

Course Project Title

Social Network Analysis for Computer Scientists — Course Project Paper

Name One

LIACS, Leiden University
sXXXXXX@umail.leidenuniv.nl

Name Two

LIACS, Leiden University
sXXXXXX@umail.leidenuniv.nl

ABSTRACT

Abstract.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous;

D.2.8 [Software Engineering]: Metrics—*complexity measures, performance measures*

General Terms

Theory

Keywords

ACM proceedings, L^AT_EX, text tagging

1. INTRODUCTION

One of the main ideas behind link prediction is that it offers several ways in which one can study networks of any kind in which data scientists, software engineers and researchers could greatly benefit from. Social networks are a popular way to model the interactions among people in a group or community. Their connections can be visualized as a graph, where a vertex corresponds to a person in a group and an edge represents some form of association between the corresponding persons.

Organizations, such as Amazon, can extract information based on raw data in order to predict what the customer might actually buy or find interesting. Another example would be the professional social-media platform, Linked-in. Based on incoming data they would be able to, for instance predict your next connection; link back jobs which are relevant to you; link contents/articles based on your connections, interests, and things you read on the web. Similarly, security corporate companies could more precisely focus their efforts based on probable relationships in malicious networks that have heretofore gone unobserved, and researchers can easily adapt link prediction methods to iden-

tify links that are surprising given their surrounding network, or links that may not exist at all [11].

That being said, links or associations are usually based on someone's behavior and interests. In social networks however, objects are dynamic given that the number of edges and nodes are changing continuously. Getting a hold of those dynamics [number of nodes and edges] is one of the fundamental problems in social network analysis. One of the essential question(s) which we are going to cover in this paper is, given a network at time t , accurately predict the links that are going to form at a future time t' . This is known as the link prediction problem.

In this paper, we approach the problem of link prediction using supervised learning [6], [7], [11], [14]. Unsupervised techniques have also been used. The main difference between the two is that in the unsupervised case, we calculate metrics that individually provide predictions of whether a link will form or not. Supervised learning employs a more sophisticated framework where each of these metrics could be considered as an input to a machine learning model.

The main purpose of this paper is to compare the methods described in [7] and [11]. Both papers consider collaboration networks of co-authors on which several machine learning models are applied in order to predict future links (collaborations) between co-authors. Under sampling is used in both papers in order to solve the problem of class imbalance and decrease the size of the training sets. Additionally, in [11], the training set is constructed by considering the n -neighborhood of each node. Moreover, it is proposed to use a specific type of decision trees, called Hellinger trees, applied in the original distribution of the data, that is without employing under sampling. Although this recommendation is given, the authors do not provide results of this method. Thus, the aim of this paper is to first compare the methods described in [7] and [11], and second, apply Hellinger trees in the original data distribution.

The paper is structured as follows. In Section 2, we formalize the link prediction problem. Specifically, we discuss the transformation of a network to an ordinary data set, and the class imbalance problem. Related work is provided in Section 3. In Section 4, we describe in detail the methods compared in this paper. Section 5 describes the data sets in which the experiments and results of Section 6 are based on. Finally, we conclude with Section 6.

This paper is the result of a student course project, and is based on methods and techniques suggested in [?, ?, ?, ?]. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice on the first page. SNACS '17 Social Network Analysis for Computer Scientists, Master CS, Leiden University (liacs.leidenuniv.nl/~takesfw/SNACS).

2. PROBLEM STATEMENT

Consider at a particular time t , an undirected network $G = \langle V, E \rangle$, where V is the set of nodes and E is the set of edges. The goal is to predict the edges that will form in the network at time $t' > t$. As an example, consider a collaboration network of co-authors. Two authors v and w are linked when they have published at least one research paper together. The goal is to predict future connections between co-authors that have not yet collaborated.

In this paper, we approach the problem of link prediction using supervised learning. In this framework, in order to build models with predictive ability, one needs to construct two sets, mainly a *train* and a *test* set. Then, the model learns features and patterns from the train set, and it is evaluated on the test set. These two sets form together the *data set*. This set is of the form (\vec{x}, y) , where \vec{x} indicates the inputs to the model, and y indicates the label, that is whether or not a link is present between two nodes. The model learns from the instances (\vec{x}, y) of the train set. In the evaluation phase, we feed the instances \vec{x} of the test set to the trained model, and the output is the predicted label, i.e. our prediction of whether or not a link will form between two nodes.

We face two problems here. First, how to construct the train and test set, and secondly, what are \vec{x} , y in our model.

2.1 Constructing Data Sets

The process of constructing the train and test set is described in detail [7], [11]. In order to transform the network to a data set, we first choose two adjacent periods, the *train* and *test period*. In the train period, denoted by $[0, t_x]$, we have pairs of authors that are connected, that is, they have written at least one research paper together. The same holds for the test period $[t_{x+1}, t_y]$. We construct a new network $G_x = \langle V_x, E_x \rangle$ consisted of all pair of authors that do not share a connection in the train period. We then examine whether or not they co-authored a paper in the test period. If they did, we assign the pair a positive label, and otherwise a negative label. The data set contains $\binom{|V_x|}{2} - E_x$ instances, since we do not account for self-loops, and in undirected networks each pair is recorded once.

Selecting an appropriate set of features is one of the most important part in supervised learning algorithms [7]. The inputs to the model are selected by calculating topology-based and node-based metrics from G_x that will serve as attributes to the data set. Such metrics include, but not limited to, *common neighbors*, *Jaccard coefficient*, *Adamic Adar*, *preferential attachment*, *Katz measure*, *PropFlow*, *SimRank*, and many others. All these features can be used individually to provide naive predictions. This is sometimes called the unsupervised framework of link prediction. In the supervised case, we use state-of-the-art predictive models such as Neural Networks, Support Vector Machines, Decision Trees, and Random Forests that treat these individual metrics as inputs. The output, or prediction of the model is whether or not a link will form between two specific nodes.

That being said, we end up with a data set in the standard format (\vec{x}, y) , where \vec{x} are metrics extracted from G_x and y indicates the label, that is whether or not a pair of nodes is connected or not.

2.2 Class Imbalance and Size

Although we have constructed the data set, there are two main problems. First, the size of the data set is very large which can make the training process infeasible. For example, consider the relatively small in size *ca-CondMat* network [9], consisted of 23133 nodes and 93497 edges. Defining the train period to be two-thirds of the networks' edges, and assuming that all node appear in V_x , the resulting data set would be of size $\binom{23133}{2} - 62631$, which yields more than 267 million instances. Additionally, the distribution of the classes is highly unbalanced. That is, the amount of instances belonging to the negative class is overwhelmingly greater than the instances of the positive class. Those two issues make the training process difficult or even impossible. There mainly exist two approaches to solve these problems.

One method consists of under sampling the data set to balance in order to create classes of almost the same size. In this way, the size of the training set decreases, and at the same time we obtain balanced classes in which we can fit ordinary predictive models such as, Decision Trees, Support Vector Machines, Neural Networks etc. Note that many of these models perform poorly in highly unbalanced data.

A second method is proposed in [11]. Specifically, it is suggested to treat each neighborhood as a separate problem. We define the n -neighborhood of a node v as the set of nodes that are 0 to n steps away from v . Instead of considering all pairs of nodes to enter the data set, we choose for each node, and for a specified n , only those that belong to the n -neighborhood of the node. In this way, the size of the data set decreases greatly. Furthermore, it is proposed to implement a particular type of decision trees, called Hellinger trees [2], [1], without altering the distribution of the data, that is without under sampling. Hellinger trees use the Hellinger distance as splitting criterion, which is skew insensitive [2].

3. RELATED WORK

The link prediction problem was formalized in [10], where there is an extensive analysis of different baseline predictors, applied to co-authorship networks. Nevertheless, state-of-the-art supervised learning models are not studied in this paper. Furthermore, co-authorship networks have been analyzed in [7], [12], and [3], but this time several machine learning algorithms such as Decision Trees, SVM, Neural Networks, and K-Nearest Neighbors are used for prediction. In contrast to [7], where static, unweighted networks are used, [12] takes into account the evolution of the network, and the authors of [3] analyze a weighted network. Several recommendations and suggestions are presented in [11], regarding how to perform link prediction, and how to approach inherent issues such as dealing with class imbalance and huge train sets.

In the domain of counter-terrorism, link predictions has been studied in [4], [6]. In the latter paper, several networks are constructed by removing nodes from the original graph, called visible networks. A set of features is calculated from these networks in order to predict links of the original one, using supervised learning models. A more probabilistic approach is taken in [5], where chance-constrained programs are used, and in [13], in which the framework of Relational

Markov Network is utilized. Finally, [15], and [8] are extensive surveys regarding the different metrics, classification methods, and approaches that have been used for link prediction.

The main purpose of this paper is to compare the methods applied in [7], and [11]. Specifically, in [7], under sampling of the data is used in order to decrease the size of the training set and apply several machine learning algorithms. On the other hand, in [11] the problem is approached by considering each neighborhood as a separate problem, as we discussed in the Section 2.2. In the latter paper, both the methods of under sampling to balance, and the use of Hellinger trees are discussed. Nevertheless, the authors chose to provide results only for the under sampling method. Therefore, the purpose is to fill this gap by first, comparing the methods in the two papers, and additionally apply Hellinger trees in the original distribution of the data, without the use of under sampling. Details regarding the two approaches and the differences between them are provided in the next section.

4. REFERENCES

- [1] D. A. Cieslak and N. V. Chawla. *Learning Decision Trees for Unbalanced Data*, pages 241–256. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [2] D. A. Cieslak, T. R. Hoens, N. V. Chawla, and W. P. Kegelmeyer. Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery*, 24(1):136–158, Jan 2012.
- [3] H. R. De Sá and R. B. Prudêncio. Supervised link prediction in weighted networks. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2281–2288. IEEE, 2011.
- [4] M. Dombroski, P. Fischbeck, K. Carley, et al. Estimating the shape of covert networks. In *Proceedings of the 8th International Command and Control Research and Technology Symposium*, 2003.
- [5] J. Doppa, J. Yu, P. Tadepalli, and L. Getoor. Chance-constrained programs for link prediction. In *NIPS Workshop on Analyzing Networks and Learning with Graphs*, 2009.
- [6] M. Fire, R. Puzis, and Y. Elovici. *Link Prediction in Highly Fractional Data Sets*, pages 283–300. Springer New York, New York, NY, 2013.
- [7] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.
- [8] M. A. Hasan and M. J. Zaki. *A Survey of Link Prediction in Social Networks*, pages 243–275. Springer US, Boston, MA, 2011.
- [9] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1), Mar. 2007.
- [10] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management, CIKM '03*, pages 556–559, New York, NY, USA, 2003. ACM.
- [11] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10*, pages 243–252, New York, NY, USA, 2010. ACM.
- [12] M. Pavlov and R. Ichise. Finding experts by link prediction in co-authorship networks. In *Proceedings of the 2Nd International Conference on Finding Experts on the Web with Semantics - Volume 290, FEWS'07*, pages 42–55, Aachen, Germany, Germany, 2007. CEUR-WS.org.
- [13] B. Taskar, M. fai Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In S. Thrun, L. K. Saul, and P. B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 659–666. MIT Press, 2004.
- [14] C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, ICDM '07*, pages 322–331, Washington, DC, USA, 2007. IEEE Computer Society.
- [15] P. Wang, B. Xu, Y. Wu, and X. Zhou. Link prediction in social networks: the state-of-the-art. *CoRR*, abs/1411.5118, 2014.