

Joint Neighborhood Subgraphs Link Prediction

Dinh Tran-Van¹, Alessandro Sperduti¹, and Fabrizio Costa²

¹ Department of Mathematics, Padova University, Italy

² Department of Computer Science, University of Exeter, United Kingdom
{dinh, sperduti}@math.unipd.it, f.costa@exeter.ac.uk

Abstract. Graphs are common data structure used to represent data in which the relation between entities are encoded. A number of graph-based methods and systems have been proposed, varying from biology to recommendation to social network. Most methods have to take partially observed graphs as the input to proceed due to the lack of information. This prevents them from showing promising results. Link prediction becomes an effective paradigm to solve this problem. However, most existing link prediction methods are based on the transitive manner which cannot effectively exploit graph structures. Here we propose a method that first represents each link between two nodes by a graph composed of their neighborhood subgraphs. We then employ a compositional graph kernel to measure the similarity between links. An empirical experiment on different datasets proves that our proposed method shows the state of the art for link prediction.

Keywords: Link prediction, joint neighborhood subgraphs

1 Introduction

A huge amount and variety of relational data are available in a wide range of fields. On the one hand, it provides an opportunity to build systems which automatically extract useful information. On the other hand, it raises challenges for machine learning and data scientists to design high performance predictive systems. A common data structure for representing relational data is graph whose nodes describe for entities and links depict their relations. We can easily see graph-based systems in different domains like biology, chemistry or networking. These systems require the data form used for their input as graphs. However, despite the availability of huge data, most graphs are only partially observed graphs. That means a number of actual links in each graph are absent or missing. Tackling this issue is a necessary condition to obtain high performance graph-based predictive systems. As a consequence, many link prediction methods have been proposed with the aim of recovering the missing links. A graph could be associated with external information sources, so called "side information". Therefore, a link prediction method attempts to exploit the graph topology and side information to build a the model for predicting the presence or absence of link any between couple of nodes. An expected method is required

to make use of both such kind of data sources. In this paper, we only focus on the topological information. Thus we term link prediction methods to refer to topology-based link prediction methods.

Concerning link prediction methods, we can classify into two classes: unsupervised learning and supervised learning. Unsupervised learning methods compute the scores for node couples based on topological features and directly use these scores to consider non-observed links to be present or absent. Typical examples link prediction methods which fall into this class are Adamic-Adar [1], Katz [2], Leicht-Horne-Newman [3], etc. Supervised learning link prediction methods are based on a learning process which try to separate present links and absent links by their characteristics. A link prediction method via matrix factorization [5] is for instance. This group of methods can also be further grouped into four subsets: feature-based models, graph regularization models, latent class models and latent feature models as presented in [5]. For the performance, supervised methods tend to show better performance comparing to unsupervised ones in general. However, they normally face with the high complexity in term of computation and memory consumption.

The common feature of most existing supervised and unsupervised methods for link prediction is that they use transitive manner to exploit the graph structure. This manner normally does not show high discriminative capacity. Therefore, in this paper, we propose a novel supervised method named joint neighborhood subgraphs link prediction (JNSL). Our method first introduces a method to represent each link between two nodes as a graph composed by taking a union of neighborhood subgraphs rooted at these corresponding nodes. It then adopts a compositional graph kernel which allow to fully exploit graphs' structure to measure the similarity, one of the key point in machine learning algorithms.

2 Method

In this section, we describe our proposed link prediction method which is based on the graph structure exploitation only.

2.1 Definitions and notation

We consider a graph $G = (V, E)$ where V is the set of nodes and E is the set of links. The set E can be divided into a subset of observed links (O) and a subset of unobserved links (U). The problem is that the set O is not a complete set. Therefore, *link prediction* is the task that aims at exploiting graph structure to rank links in U from the most to the least probable to be considered as observed links in O .

The *distance* between two nodes u and v , notated as $\mathcal{D}(u, v)$, is the length of the shortest path between them. The *neighborhood* with radius r of a node v is the set of nodes at a distance no greater than r from v and denoted by $N_r(v)$. The *neighborhood subgraph* with radius r of node v , denoted by \mathcal{N}_r^v , is

the subgraph formed by the nodes in the neighborhood with radius r of v and the relative edges with endpoints in $N_r(v)$. The degree of a node u denoted as $d(u)$ is the number of its neighborhood $d(u) = \mathcal{N}_1^u$.

2.2 The neighborhood subgraph pairwise distance kernel

The NSPDK [13] is an instance of convolution kernel [14] where given a graph $G \in \mathcal{G}$ and two rooted graphs A_u, B_v , the relation $R_{r,d}(A_u, B_v, G)$ is true iff $A_u \cong \mathcal{N}_r^u$ is (up to isomorphism \cong) a neighborhood subgraph of radius r of G and so is $B_v \cong \mathcal{N}_r^v$, with roots at distance $\mathcal{D}(u, v) = d$. We denote R^{-1} as the inverse relation that returns all pairs of neighborhoods of radius r at distance d in G , $R_{r,d}^{-1}(G) = \{A_u, B_v | R_{r,d}(A_u, B_v, G) = \text{true}\}$. The kernel $\kappa_{r,d}$ over $\mathcal{G} \times \mathcal{G}$, counts the number of such fragments in common in two input graphs:

$$\kappa_{r,d}(G, G') = \sum_{\substack{A_u, B_v \in R_{r,d}^{-1}(G) \\ A'_{u'}, B'_{v'} \in R_{r,d}^{-1}(G')}} \mathbf{1}_{A_u \cong A'_{u'}} \cdot \mathbf{1}_{B_v \cong B'_{v'}},$$

where $\mathbf{1}_{A \cong B}$ is the *exact matching function* that returns 1 if A is isomorphic to B and 0 otherwise. Finally, the NSPDK is defined as $K(G, G') = \sum_r \sum_d \kappa_{r,d}(G, G')$, where for efficiency reasons, the values of r and d are upper bounded to a given maximal r^* and d^* , respectively.

2.3 Node labeling

NSPDK takes labeled graphs as its input to proceed. Therefore, graph node labeling is a step needs to be done before employing NSPDK for graph similarity measure. Here, we propose to use node labeling functions L that take into account node degrees for their labels such that $\ell(u) = \ell(v)$ if $|d(u) - d(v)| \leq \lambda$, in which λ is a hyper parameter. It is due to the assumption that nodes with similar degrees tend to share common properties. The values of λ depend on the histogram of graph node degree. We can use tuning in order to obtain suitable value for λ .

2.4 Link representation

In this section, we propose a method to represent links in the form of neighborhood subgraphs.

Let us consider two nodes u and v ($u, v \in V$), we first extract two neighborhood subgraphs with radius R rooted at u and v to obtain $N_R(u)$ and $N_R(v)$, respectively. We then represent a link connecting u and v by a graph G_{uv} formed by taking a union of these two achieved subgraphs: $G_{uv} = N_R(u) \cup N_R(v)$.

2.5 Joint neighborhood subgraphs link prediction

Given a graph $G(V, E)$ and a set of links $L = \{(u, v) \mid (u, v) \in O \text{ or } (u, v) \in U\}$, our method consists five steps:

- Node labeling: Nodes in G are labeled using a defined node label function ℓ .
- Link representing: Link in L are represented as joint neighborhood subgraphs. As a result, we have a set of graphs $SG = \{G_{uv} \mid (u, v) \in L\}$
- Kernel computation: Obtained graph set SG is used to compute a gram matrix, K , by using NSPDK. K shows the similarities between graphs (links).
- Model configuration: K is then fed into a kernel machine to build a model which tries to learn a function to separate observed links from non-observed ones in the sense of properties.
- Scoring and ranking: Now the model is used to return scores for coming links. These scores are used to rank links from the most to the least probable of being present in the graph.

3 Experiment

In this section, we intend to measure the performance of our proposed method and compare it with other link prediction methods. We follow the procedure used in [5] which employs six datasets in different fields. Following, we give each dataset a short description.

- *Protein* [6]: the medium confidence network which encodes the interaction between proteins. It contains 2617 nodes and 11855 links.
- *Metabolic* [7]: a biological network which relates enzyme proteins and chemical compounds and includes 668 nodes and 2782 links.
- *Nips* [8]: a network of co-authors at NIPS conference containing 2865 nodes and 4733 links.
- *Condmatt* [9]: a network of co-authors for condensed matter physicists. This network has 14230 nodes and 1196 links.
- *Conflict* [10], [11]: a network describing the dispute between countries. It consists of 130 nodes and 180 links.
- *Powergrid* citepowergrid: a network of electric powergrid in US. It has 4941 nodes and 6594 links.

We evaluate the performance of employed methods via a 10 splits. Each round, a given ratio of links will be used to train models, and the rest of links are used to test. For *Protein*, *Metabolic*, *Nips* and *Conflict* networks, we set the training ratio to 10% while the training ratio for *Condmatt* and *Powergrid* to 90%. The performance of each method is computed by taking an average of ROC-AUC over 10 rounds.

Model Selection: The values of different hyper-parameters are set by using a 3-fold on training set in which we use one fold for training and the rest two folds for validation. We tune the values of radius for extracting subgraphs R in $\{1, 2\}$, λ for node degree function in $\{1, 2\}$, for r and d parameters of NSPDK in $\{1, 2\}$ and $\{1, 2, 3\}$, respectively. Finally, the regularization tradeoff C for the SVM is picked up in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4\}$.

4 Results and Discussion

Table 1 shows the performance of link prediction methods in ROC-AUC on six datasets. From the results on the table, we can group methods into two groups based on their performances: supervised methods and unsupervised methods. The performance of supervised methods are considerably higher than unsupervised ones in most cases, except in the Conflict dataset where Sup-Top outperforms Fact+Scores, but with a very small difference. Concerning supervised methods, JNSL outperforms Fact-Scores in all cases. The difference between their performance is small in PowerGrid and Protein datasets with 0.5% and 0.8%, respectively.

Table 1. The performance in percentage on six datasets of different link prediction methods in which AA: Adamic-Adar, PA [4] preferential Attachment, SHP: Shortest Path, Sup-Top: Linear regression running on unsupervised scores [5], SVD [5]: Singular value decomposition, Fact+Scores [5]: Factorization with unsupervised scores.

	Datasets					
Methods	Protein	Metabolic	Nips	Condmat	Conflict	PowerGrid
AA	56.4±0.5	52.4±0.5	51.2±0.2	56.7±1.4	50.7±0.8	58.9±0.3
PA	75.0±0.3	52.4±0.5	54.3±0.5	71.6±2.6	54.6±2.4	44.2±01.0
SHP	72.6±0.5	62.6±0.4	51.7±0.3	67.3±1.8	51.2±1.4	65.9±1.5
Katz	72.7±0.5	60.8±0.7	51.7±0.3	67.3±1.7	51.2±1.4	65.5±1.6
Sup-Top	75.4±0.3	62.8±0.1	54.2±0.7	72.0±2.0	69.5±7.6	70.8±6.2
SVD	63.5±0.3	53.8±1.7	51.2±3.1	62.9±5.1	54.1±9.4	69.1±2.6
Fact+Scores	79.3±0.5	69.6±0.2	61.3±1.9	81.2±2.0	68.9±4.2	75.1±2.0
JNSL	80.1±0.8	72.5±0.7	62.1±0.8	88.6±2.3	72.0±0.9	75.6±0.7

5 Conclusion and Future Work

In this paper, we have proposed an effective method for link prediction. Our method takes advantages from the way to represent links as joint neighborhood subgraphs and the employment of a convolution kernel that is able to efficiently exploit the graph structure. The results from the experiment show that our method is the state of the art for graph structured link prediction, especially when the side information are not available.

For the future work, we plan to investigate to propose a method for link prediction which can make use of the variety of information sources associated with graphs.

References

1. Adamic, L.A., and Eytan, A.: Friends and neighbors on the web. Social networks 25.3 (2003): 211-230.

2. Katz, L.: A new status index derived from sociometric analysis. *Psychometrika* 18.1 (1953): 39-43.
3. Leicht, E.A., et al.: Vertex similarity in networks. *Physical Review E* 73.2 (2006): 026120.
4. Barabasi, A.L, Albert, R.: Emergence of Scaling in Random Networks, *Science* 286 (1999) 509.
5. Menon, A., and Charles, E.: Link prediction via matrix factorization. *Machine Learning and Knowledge Discovery in Databases* (2011): 437-452.
6. Von. M.C, et al.: Comparative assessment of large-scale data sets of proteinprotein interactions. *Nature* 417.6887 (2002): 399-403.
7. Yamanishi, Y., et al.: Supervised enzyme network inference from the integration of genomic data and chemical information. *Bioinformatics* 21.suppl-1 (2005): i468-i477.
8. Rowies, S.: NIPS dataset (2002), <http://www.cs.nyu.edu/~rwoeis/data.html>
9. Lichtenwalter, R. N., et al.: New perspectives and methods in link prediction. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010.
10. Ghosn, F., et al.: The MID3 data set, 19932001: Procedures, coding rules, and description. *Conflict Management and Peace Science* 21.2 (2004): 133-154.
11. Ward, M.D., et al.: Disputes, democracies, and dependencies: A reexamination of the Kantian peace. *American Journal of Political Science* 51.3 (2007): 583-601.
12. Watts, D.J., and Steven, H.S.: Collective dynamics of small-worldnetworks. *nature* 393.6684 (1998): 440-442.
13. Costa, F, and Kurt, D.G.: Fast neighborhood subgraph pairwise distance kernel. *Proceedings of the 26th International Conference on Machine Learning*. Omnipress, 2010.
14. Haussler, D.: Convolution kernels on discrete structures. Vol. 646. Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.