Joint Neighborhood Subgraphs Link Prediction

Dinh Tran-Van[1], Alessandro Sperduti[1], and Fabrizio Costa[2]

[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 recomendation 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 promissing 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 decompositional graph kernel to measure the similarity between links. An empirical experiment on diferent datasets proves that our proposed method shows the state of the art for link prediction.

Keywords: Link prediction, joint neighborhood subgraphs

1 Introduction and related work

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 struture 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 associtated 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 the term link prediction methods to refer to topology-based link prediction methods. Conerning 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. In [1], a method, *Adamic-Adar*, is proposed in which a score between two nodes is computed by a weighted sum over their common nodes. Nodes with smaller degree are weighted more comparing to the higher ones. Another method also considers node degree when computing proximity is presented. The probability of a existing link connecting two nodes is proportional to the product of their node degrees. Different from the method in [1] which considers the common nodes, a link prediction method, *Katz*, is introduced in [2] that takes into acount the number of common paths with different lengths between two nodes to measure the proximity between them. The shorter paths are given more weights, meanwhile the longer ones are weighted less. Another method which is a modification of *Katz* notated as Leicht-Holme-Newman is presented in [3]. In this method the similarity of a node couple is based on an assumption that two nodes are more similar if they have more similar intermediate nodes. Besides, in [5] they also apply a raw singular value decomposition, SVD, on the adjacency matrix for link prediction. Generally, unsupervised learning methods takes the advantage from their complexity since they do not face with learning process. However, in many case, they do not show impressing results.

Supervised learning link prediction methods are based on a learning process which try to learn a curve that separates present links and absent links relied on their characteristics. 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]. In [6], a nonparametric latent feature models for link prediction is introduced in which latent features are extracted by based on an adaptation of Bayesain nonparametric approach. Besides, this method is able to concurrently learn number of latent features and assign which entities have these features. Another method related to latent features is presented in [5], they propose to use matrix factorization to extract latent features. This method can also take the output of unsupervised methods as its input to proceed. They show impressive results with different use of loss functions: square loss function, log loss function. Compared to unsupervised learning methods, supervised methods tend to show better performance in general. However, they normally face with the high compexity 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 these corresponding nodes. It then adopts a decompositional 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 devided into a subset of observed links (O) and a subset of unobserved links (U). It is often that the set O is not a complete set since they are many actual links are missing. Therefore, finding most likely set of links in U to be considered as the complement set of O is an important task which impacts on the performance of graph-based systems. 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.

We define 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* of a node u with radius r is the set of nodes such that they are at distance no greater than r from u and it is denoted by $N_r(u)$ and $N_r(u) = \{v \mid d(u, v) \leq r\}$. The *neighborhood subgraph* with radius r of a given node u, denoted by $\mathcal{N}_r^u$, is the subgraph formed by the nodes in the *neighborhood* with radius r of u and the relative edges with endpoints in $N_r(u)$. The degree of a node u denoted as $d(u)$ is the cardinality of its neighborhood $d(u) = |\mathcal{N}_1^u|$. We also notate the maximum node degree in the graph G as $d(G)$, $d(G) = max\{d(u) \mid u \in V\}$

2.2 The neighborhood subgraph pairwise distance kernel

In this section, we briefly desbribe an efficient kernel named NSPDK [14] which allows to measure the similarity between labeled graphs. NSPDK is an instance of decompositional kernel [15].

Given a labeled graph $G \in \mathcal{G}$ and two rooted graphs A_u, B_v, we first define the relation $R_{r,d}(A_u, B_v, G)$ to be true *iff* $A_u \cong \mathcal{N}_r^u$ is (up to isomorphism $\cong$) a neighborhood subgraph with radius r of G and so is $B_v \cong \mathcal{N}_r^v$, such that v is a distance d from u: $\mathcal{D}(u, v) = d$. We then define the inverse relation R^{-1} that returns all pairs of neighborhoods of radius r at distance d in G, $R_{r,d}^{-1}(G) = \{A_u, B_v | R_{r,d}(A_u, B_v, G) = true\}$. The kernel $\kappa_{r,d}$ over $\mathcal{G} \times \mathcal{G}$ is 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 $K(G,G') = \sum\limits_r \sum\limits_d \kappa_{r,d}(G,G')$, where for efficiency reasons, the values of r and d are upper bounded to a given maximal r^* and d^*, respectively.

Related to exact matching function, this function needs to solve the graph isomorphism problem. This problem is not known to be solvable in polynomial time nor to be NP-complete. In [14], they propose an approximate method to figure out graph isomorphism problem. It first transforms graphs into strings. It then employ a hash function to map each unique string with an integer number. Finally, two graphs are isomorphic if they have the same corresponding integer numbers.

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 a node labeling fuction ℓ such that nodes with similar degrees are assigned the same label. It is due to the assumption that nodes with similar degrees tend to share common properties. Following is the definition of the the function:

$$\ell(u) = \begin{cases} d(u), \text{ if } d(u) \leq 5 \\ 6+i, \text{ if } 5 + \frac{d(G)-5}{\lambda-5} * i \leq d(u) < 5 + \frac{d(G)-5}{\lambda-5} * (i+1) \end{cases},$$

where $\lambda > 5$ is the number of labels used to label for nodes in the graphs. The values of λ depend on the histogram of graph node degree. We can use tuning technique in order to obtain suitable value for λ.

2.4 Link representation

Most methods for link prediction attempt to computes similarities between nodes and consider to rank links. We desire to take links as examples to learn and predict. Therefore, we propse a method which model the context of two nodes to represent for link connecting them.

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 create an artificial node w. This node is connected to both u and v. As a consequence, we achieve a single graph representing for a link connecting u and v.

2.5 Joint neighborhood subgraphs link prediction

Given a graph $G(V,E)$ and a set of links $L = \{(u,v) \mid (u,v) \in\ O \ or\ (u,v) \in U\}$, 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* [7]: the medium confidence network which encodes the interaction between proteins. It contains 2617 nodes and 11855 links.
- *Metabolic* [8]: a biological network which relates enzyme proteins and chemical compounds and it includes 668 nodes and 2782 links.
- *Nips* [9]: a network of co-authors at NIPS conference from first to twelfth edition. Nodes are authors and links encode co-author relation between authors. This network contains 2865 nodes and 4733 links.
- *Condmat* [10]: a network of co-authors for condensed matter physists. This network has 14230 nodes and 1196 links.
- *Conflict* [11], [12]: a network describing the dispute between 130 countries. A link connecting two nodes (countries) if they dispute. We have 180 links in total in this network.
- *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 *Condmat* 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 in node label function in $\{10, 15\}$, 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 in which we mark as bold for the best ROC-AUC on each dataset. 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 unsuperivsed 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 ROC-AUC performance on six datasets of different link prediction methods in which AA: Adamic-Adar, PA [4] preferential Attachment, SHP: Shortest Path, Sup-Top: Liear 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 an 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. Miller, K., Michael, et al.: Nonparametric latent feature models for link prediction. Advances in neural information processing systems. 2009.
7. Von. M.C, et al.: Comparative assessment of large-scale data sets of proteinprotein interactions. Nature 417.6887 (2002): 399-403.
8. Yamanishi, Y., et al.: Supervised enzyme network inference from the integration of genomic data and chemical information. Bioinformatics 21.suppl-1 (2005): i468-i477.
9. Rowies, S.: NIPS dataset (2002), http://www.cs.nyu.edu/ rwoeis/data.html
10. 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.
11. Ghosn, F., et al.: The MID3 data set, 19932001: Procedures, coding rules, and description. Conflict Management and Peace Science 21.2 (2004): 133-154.
12. 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.
13. Watts, D.J., and Steven, H.S.: Collective dynamics of small-worldnetworks. nature 393.6684 (1998): 440-442.
14. Costa, F, and Kurt, D.G.: Fast neighborhood subgraph pairwise distance kernel. Proceedings of the 26th International Conference on Machine Learning. Omnipress, 2010.
15. Haussler, D.: Convolution kernels on discrete structures. Vol. 646. Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.