Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)

How Fast Will You Get a Response? Predicting Interval Time for Reciprocal Link Creation

Vachik S. Dave, Mohammad Al Hasan
Department of Computer & Information Science
IUPUI, Indianapolis, USA
vsdave@iupui.edu, alhasan@iupui.edu

Chandan K. Reddy
Department of Computer Science
Virginia Tech, Arlington, USA
reddy@cs.vt.edu

Abstract

In the recent years, reciprocal link prediction has received some attention from the data mining and social network analysis researchers, who solved this problem as a binary classification task. However, it is also important to predict the interval time for the creation of reciprocal link. This is a challenging problem for two reasons: First, the lack of effective features, because well-known link prediction features are designed for undirected networks and for the binary classification task, hence they do not work well for the interval time prediction; Second, the presence of censored data instances makes the traditional supervised regression methods unsuitable for solving this problem. In this paper, we propose a solution for the reciprocal link interval time prediction task. We map this problem into survival analysis framework and show through extensive experiments on real-world datasets that, survival analysis methods perform better than traditional regression , neural network based model and support vector regression (SVR).

1 Introduction

Reciprocal altruism is a behavior whereby one performs an act of gift-giving with the expectation that the receiving person will act in a similar manner at a later time (Trivers 1971). People's activities on online social networks are filled with many examples of reciprocal altruism: we follow a friend's Twitter feed with the hope that he will follow back our feed; we endorse our friends for their technical skills in LinkedIn hoping that they will return the favor in a similar manner.

However reciprocity usually has a conflict with another social phenomenon called *social stratification*, which favors hierarchical arrangement of people in a society based on various factors such as power, wealth, and reputation (Hopcroft, Lou, and Tang 2011). This phenomenon is prevalent in online social networks as well. People who are higher up in the hierarchy are sometimes reluctant to perform a reciprocal act to an individual who is at a lower hierarchy; they defer the reciprocal action for a later time or sometimes indefinitely.

For reciprocal link creation, understanding the criteria which control the interval time and building learning models which predict the interval time are important. From the research standpoint, such studies help scientists to study the interaction between reciprocity and different social phenomena. From the perspective of real-life applications in social network analysis, such prediction model enables better link suggestions, where the interval time is also factored in within the suggestion.

The majority of the existing works on link prediction (Hasan and Zaki 2011) assume an undirected network, in which the concept of reciprocal edges do not exist. A few works consider reciprocal link prediction (Hopcroft, Lou, and Tang 2011; Gong and Xu 2014) in a directed network where the prediction is binary, yielding yes/no answer to the question whether a reciprocal link will be created within a fixed observation window. Other works utilize reciprocity as a tool for network compression (Chierichetti et al. 2009) and information propagation in social networks (Zhu et al. 2014). However, none of the existing works consider predicting the interval time for the creation of a reciprocal edge.

Extending a model which predicts a binary answer for reciprocal link prediction to a model which predicts the interval time of reciprocal link is non-trivial. The major challenge for interval time prediction is that, typical link prediction features for undirected network, such as common neighbors, Jaccard's similarity, Adamic-Adar do not have a well-defined counterpart for directed networks, which makes interval prediction a difficult task. Besides, we observe a network for a finite time window, and the absence of a reciprocal link within that time window does not necessarily mean the absence of that reciprocal edge, because a reciprocal edge might have formed outside (after) the observation time window. This yields numerous right censored data instances, for which the target variable, i.e., the reciprocal link formation time is not available. Traditional supervised regression models cannot include censored data instances in the training data and hence perform poorly in predicting reciprocal link creation time.

In this work, we present a supervised learning model for predicting the interval time for the creation of a reciprocal edge between a pair of vertices in an online social networks, given that a parasocial edge already exists between the vertex-pair. We study real-life networks and validate a collection of topological features that may influence the reciprocal edge creation time. Then, we design the prediction task as a survival analysis problem and propose five censored regression models. Our experimental results show that

Cox regression performs better than traditional supervised learning models for reciprocal link prediction.

2 Our Methodology

In this section, we first define the problem of *Reciprocal Link Time Prediction* (*RLTP*). Then we discuss topological features used to solve the *RLTP* problem and how we can map the *RLTP* problem into survival analysis framework. Finally we discuss different survival analysis methods which we have used for solving the *RLTP* problem.

2.1 Problem Formulation

$G(V, E)$ is a **directed time-stamped network** where V is the set of vertices and E is the set of directed edges. For a vertex $u \in V$, $\Gamma_{in}(u)$ and $\Gamma_{out}(u)$ are the set of in-neighbors and the set of out-neighbors of u. $d(u, v)$ is the directed shortest path distance from u to v. There also exists a mapping function $\tau : E \to T$, which maps each link $e \in E$ to a specific time-stamp $\tau(e) = t_e \in T$ denoting the creation time of the link e. For vertices $u, v \in V$ and link $e = (u, v) \in E$ the corresponding time-stamp t_e can be represented as t_{uv}. For a link $(u, v) \in E$, if $\exists (v, u) \in E$ and $t_{vu} < t_{uv}$ then (u, v) is called a **reciprocal link**; on the other hand, if $(v, u) \notin E$, it is called **parasocial link**.

The **time interval** of a reciprocal link (u, v) is defined as $Int(u, v) = t_{uv} - t_{vu}$. Objective of the *RLTP* problem is to predict the time interval of a reciprocal link $Int(u, v)$ given the time-stamp of the parasocial link t_{vu} in a directed network. From the knowledge of $Int(u, v)$ and t_{vu}, the reciprocal link creation time t_{uv} can be obtained easily. The main reason for defining the problem in terms of interval is that interval avoids the problem of temporal bias that exists between the train and test datasets. We use a supervised learning approach for this prediction using only topological features constructed from G.

2.2 Topological Feature Design

The majority of the existing topological features for link prediction are defined for an undirected network, hence we adapt those features for predicting reciprocal links. Our features belong to following two groups: directed altruism based features and social stratification based features.

Directed Altruism Based Features

Here, we define topological features which quantify the directed altruism phenomenon of reciprocal link prediction.

Shortest directed distance: For a parasocial link (v, u), we use directed distance for reciprocal link i.e., $DirectDist(u, v) = d(u, v)$.

Common in/out neighbors count: For directed graphs, we have two separate features: common in-neighbors and common out-neighbors. $Common_{in}(u, v) = |\Gamma_{in}(u) \cap \Gamma_{in}(v)|$, $Common_{out}(u, v) = |\Gamma_{out}(u) \cap \Gamma_{out}(v)|$.

Jaccard coefficient (In/Out): It is normalized version of common neighbors counts, hence similar to common neighbors, Jaccard coefficients can also be presented by two separate features. $Jacccard_{in} = \frac{|\Gamma_{in}(u) \cap \Gamma_{in}(v)|}{|\Gamma_{in}(u) \cup \Gamma_{in}(v)|}$ $Jacccard_{out} = \frac{|\Gamma_{out}(u) \cap \Gamma_{out}(v)|}{|\Gamma_{out}(u) \cup \Gamma_{out}(v)|}$.

Local Reciprocity. In (Gong and Xu 2014), the authors studied two local reciprocity features and showed relative influence of both on linking back probability. Acceptance Local Reciprocity (*ALR*): $ALR(v) = \frac{|\Gamma_{in}(v) \cap \Gamma_{out}(v)|}{|\Gamma_{in}(v)|}$. Request Local Reciprocity (*RLR*): $RLR(u) = \frac{|\Gamma_{in}(u) \cap \Gamma_{out}(u)|}{|\Gamma_{out}(u)|}$. We consider RLR of the tail node (RLR(u)) and ALR for head node (ALR(v)) for reciprocating link (u, v). These features capture the tendency of u to request and the tendency of v to accept a link.

Social Stratification Based Features

The following topologial features quantify the social stratification phenomenon.

Preferential Attachment: The basic idea of preferential attachment is to give more weight to the higher degree nodes. For directed graphs, we consider out degree of tail node and in degree of head node of the future (reciprocating) link, which is given as follows: $PrefAtt(u, v) = |\Gamma_{out}(u)| \times |\Gamma_{in}(v)|$.

Preferential Jaccard: *PrefJacc* is inspired by both Preferential Attachment and Jaccard Coefficient. We calculated *PrefJacc* using the following equation: $PrefJacc(u, v) = \frac{|\Gamma_{out}(u) \cap \Gamma_{in}(v)|}{|\Gamma_{out}(u) \cup \Gamma_{in}(v)|}$.

In/Out Ratio: These features capture the social stratification. A node in the upper hierarchy has a higher tendency to create reciprocal edge with another node at the same level in the hierarchy (Hopcroft, Lou, and Tang 2011). Hierarchy of a node can be identified by the ratio of their in-degrees and out-degrees. Hence, we consider *InRatio* and *OutRatio* as features. $InRatio = \frac{|\Gamma_{in}(u)|}{|\Gamma_{in}(v)|}$ $OutRatio = \frac{|\Gamma_{out}(u)|}{|\Gamma_{out}(v)|}$.

PageRank: *PageRank* represents the prestige of the node in the network. We use both, pagerank of u and pagerank of v as features.

2.3 RLTP and Survival Analysis

In this section, we describe how the *RLTP* problem can be mapped into survival analysis framework and provide definitions of the required concepts to comprehend our approach. Survival analysis is widely used in the medical domain to predict survival time or time to a specific event (Ping Wang and Reddy 2017). For a set of instances under observation, events happen over a time period, from which a survival model learns the temporal patterns of these events.

Survival analysis assumes a starting time of the study, from when a model starts to observe for the events. For the *RLTP* problem, at the first time-stamp, a given directed time-stamped network is static (initialized), the second time-stamp from when new links are added to the static network is called the **beginning of graph expansion**, which serves as the starting time of the study. For *RLTP*, the last time-stamp in the training period is considered to be the **end of the study**. Hence, the time window from beginning of graph expansion to the end of the study is considered to be **study period**. For a parasocial link (v, u), if a reciprocal link (u, v)

is created during the study period, we call it a **reciprocal event**. Time-stamp of a parasocial link is the time when the data instance is considered into the network for study, which is called as the **starting time of observation** for the data instance. We study the network for a limited time window (study period), and hence for a set of parasocial links, the corresponding reciprocal event may not be observed before the end of the study, we call these links as **immortal links**. These immortal links carry the information about links for which the reciprocal link creation event does not happen for a specific period of time. In the survival analysis terminology the immortal links are called censored instances.

Time difference, from starting time of observation for a parasocial link to the time-stamp of the reciprocal event is considered to be the **life-span of the parasocial link**. In the *RLTP* problem, the interval time of a reciprocal link is exactly defined as the life-span of parasocial link, which is the **survival time** in the survival analysis problem.

In a traditional regression task, immortal links may either be ignored, or their survival time may be replaced by a large number which is higher than the time difference between the end of study time and the starting time of observation for that parasocial link. The first of the above cases will be ignoring important information, and the second is simply a crude approximation. These links give precise information that the survival time for immortal links is higher than the time difference between the end of the study and the starting time of observation for that parasocial link.

2.4 Survival Analysis Models

The most widely used survival analysis model is Cox regression model (Cox 1972) which predicts the time taken for an event to occur. We used cocktail algorithm (Yang and Zou 2013) for optimization of the elastic net penalized Cox model. We used two parametric survival models: Accelerated Failure Time (AFT) model and Buckley-James (BJ) model. The AFT model is used with three distributions for survival time: weibull, log-logistic and log-normal. For AFT models and BJ regression, we used *Survival* package[5] and *Bujar* package[6], respectively, available in **R**.

3 Experiments and Results

We conducted a set of rigorous experiments to demonstrate the benefit of using censored information and the superiority of censored models to solve the *RLTP* problem. We used the following five censored models: Cox regression model, three AFT models with Weibull, log-normal and log-logistic distributions respectively, and Buckley-James (BJ) regression model. To prove the fact that the censored models are more suitable for solving the *RLTP* problem, we compared them with traditional regression models, such as, ridge regression (RidgeReg), lasso regression (LassoReg), feed forward neural networks (FFNN) and support vector regression (SVR). Note that, these traditional regression models cannot use censored information (immortal links).

[5]cran.r-project.org/package=survival

[6]cran.r-project.org/web/packages/bujar/index.html

Table 1: TD-AUC results [mean (± standard deviation)] for various methods on real-world datasets.

Models	*Epinion*	*MC-Email*	*Enron*
RidgeReg	0.6086 (±.0013)	0.6083 (±.0146)	0.5847 (±.0159)
LassoReg	0.6020 (±.0014)	0.5709 (±.0201)	0.5850 (±.0152)
FFNN	0.5048 (±.0822)	0.4609 (±.0964)	0.5407 (±.0434)
SVR	0.4871 (±.0039)	0.5737 (±.0187)	0.5680 (±.0176)
BJ Model	0.7339 (±.0020)	0.5910 (±.0146)	0.6096 (±.0076)
Weibull	0.5210 (±.1446)	0.6171 (±.0069)	**0.6319** (±.0050)
logNormal	0.4461 (±.0283)	0.6463 (±.0015)	0.6146 (±.0097)
logLogistic	0.5110 (±.0196)	0.6494 (±.0062)	0.6224 (±.0069)
Cox	**0.7436** (±.0016)	**0.6558** (±.0125)	0.6311 (±.0110)

3.1 Datasets

We used three real-world directed network datasets for our experiments. We selected datasets where reciprocal link creation is an important (meaningful) event; another selection criterion is that the selected datasets have a sufficient number of reciprocal links to train and test the models. Our first dataset, *Epinion*[2] is a trust network where a directed link from one vertex to another vertex represents the fact that the former trusts the latter. The network has $131,828$ vertices and $841,373$ edges created during 938 timestamps. The prediction task for this dataset is to find the time at which a trusted person acknowledges that (s)he also holds a similar sentiment towards the other person. We also collected two enterprise email datasets: *MC-Email*[3] and *Enron*. For both these datasets the task is to predict the response time of an email. *MC-Email* has 167 users with $5,783$ email conversations with 237 timestamps, and *Enron* has 182 users with $3,007$ email links with timestamps range $0-944$.

3.2 Experimental Setting

For our experiments, we divided the time-stamps of a dataset into two non-overlapping continuous partitions, where the earlier partition is the train period and the latter is the test period. We used 70% split of the time-stamps as the training period for the experiments. For calculating the topological feature values, we considered a snapshot of the network until the time-stamp of the reciprocating link or the end of the train period (whichever is earlier).

[2]http://konect.uni-koblenz.de/networks/

[3]This is Manufacturing Company email dataset available from R. Michalski's website, https://www.ii.pwr.edu.pl/~michalski

Like any other link prediction task, *RLTP* also suffers from the class imbalance issue, where the number of positive instances are much smaller than that of the negative instances. To alleviate this problem we use majority undersampling as follows: all the reciprocal links generated during a training period are considered in the training data pool as positive instances and only 50% of the parasocial links generated during the same period are censored negative instances in the pool. The test data pool (and their labels) are also generated similarly from the test period. As train and test data instances need to be from their corresponding time periods, we use a modified K-fold cross validation, where each fold contains a random subset of train and test data instances from their respective pools. For all our experiments, we used 5-fold cross validation in this manner.

For RidgeReg, LassoReg and SVR, we used scikit-learn python library and for FFNN, we used Matlab NNtoolbox. To choose the best parameters of SVR, we used grid search, where the cost parameter C takes values from $\{0.0001, 0.001, 0.01, 0.1, 1.0\}$ and Epsilon (ϵ) takes values from $\{0.0001, 0.001, 0.01, 1.0\}$.

3.3 Evaluation Metrics

Datasets generated from directed time-stamped networks are longitudinal data and for the *RLTP* problem, the datasets also have censored information. Evaluating models on these datasets using traditional evaluation metrics is not suitable, instead we used time-dependent AUC (also known as c-Index), which is widely used in longitudinal data analysis (Pencina and D'Agostino 2004). Time-dependent AUC (TD-AUC) is calculated as follows:

$$TD\text{-}AUC = \frac{1}{N_{cnt}} \sum_{i:C_i=1} \sum_{y_j > y_i} \mathbb{1}(\widehat{y_j} > \widehat{y_i}) \tag{1}$$

where, N_{cnt} is total count of (y_i,y_j) pairs such that $C_i = 1$ (the event has occured) and $y_j > y_i$ holds.

For the i'th data instance $\mathbf{x}_i$ and Cox model parameter $\boldsymbol{\beta}$ the TD-AUC for Cox model can be calculated as:

$$TD\text{-}AUC = \frac{1}{N_{cnt}} \sum_{i:C_i=1} \sum_{y_j > y_i} \mathbb{1}(\mathbf{x}_i^T \widehat{\boldsymbol{\beta}} > \mathbf{x}_j^T \widehat{\boldsymbol{\beta}}) \tag{2}$$

3.4 Comparison results of censored models and regression models

We compared proposed survival models with four traditional regression models and results are shown in Table 1. For the *Epinion* dataset, as depicted in Table 1, Cox regression model performs the best with mean TD-AUC 0.7436. BJ model also performs very good with small degradation (0.7339) compared to the Cox model. Among competing methods, ridge regression performs the best with mean TD-AUC 0.6086. For the *MC-Email* dataset, again Cox regression model performs the best with mean TD-AUC 0.6558. AFT models also perform better than all competing traditional regression methods. Especially, AFT with log-logistic and log-normal distributions perform excellent and their mean TD-AUC is very close to results of Cox regression. Best among competing methods is ridge regression with mean TD-AUC of 0.6083; all other competing methods perform poorer than any censored models. For the *Enron* dataset, AFT model with Weibull distribution performs the best with mean TD-AUC 0.6319. BJ model performs poorly compared to the other survival models with mean TD-AUC 0.6096, still better than all competing methods.

4 Conclusion

In this paper, we proposed a novel problem, namely, reciprocal link time prediction (*RLTP*), which has wide applicability in email, social and other directed networks. We designed various socially meaningful topological features specifically for directed networks, which are useful to solve the *RLTP* problem. We map the *RLTP* problem into a survival analysis framework and through experiments on three real-life network datasets, we show that such a framework is better suited than traditional regression based approaches for solving *RLTP*. To the best of our knowledge this is the first study for the prediction of time interval of reciprocal links.

5 Acknowledgments

This research is supported by National Science Foundation (NSF) career award (IIS-1149851) and in part by the NSF grants IIS-1707498, IIS-1619028 and IIS-1646881.

References

Chierichetti, F.; Kumar, R.; Lattanzi, S.; Mitzenmacher, M.; Panconesi, A.; and Raghavan, P. 2009. On compressing social networks. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, 219–228.

Cox, D. R. 1972. Regression models and life-tables. *J. of the Royal Statistical Society. Series B (Methodological)* 34(2):187–220.

Gong, N. Z., and Xu, W. 2014. Reciprocal versus parasocial relationships in online social networks. *Social Network Analysis and Mining* 4(1):1–14.

Hasan, M. A., and Zaki, M. J. 2011. *Social Network Data Analytics*. Springer. chapter A Survey of Link Prediction in Social Networks, 243–275.

Hopcroft, J.; Lou, T.; and Tang, J. 2011. Who will follow you back?: Reciprocal relationship prediction. In *Proc. of ACM CIKM*, 1137–1146.

Pencina, M. J., and D'Agostino, R. B. 2004. Overall-c as a measure of discrimination in survival analysis: model specific population value and confidence interval estimation. *Stat. in Medicine* 23(13):2109–2123.

Ping Wang, Y. L., and Reddy, C. K. 2017. Machine learning for survival analysis: A survey. *ACM Computing Surveys*.

Trivers, R. L. 1971. The evolution of reciprocal altruism. *Quarterly Review of Biology* 46:33–57.

Yang, Y., and Zou, H. 2013. A cocktail algorithm for solving the elastic net penalized cox regression in high dimensions. *Stat. and Its Interface* 6(2):167–173.

Zhu, Y.-X.; Zhang, X.-G.; Sun, G.-Q.; Tang, M.; Zhou, T.; and Zhang, Z.-K. 2014. Influence of reciprocal links in social networks. *PloS one* 9(7).