Link Weight Prediction for Directed WSN Using Features from Network and Its Dual

Ritwik Malla(✉) and S. Durga Bhavani

School of Computer and Information Sciences, University of Hyderabad, Hyderabad, India
ritwikmalla@gmail.com, sdbcs@uohyd.ernet.in

Abstract. Link prediction problem in social networks is a very popular problem that has been addressed as an unsupervised as well as supervised classification problem. Recently a related problem called link weight prediction problem has been proposed. Link weight prediction on Weighted Signed Networks (WSNs) holds great significance as these are semantically meaningful networks. We consider two groups of features from the literature - edge-to-vertex dual graph features and fairness-goodness scores in order to propose a supervised framework for weight prediction that uses fewer features than those used in the literature. Experimentation has been done using three different feature sets and on three real world weighted signed networks. Rigorous assessment of performance using (i) Leave-one-out cross validation and (ii) N% edge removal methods has been carried out. We show that the performance of Gradient Boosted Decision Tree (GBDT) regression model is superior to the results presented in the literature. Further the model is able to achieve superior weight prediction scores with significantly lower number of features.

Keywords: Link prediction · Line graph · Centrality measures · Fairness and goodness · Supervised regression

1 Introduction

Liben Nowell et al. formulated the problem of link prediction [9] which attempts to predict links that are either missing or that are highly likely to appear in future. In the literature, several unweighted similarity measures for link prediction have been proposed - Adamic Adar (AA), Jaccard Coefficient (JC), PropFlow (PF), Katz (KZ), etc. [14]. Weighted versions of these measures have also been proposed and utilized for link prediction in weighted networks [7,11,13].

Only recently [3], the problem of prediction of link along with weight has been addressed using these weighted similarity measures. In the literature [16] it has been argued that separately designed algorithms for link and weight prediction

B. Deka et al. (Eds.): PReMI 2019, LNCS 11941, pp. 56–64, 2019.
https://doi.org/10.1007/978-3-030-34869-4_7

should yield better results in comparison to a single algorithm designed for both the link and weight prediction.

Link weight prediction attempts to predict the weight of links with missing weight information or links predicted to appear in the future. It has potential applications in recommendation systems, sentiment analysis, network evolution, etc. Importance of weight of link cannot be overstated. There is an earlier work on prediction of sign of the links [8], which is a simpler problem compared to the weight prediction.

The networks may be modeled as graphs of different types such as weighted, unweighted, multi-graph, directed, undirected, etc. In this work, we restrict our attention to weighted signed networks (WSNs) which are weighted graphs where the edge labels contain positive or negative values. WSNs can represent the real world asymmetric nature of human relationships, where person A may like person B but person B may not like person A.

In machine learning, link prediction can be formulated as a binary classification problem [4] where the links are either present or absent and weight prediction can be formulated as a regression problem that yields continuous weight values. In this paper we address the problem of weight prediction of edges in different WSNs.

2 Problem Statement

A WSN is modeled as a directed weighted graph $G(V, E, W)$ where V is the set of vertices, E is the set of edges connecting vertex pairs and $W(e)$ is the set of edge weights, where $e \in E$ and weight takes $-ve$ or $+ve$ sign with $W(e) \in [-1, 1]$. The problem of weight prediction attempts to predict the weights of the edges $e \in E$, where edges e have their weight information missing.

3 Related Literature

J. Zhao et al. propose that weight between any two nodes that do not have a direct edge can be calculated by multiplying the weights of the paths linking them [15].

Fu et al. [3] treat link weight prediction problem as a supervised regression problem where weight is predicted for an undirected network. Fu et al. use a edge-to-vertex dual representation of the graph, called line graph, in which edges of the original graph are transformed into vertices in line graph so that vertex centrality indices can be used to predict link weights. Further the authors argue that weight information may not be captured fully using only similarity indices, hence a combination of similarity indices and node centrality indices have been used for weight prediction.

Kumar et al. [6] are the first authors to predict weights on weighted signed networks (WSNs). Kumar et al. used 11 special features including triadic balance, bias and deserve, along with their proposed new features called goodness and fairness measures. They conduct supervised regression in order to predict the weights on the links.

4 Proposed Approach

We basically take the approach of Fu et al. who take two groups of features, one from the original graph and the other from the edge-to-vertex dual (line graph) in order to perform link weight prediction. We extend this approach to directed weighted signed networks. We choose path based node centrality indices that are relevant for directed graphs such as Page Rank and Betweenness Centrality as features on the line graph. We find that the unsupervised measures proposed by Kumar et al. such as Fairness and Goodness measures [6] are good candidates for graph features.

We use Fairness-Goodness (FG) measures of Kumar et al. as the original graph features along with node centrality features on the line graph and propose a supervised regression approach.

The work done by Kumar et al. has been considered as the baseline method and we investigate if the performance can be improved as well as if the number of features can be reduced in the supervised regression approach to weight prediction.

A flowchart depicting our approach is shown in Fig. 1.

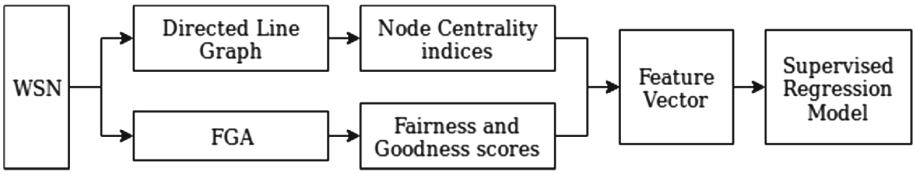
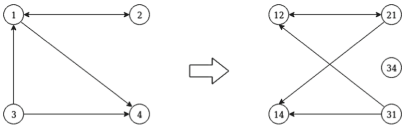


Fig. 1. Proposed approach



$$g(v) = \frac{1}{|in(v)|} \sum_{u \in in(v)} f(u) \times W(u,v) \quad (1)$$

$$f(u) = 1 - \frac{1}{|out(u)|} \sum_{v \in out(u)} \frac{|W(u,v) - g(v)|}{R} \quad (2)$$

Fig. 2. Line graph transformation

4.1 Directed Line Graph

The approach has to be carefully tuned for directed graphs. We use the notion of directed line graph in the following way:

If the directed edges $(u, v), (v, w) \in E$ then an edge is added between the vertices (uv,vw) in the line graph. An example for line graph construction that depicts the conversion of original directed graph to directed line graph is given (See Fig. 2).

4.2 Fairness Goodness Measures

The central recursive equations of fairness goodness algorithm(FGA) of Kumar et al. [6] are given in Eqs. (1) and (2) respectively.

Goodness of a vertex v, $g(v)$, is the mean weighted incident rating over the links $(u, v) \in E$ of $G(V, E)$. $g(v)$ represents the trust that the in-neighbours of v place on vertex v. Therefore, $g(v)$ is high if more number of fair nodes rate vertex v. $g(v)$ is considered to be an estimate of $w(u, v)$ and fairness of vertex u, $f(u)$, is high if it has more number of good out-neighbors.

The fairness and goodness scores are mutually recursive and the FGA algorithm stops when in consecutive iterations for all vertices, the difference between the consecutive scores of $f(u)$ and $g(v)$ is less than an error threshold. Fairness scores lie in [0, 1]. Goodness scores and edge weights lie in $[-1, 1]$. The largest difference between a weight and goodness score is the range of difference, R = 2.

4.3 Feature Extraction

Feature extraction is done by calculating *fairness and goodness* (FG) features on the original graph and *node centrality* features on the line graph (Line). We used LPMade [10] tool to convert the line graph from an edge list to the network file format and compute the standard path based node centrality indices like node betweenness centrality and pagerank [14]. Fairness and goodness measures have been computed using the code made available by the authors [2].

Three different feature sets have been considered for the weight prediction experimentation.

1. **Feature set I (Line + FG)** This experiment builds a supervised learning model for directed link (u, v) using two FG measures - $f(u) * g(v)$, $g(v)$ and two Line measures - page rank and node betweenness centrality measures.
2. **Feature set II (Line)** This experiment is done using only the two Line measures, namely page rank and node betweenness centrality on the line graph.
3. **Feature set III (FG)** This experiment uses only the two FG measures $f(u) * g(v)$ and $g(v)$.

5 Experiments and Results

Table 1 gives a concise description of the three real world weighted signed network (WSN) datasets used by Kumar et al. These data sets have been chosen in order to compare the performance with the work of Kumar et al. [6]. The original datasets are available on SNAP and the normalized datasets after data preprocessing are made available by the authors at [2]. Bitcoin-Alpha (BTC Alpha) and Bitcoin-OTC (OTC Net) are the two directed signed bitcoin exchange networks that have been considered in this paper where each vertex is a Bitcoin user and each edge is the trust rating that they give to each other. Wikipedia

Requests for Adminship (Wiki RfA) network is a directed signed network where each vertex is a Wikipedia user and each edge (A, B) represents the weight corresponding to the vote that user A gave to user B.

5.1 Datasets

Table 1. Weighted signed networks (WSNs)

Network	Vertices	Edges
BTC Alpha	3783	24,186
OTC Net	5881	35,592
Wiki RfA	9654	104,554

5.2 Graph Preprocessing

The transformation of original graph to the corresponding line graph should satisfy the criteria that the number of vertices of line graph is the same as the number of edges in the original graph. Since the original graph is a directed graph, the transformation can lead to isolated vertices in the line graph. Hence the isolated vertices in the line graph and the corresponding directed edges in the original graph are removed from further consideration.

The impact of isolated vertices removal from line graph is shown in Table 2.

Table 2. Undirected and directed line graphs on isolated vertices removal

	BTC Alpha			OTC Net			Wiki RfA		
	Vertices	Edges	File size	Vertices	Edges	File size	Vertices	Edges	File size
Undirected line graph	24184	2518684	28.2 MB	35591	4693528	54.1 MB	104554	11116567	131.5 MB
Directed line graph	24177	1256332	14.1 MB	35586	2301858	26.5 MB	99307	3564619	41.9 MB

5.3 Evaluation

Two standard metrics being used to calculate the performance of link weight prediction are *Root Mean Square Error (RMSE)* and *Pearson Correlation Coefficient (PCC)*. *RMSE* is the error calculated between the actual and predicted weight and the range of the value lies in between 0 to 2, as edge weights are between -1 and 1. Therefore, the value 0 is the best and 2 is the worst. PCC is a measure of the linear correlation between two variables X and Y. The range for *PCC* lies between -1 to 1 and 0 denotes randomness.

5.4 Regression Model

Three regression models namely, *Linear Regression* and *Elastic Net* and *Gradient Boosted Decision Tree (GBDT)* have been used. The first two models are considered in order to compare the results of our approach with those of Kumar et al. And among the various experiments we carried out with other standard models available in WEKA [1], *GBDT* is chosen for its superior performance in comparison with the rest of the models.

Gradient boosting is an ensemble of typically weak prediction models like decision trees and further that uses boosting. The implementations have been done in WEKA. For squared error loss function, *GBDT* has been implemented using *Additive Regression*, *Bootstrap Aggregating (Bagging)* and *Reduced Error Pruning Tree (REPTree)*. For the GBDT experimentation, we consider default values of $batchSize = 100$, $numIterations = 10$, $shrinkage = 1.0$, $seed = 1$ and $maxDepth = 6$. We found that depth restricted tree gives better prediction scores in comparison to that of depth unrestricted tree.

The robustness of the models is tested thoroughly following the approach of Kumar et al. using the methods given below:

1. **Leave-one-out cross validation.** This is an extreme form of cross validation where number of folds equals the number of instances. The number of models being built for training/testing is equal to the number of instances in the dataset and hence involves a large amount of experimentation.
2. **N% edge removal.** The network data is split into train and test sets where test dataset is of size N% edges, where N is taken from 10 to 90 with a step size of 10. The model is trained with remaining $(100 - N)$ edges and the calculation of the prediction scores on test set is conducted. The process is repeated for each split 10 times with a random seed value from 1 to 10. And the mean of the scores is computed and tabulated.

5.5 Results and Analysis

The results obtained by the supervised regression model ($AdditiveRegression + Bagging + REPTree$) are reported here. Also, we conducted experiments by replacing *REPTree* with *Random Tree* to insert randomness in the learning process and found that both models show equal performance. Therefore, we only consider the former model.

Leave-One-Out Cross Validation. The results given in Tables 3, 4 and 5 show that *GBDT* regression model achieves an improvement in PCC of 4%, 9% and 5% in case of BTC Alpha, OTC Net and Wiki RfA respectively in comparison to all the models considered by Kumar et al. Note that, the *GBDT* model built using feature set I is composed of only 4 features whereas the model of Kumar et al. is built using 13 features. It is interesting to note that just by taking the two FG features of Kumar et al. in a supervised framework, the performance of *GBDT* model outperforms the best results of Kumar et al.

Table 3. Leave-one-out cross validation results for weight prediction on BTC Alpha

	Number of features	Linear regression		Elastic net		GBDT	
		RMSE	PCC	RMSE	PCC	RMSE	PCC
Feature set I	4	0.24	0.59	0.24	0.59	**0.22**	**0.66**
Feature set II	2	0.29	0.09	0.29	0.09	0.28	0.25
Feature set III	2	0.24	0.58	0.24	0.58	0.23	0.63
Kumar et al. supervised	13	**0.22**	0.62	0.24	0.60	–	–

Table 4. Leave-one-out cross validation results for weight prediction on OTC Net

	Number of features	Linear regression		Elastic net		GBDT	
		RMSE	PCC	RMSE	PCC	RMSE	PCC
Feature set I	4	0.27	0.66	0.27	0.65	**0.24**	**0.75**
Feature set II	2	0.35	0.15	0.35	0.15	0.32	0.45
Feature set III	2	0.27	0.65	0.27	0.65	0.25	0.72
Kumar et al. supervised	13	0.26	0.66	0.27	0.65	–	–

Now, comparing the performance of different feature sets among the models chosen by Kumar et al., we observe the following. *Linear Regression* and *Elastic Net* models with feature set I (4 features) closely approximate the results of Kumar et al. (13 features) for *PCC* and *RMSE*. It is to be noted that in Tables 4 and 5 the results obtained using only FG features (Feature set III) are almost equal to that of Kumar et al.

Clearly, Feature set I which combines Line and graph features outperforms all the other combinations. It is interesting to note that though line graph features by themselves perform poorly and FG measures perform very well, their combination gives slightly improved results. For example in case of OTC Net in Table 4, $(Line + FG)$ gives a PCC value of 0.75 whereas Kumar et al. obtain a PCC of 0.65 using 13 features. In fact, among the models tested *GBDT* gives the lowest RMSE of 0.24 as compared to 0.26 obtained by Kumar et al. A similar trend can be seen for the other two datasets in Tables 3 and 5.

N% Edge Removal. In Fig. 3, the performance using N% edge removal of four models: *Linear Regression*, *Elastic Net* and two *GBDT* models are plotted. The two GBDT models are built using Line features, one generated using undirected line graph *GBDT_U* and the other using directed line graph *GBDT_D*. In Wiki RfA, *GBDT_D* model achieves better scores compared to that of *GBDT_U* model. In BTC Alpha and OTC Net, *GBDT_U* model achieves comparable scores to that of *GBDT_D* model. And in all the networks, *GBDT* models give lower *RMSE* and higher *PCC* in comparison to the *Linear Regression* and *Elastic Net* models. Therefore, we do not include BTC Alpha and OTC Net plots. And *GBDT_D* is preferable since undirected line graph is not scalable as can be seen in Table 2.

Table 5. Leave-one-out cross validation results for weight prediction on Wiki RfA

	Number of features	Linear regression		Elastic net		GBDT	
		RMSE	PCC	RMSE	PCC	RMSE	PCC
Feature set I	4	0.23	0.49	0.23	0.50	**0.22**	**0.55**
Feature set II	2	0.26	0.04	0.26	0.04	0.25	0.24
Feature set III	2	0.23	0.49	0.23	0.50	**0.22**	0.54
Kumar et al. supervised	13	**0.22**	0.50	0.23	0.47	–	–

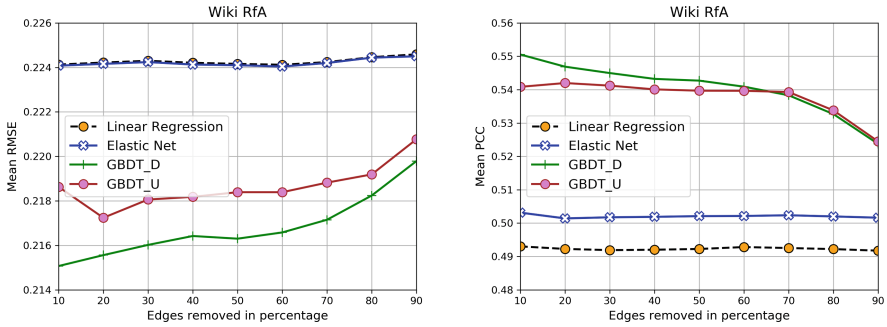


Fig. 3. N% edge removal performance evaluation in terms of RMSE and PCC

6 Conclusion

We have demonstrated that *GBDT* models outperform *Linear Regression* and *Elastic Net*. Our $(Line + FG)$ approach which takes only four features (Feature Set I) and applies *GBDT* model performs better than that of Kumar et al. which considers 13 features. In case of WSNs, even though *GBDT_D* and *GBDT_U* give comparable results, *GBDT_D* approach has significant computational advantage over *GBDT_U*.

This work is done on a static snapshot of the network. The results need to be validated further using larger networks. We would like to apply these ideas to predict ratings on a recommendation network. And since temporal information in the network is utilized to enhance the performance of link prediction [5,12], a similar approach for weight prediction may be a useful direction to pursue.

References

1. Weka project webpage. https://www.cs.waikato.ac.nz/ml/weka/. Accessed 30 May 2019
2. WSN code and data. https://cs.stanford.edu/~srijan/wsn/. Accessed 30 May 2019
3. Fu, C., et al.: Link weight prediction using supervised learning methods and its application to yelp layered network. IEEE Trans. Knowl. Data Eng. **30**(8), 1507–1518 (2018)

4. Hasan, M.A., Chaoji, V., Salem, S., Zaki, M.: Link prediction using supervised learning. In: Proceedings of SDM 06 workshop on Link Analysis, Counterterrorism and Security, Bethesda, Maryland, USA (2006)
5. Jaya Lakshmi, T., Durga Bhavani, S.: Temporal probabilistic measure for link prediction in collaborative networks. Appl. Intell. **47**(1), 83–95 (2017)
6. Kumar, S., Spezzano, F., Subrahmanian, V.S., Faloutsos, C.: Edge weight prediction in weighted signed networks. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 221–230, Barcelona, Spain, December 2016
7. Lakshmi, T.J., Bhavani, S.D.: Link prediction measures in various types of information networks: a review. In: IEEE/ACM 2018 International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018, Barcelona, Spain, 28–31 August 2018, pp. 1160–1167 (2018)
8. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 641–650. ACM, New York (2010)
9. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. J. Am. Soc. Inform. Sci. Technol. **58**(7), 1019–1031 (2007)
10. Lichtenwalter, R.N., Chawla, N.V.: LPmade: link prediction made easy. J. Mach. Learn. Res. **12**, 2489–2492 (2011)
11. Lü, L., Zhou, T.: Link prediction in weighted networks: the role of weak ties. EPL (Europhys. Lett.) **89**(1), 18001 (2010)
12. Munasinghe, L., Ichise, R.: Time aware index for link prediction in social networks. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2011. LNCS, vol. 6862, pp. 342–353. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23544-3_26
13. Murata, T., Moriyasu, S.: Link prediction of social networks based on weighted proximity measures. In: IEEE/WIC/ACM International Conference on Web Intelligence (WI 2007), pp. 85–88. CA, USA, November 2007
14. Newman, M.: Networks : An Introduction, 1st edn. Oxford University Press, Oxford (2010)
15. Zhao, J., et al.: Prediction of links and weights in networks by reliable routes. Sci. Rep. **5**, 12261 (2015)
16. Zhu, B., Xia, Y., Zhang, X.J.: Weight prediction in complex networks based on neighbor set. Sci. Rep. **6**, 38080 (2016)