari, S. M.; Beheshti, A.; Yakhchi, S.; and Orgun, M. 2018. Social context aware trust prediction: A method for identifying fake news. *WISE* 19:161–177.

Glanville, J., and Paxton, P. 2007. How do we learn to trust? a confirmatory tetrad analysis of the sources of generalized trust. *Social Psychology Quarterly* 70(3):230–242.

Golbeck, J., H. J. 2006a. Filmtrust: Movie recommendations using trust in web-based social networks. *ACM Transactions on Internet Technology* 6(4):497–529.

Golbeck, J. 2006b. Generating predictive movie recommendations from trust in social networks. In *Trust Management*, 93–104. Berlin, Heidelberg: Springer Berlin Heidelberg.

Granovetter, M. 1973. The strength of weak ties. *The American Journal of Sociology* 78(6):1360–1380.

Guha, R.; Kumar, R.; Raghavan, P.; and Tomkins, A. 2004. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web*, 403–412.

H. Liu, E.-P. Lim, H. W. L.-M.-T. L. A. S. J. S., and Kim, Y. A. 2008. Predicting trusts among users of online communities: An epinions case study. *ACM Conf. Electron. Commerce EC* 310–319.

Huang, B.; Kimmig, A.; Getoor, L.; and Golbeck, J. 2012. Probabilistic soft logic for trust analysis in social networks. In *International Workshop on Statistical Relational AI (StarAI)*. Catalina Island, CA, USA: StarAI.

Huang, B.; Kimmig, A.; Getoor, L.; and Golbeck, J. 2013. A flexible framework for probabilistic models of social trust. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction (SBP)*. Washington, DC, USA: Springer-Verlag.

Kim, Y. A., and Phalak, R. 2012. A trust prediction framework in rating-based experience sharing social networks without a web of trust. *Information Sciences* 191:128–145.

Li, Z.; Zhang, X.; Shen, H.; Liang, W.; and He, Z. 2015. A semi-supervised framework for social spammer detection. *Advances in Knowledge Discovery and Data Mining, PAKDD, Springer* 177–188.

Liu, S.; Zhang, L.; and Yan, Z. 2018. Predict pairwise trust based on machine learning in online social networks: A survey. *IEEE Access* 6:51297–51318.

M. Graña, J. D. Nuñez-Gonzalez, L. O., and Kamińska-Chuchmała, A. 2015. Experiments of trust prediction in social networks by artificial neural networks. *Cybern. Syst.* 46(1-2):19–34.

Ma, X.; Lu, H.; and Gan, Z. 2015. Implicit trust and distrust prediction for recommender systems. *Int. Conf. Web Inf. Syst. Eng.* 185–199.

Matsuo, Y., and Yamamoto, H. 2009. Community gravity: Measuring bidirec- tional effects by trust and rating on online social networks. *Int. Conf. World Wide Web WWW* 18:751–760.

Mayer, R. C.; Davis, J. H.; and Schoorman, F. D. 1995. An integrative model of organizational trust. *The Academy of Management Review* 20(3):709–734.

N. Ma, E.-P. Lim, V.-A. N.-A. S., and Liu, H. 2009. Trust relation- ship prediction using online product review data. *CNIKM* 47–54.

Nuñez-Gonzalez, J. D., and Manuel Graña, B. A. 2015. Reputation features for trust prediction in social networks. *Neurocomputing* 166:1–7.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking : Bringing order to the web. In *WWW 1999*.

R. Bhattacharya, T. D., and Pillutla., M. 1998. A formal model of trust based on outcomes. *The Academy of Management Review,* 23(3):459–472.

S. M. Ghafari, A. Joshi, A. B. C. P. S. Y., and Orgun, M. 2019. Dcat: A deep context-aware trust prediction approach for online social networks. *Int. Conf. Adv. Mobile Comput. Multimedia (ICAMCM)* 17:20–27.

Tang, J., and Liu, H. 2015. Trust social media. *Morgan Claypool*.

Tang, J.; Gao, H.; Hu, X.; and Liu, H. 2013. Exploiting homophily effect for trust prediction. *ACM Int. Conf. Web Search Data Mining WSDM* 53–62.

Tang, J.; Gao, H.; and Liu, H. 2012. Mtrust: Discerning multi-faceted trust in a connected world. *ACM Int. Conf. Web Search Data Mining WSDM* 93–102.

W. Sherchan, S. N., and Paris, C. 2013. A survey of trust in social net- works. *ACM Comput. Surv.* 45(4):1–33.

2017. 2017(8):083405.

Wang, X.; Zhang, Z.; Wang, J.; Cui, P.; and Yang, S. 2018a. Power-law distribution aware trust prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 3564–3570. International Joint Conferences on Artificial Intelligence Organization.

Wang, Y.; Chen, I.; Cho, J.; Swami, A.; Lu, Y.; Lu, C.; and Tsai, J. J. P. 2018b. Catrust: Context-aware trust management for service-oriented ad hoc networks. *IEEE Transactions on Services Computing* 11(6):908–921.

Wang, X.; Wang, Y.; and Sun, H. 2016. Exploring the combination of dempster-shafer theory and neural network for predicting trust and distrust. *Computational Intelligence and Neuroscience* 2016:1–12.

Y. Wang, X. Wang, J. T. W. Z., and Cai, G. 2015. Modeling status theory in trust prediction. *AAAI Conf. Artif. Intell.* 29:1875–1881.

Y. Wang, L. L., and Liu, G. 2015. Social context-aware trust inference for trust enhancement in social network based recommendations on service providers. *World Wide Web* 18(1):159–184.

Yu, Y.; Gao, Y.; Wang, H.; and Wang, R. 2016. Joint user knowledge and matrix factorization for recommender systems. *Int. Conf. Web Inf. Syst. Eng. WISE* 17:77–91.

Zhao, K., and Pan, L. 2014. A machine learning based trust evaluation framework for online social networks. In *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, 69–74.

Zheng, X.; Wang, Y.; Orgun, M. A.; Liu, G.; and Zhang, H. 2014. Social context-aware trust prediction in social networks. *ICSOC* 527–534.

Ziegler, C.-N., and Golbeck., J. 2007. Investigating interactions of trust and interest similarity. *Decis. Support Syst.* 43(2):460–475.