JNSL - Joint Neighborhood Subgraph Link Prediction Method

Anonymous Author(s)
Affiliation
Address
email

Abstract

Graphs are common data structure used to present biological data in which nodes describe entities and links characterize their relationships. In many cases, there are many links are missing due to the lack of knowledge, it causes the problem for further studies and for graph-based learning systems. As a consequence, a number of link prediction methods have been proposed to discover these missing links. However, most existing link prediction methods are based on the transitive manner which first cannot effectively exploit graph structures and second do not allow to recover links whose end points locating in different isolated components. Therefore, they only show limited performance. In this paper, we propose a joint neighborhood subgraph link prediction method that not only can fully exploit the graph structure by dealing with isomorphism of joint neighborhood subgraphs associated to links but also make predictions based on a learning process. Moreover, our method is able to recover links connecting isolated components. The performance on several data sets confirms that our method shows the state of the art performance for link prediction.

1 Introduction

In the last decade, thanks to the advent of high-thoughput technology, a huge amout of biological data have been made available. Such hugh data bring a golden opportunity for us to explore and extract important information. This allows to expand our understanding about the nature of biological processes which have been (partially) hidden. Biological processes are usually represented in the form of networks whose nodes represent a set of entities and links characterize entities' relatiionships. Human Protein Reference Database (HPRD) [1] is an instance in which a node depicts a gene and a link is formed to connect two genes if their corresponding proteins interact. Despite the fact that a hugh data are accessible, most networks are still far from being complete. It is demonstrated with the sparity of networks whose limited number of present links and high number of missing links. This prevents computational systems and machine learning techniques with the aim of extracting and inferring knowledge from data, from showing high performance since they normally require a certain amount of data to train models. Link prediction is a solution for this situation. Link prediction aims at recovering the missing links by making a ranking over non-observed links based on their likelihood to be links. It can be directly used to uncover relationship between entities or used to enrich links of graphs by adding the top non-observed links into graphs. These obtained graphs then are used as the input of graph-based learning systems.

Regarding link prediction in general, a number of methods have been proposed and applied in different domains ranging from Recommandation systems to Bioinformatics, to social networks. In [2], a quite exhausive survey of link predictions methods is presented. Most popular link prediction methods are in the similarity-based class in which each non-observed link is assigned a score and this score is directly used as the similarity between starting and ending nodes of the link. For example,

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

Leicht-Holme-Newman [3], Local random walk [7] and Local path index [8] are similarity-based algorithms where the similarity between nodes are computed by considering the local paths. However, these methods are based on the transitive manner which cannot effectively exploit graph structures and the prediction is made without using any learing process. Moreover, it is impossible to have non-zero similarities measured for non-observed links whose end points located in different disconected component of the graph. Therefore, they often show limited performance.

In this paper, we propose a novel method for link prediction. In our method, we present each possible link by a joint neighborhood subgraph formed from the starting and ending node's neighborhood graphs. These graphs are used to build a model in which the graph similarity measurement is computed by employing NSPDK kernel [16] whose implementation of an approximation for graph isomorphism. Once the model is constructed, it is used to make prediction. Our method not only is able to effectively exploit the graph structure, but also able to distingush between positive and negative links by learning process. Interestingly, it allows to have un-null values for non-observed links connecting points between two isolated components. Empirical experiments in three different biological data sets illustrate the effectiveness of our proposed link prediction methods.

This paper is organized as follows: first the background and notations are presented in Section 2. In Section 3, the proposed method is presented. Next, the evaluation followed by results and discussion are shown in Section 4 and 5, respectively. Finally, we make conclusion and plan for coming work in Section 6.

2 Notation and background

In this section we first present notations and definitions used in this paper. We then briefly describe link prediction problem and some popular link prediction methods.

2.1 Notation and Definitions

A graph $G = (V, E)$ is a structure in which V is a node set and E is a link set. The adjacency matrix, A, is a symmetric whose entry $a_{ij} = 1$ if there is a direct link connecting $v_i \in V$ and $v_j \in V$, v_i and v_j are vertices corresponding to indices i and j, respectively, and $a_{ij} = 0$ otherwise. The Laplacian matrix L is defined as $L = D - A$, where D is the diagonal matrix with non-null entries equal to the summation over the corresponding row of the adjacency matrix, i.e. $D_{ii} = \sum_j A_{ij}$. 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 $N_r(v)$.

2.2 Link prediction

Link prediction is the task which aims at recovering missing links of graphs or predicting links that are going to be present in the future state of an evolving graph. A link prediction algorithm intends to make a ranking over all non-observed links from the most to the least probable by estimating scores associated to them. Link prediction can be used in two cases. It can be used directly, in the first case, to disclose the hidden relationships between entities presented as nodes in the graphs. In the second case, it is used to enrich links in the sparse graphs that are fit into graph-based learning systems.

A considered number of link prediction methods which have been proposed in literature and have been applied to different domains ranging from recommendation systems, to bioinformatics, to social networks. Following [2], we can classify these methods in: *similarity-based algorithms*, *maximum likelihood methods*, and *probabilistic models*. Following are descriptions of these categories.

- *Similarity-based algorithms:* This is the most straightforward framework for link prediction. In these methods, each couple of nodes is assigned a score which is directly used as the proximity between between them. Since the nodes' attributes are normally hidden, most similarity-based methods work by considering structures of networks.
- *Maximum likelihood methods:* In these algorithms, we assume some configurations of the network structure, rules and parameters obtained by maximizing the likelihood of the

observed structure. We then compute the likelihood for non-observed links based on those rules and parameters. Hierachical structure [4] and Stochastic block [5] are typical models for this group of link prediction methods.

- *Probabilistic models:* For probabilistic models, we desire to abstract network structures. In order to predict the likelihood for non-observed links, we construct a probabilistic model that best fit observed data by optimizing a predefined target function. Probabilistic relational [6] is an example for probabilistic models.

Most popular link prediction methods fall into similarity-based group since they are simpler and computationally more efficient than maximum likelihood and probabilistic models. Among similarity-based methods, global ones normally outperform local algorithms because global methods are able to capture the glocal similarity between nodes of graph. Below we first introduce a set of diffusion-based graph node kernels which can be used as global prediction methods. We then shortly describe some local ones.

A graph node kernel is a kernel which defines the similarity between nodes in a graph. Formally, a graph node kernel, $k(\cdot,\cdot)$, is defined as $k : V \times V \longrightarrow \mathbb{R}$ such that k is symmetric positive semidefinite. Graph node kernels have been successfully applied in various domains ranging from recommendation systems to disease gene prioritization. Diffusion-based graph kernels are kernels that define similarity between any couple of nodes by condering paths connecting them. Given a graph, in order to use a graph node kernel for link prediction we first use kernel to compute a kernel matrix which encodes the similarity between any couple of nodes of the graph. The values inside the achieved matrix are used as the scores for non-observed links to be considered as link of the graph. Following are descriptions of some popular graph node kernels.

- *Laplacian exponential diffusion kernel (LEDK)* [12]: This kernel is based on heat diffusion phenomenon: imagine to initialize each vertex with a given amount of heat and let it flow through the edges until an arbitrary instant of time. The similarity between any vertex couple v_i, v_j is the amount of heat starting from v_i and reaching v_j within a given time. The LEDK kernel matrix is computed by:
$$K_{LEDK} = e^{-\beta L} \,, \tag{1}$$
where β is the diffusion parameter used to control the rate of diffusion, and $e^X = \sum_{k=0}^{\infty} \frac{1}{k!} X^k$ refers to the matrix exponential for matrix X. Choosing a consistent value for β is very important: on the one side, if β is too small, the local information cannot be diffused effectively and, on the other side, if it is too large, the local information will be lost. K_{LEDK} is positive semi- definite as proved in [12].
- *Markov exponential diffusion kernel (MEDK)* [13]: In LEDK, similarity values between high degree vertices is generally higher compared to that between low degree ones. This could be problematic since peripheral nodes have unbalanced similarities with respect to central nodes. To make the strength of individual vertices comparable, a modified version of LEDK is introduced:
$$K_{MEDK} = e^{-\beta M} \,, \tag{2}$$
where $M = (D - A - nI)/n$ and *n*, *I* are the total number of vertices in graph and identity matrix, respectively.
- *Markov diffusion kernel (MDK)* [14]: MDK exploits the idea of diffusion distance, which is a measure of how similar the pattern of heat diffusion is between a pair of initialized nodes. In other words, it expresses how much nodes "influence" each other in a similar fashion. From the transition matrix *P* ($P = D^{-1}A$), we define $Z(t) = \frac{1}{t}\sum_{\tau=1}^{t} P^{\tau}$. MDK kernel matrix is then computed as follows:
$$K_{MDK} = Z(t)Z^{\top}(t) \,. \tag{3}$$
- *Regularized Laplacian kernel (RLK)* [15]: It represents a normalized version of the random walk with restart model. The kernel matrix is defined as:
$$K_{RLK} = \sum_{n=0}^{\infty} \beta^n (-L)^n \,, \tag{4}$$
where β is again the diffusion parameter. RLK counts the paths connecting two nodes on the graph induced by taking *-L* as the adjacency matrix, regardless of the path length. Thus,

a non-zero value is assigned to any couple of nodes as long as they are connected by any indirect path.

Local link prediction methods only require local topological information of networks. These methods normally show lower performance comparing to glocal methods, but they require less time computation. In the case of graphs with small average shortest distance, they show good performance and are comparable with global ones. Following we describe some local methods of similarity-based group.

- Local path index (LPI) [8, 9]: Similar to Common neighbors method [11] which considers local paths with length 2, LPI takes into account paths with length 2 connecting two nodes to form their similarity. However, it also includes paths with length 3. Therefore, it is able to capture wider topological information.

$$s_{uv} = A^2 + \varepsilon A^3 \ , \tag{5}$$

  where s_{uv} is the similarity between u and v, ε is a free parameter which allow to assign weight for paths with length 3.

- Local random walk index (LRWI) [10]: Consider a couple of nodes u and v in graph, to measure the similarity between them, we first let a random walker start from u and we have initial density vector $\overrightarrow{\pi}_u(0) = \overrightarrow{e}_u$, $\overrightarrow{e}_u$ is a vector whose u^{th} element equals to 1 and the rest equal to zero. This density vector at time $t+1$ is computed by $\overrightarrow{\pi}_u(t+1) = P^t \overrightarrow{\pi}_u(t)$. We then define LRWI at time t as:

$$s_{uv}(t) = q_u \overrightarrow{\pi}_{uv}(t) + q_v \overrightarrow{\pi}_{vu}(t) \ , \tag{6}$$

  where $q_u = \frac{k_u}{M}$ with k_u is degree of u and M is the totall number of links in the graph.

- Superposed random walk index (SRWI) [10]: SRWI is an extension of LRWI in which a random walker is continuously released at the starting point. It can be computed as follow:

$$s_{uv}(t) = \sum_{\tau}^{t} \left(q_u \overrightarrow{\pi}_{uv}(t) + q_v \overrightarrow{\pi}_{vu}(t) \right) . \tag{7}$$

  SRWI increases the similartities between nodes nearly located in the graph and it requires higher time complexity comparing to LRWI.

3 Method

Here we describe in detail our proposed method for link prediction and show how can we apply the method to predict links given a graph.

3.1 Node labeling

It is common that gene identifier is used to label for nodes of genetic graphs. However, in our method, we employ the node labeling procedure used in [19] for our labeling process since we desire to build a system which is based on the configuration of each gene's context. Particularly, we use as labels a discretized functional annotation based on the *gene ontology* [18]. More precisely, we use the gene ontology to construct binary vectors representing the bag-of-words encoding for each gene (i.e. if a GO-term is associated with the gene). The resulting representations are then clustered so that genes with similar description profiles receive the same class identifier as label. Finally, node labels are used to define the graph isomorphism problem.

3.2 Link representation

In this section, we describe the procedure on we represent a link between u and v of the graph G. Given a radius r, we first extract neighborhood subgraphs N_r^u and N_r^v rooted at u and v, respectively. We then take a union of these two obtained subgraphs to have a joint neighborhood subgraphs G_{uv}^r where $G_r^{uv} = N_r^u \bigoplus N_r^v$. If the link (u, v) is a non-observed link, we add (u, v) into G_r^{uv}. Finally, we label for links in G_r^{uv} by assigning a label "*a*" for the link (u, v) and a label "*b*" for the rest of links.

3.3 The Neighborhood Subgraph Pairwise Distance Kernel

To measure the similarity between graphs, we adopt a kernel named NSPDK proposed in [16]. This kernel is an instance of convolution kernel [17] 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, $R_{r,d}^{-1}(G) = \{A_u, B_v | R_{r,d}(A_u, B_v, G) = true\}$. 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 r^* and d^*, respectively.

3.4 Joint neighborhood subgraph link prediction

In genetic graphs, it is difficult to state that there is no relation between a couple of nodes u and v. Therefore, it is hard or even infeasible to form a negative link set which is normally used to train a binary classification model. This problem can be solved by using a oneclass classification model which only needs a positive link set for the learning process (training model).

Given a graph $G = (V, E)$ and a radius r, we first extract a joint neighborhood subgraph G_{uv}^r for each possible link (u, v) of G. As a consequence, we achieve a positive subgraph set G_E^r which represents for links in E and a non-observed subgraph set, G_N^r, representing for non-observed link set, respectively. The positive subgraph set G_E^r is then used as the input to build a one-class classification model using NSPDK kernel. Once the model is trained, it returns a score s_{uv} for each each non-observed link (u, v). The score s_{uv} shows how linkly of (u, v) to be related to positive links.

4 Empirical evaluation

To empirically evaluate the ability of different link prediction methods described in 2.2 to recover the missing links, we use sparse graphs induced from three datasets described below.

- **HPRD:** a database of curated proteomic information pertaining to human proteins. It is derived from [1] with 9,465 vertices and 37,039 edges. We employ the HPRD version used in [19] that contains 7311 nodes and 30503 edges.
- **Phenotype similarity:** we use the OMIM [20] dataset and the phenotype similarity notion introduced by Van Driel et al. [1] based on the relevance and the frequency of the Medical Subject Headings (MeSH) vocabulary terms in OMIM documents. We built the graph linking those genes whose associated phenotypes have a maximal phenotypic similarity greater than a fixed cut-off value. Following [1], we set the similarity cut-off to 0.3. The resulting graph has 3393 nodes and 144739 edges.
- **Biogrigen** Genetic interaction is the phenomenon through which the effects of a gene are modified by one or several other genes. This occurs in indirect way by means of knock-on effects of multiple physical interactions. In practice, this is observed when the effects of two mutations in distinct genes is not equal to the sum of the effects of the mutations alone. This kind of interaction is complementary in respect to the physical one and is important especially for complex diseases involving a large number of genes. To construct the graph, we form a link between two genes if they interact.

Evaluation method

Consider a graph $G = (V, E)$, in order to evaluate the performance of link predictions, we construct a positive set P with all links in E and an un-observed set, U, formed by randomly sample fromm

un-observed links of the graph such that $|\mathcal{P}| = \frac{1}{2}|\mathcal{U}|$. We replicate this procedure 10 times. We assess the performance of link predictions via a 10-fold cross valdation on the positive set. Each round, only positive links in one fold are kept in the graphand links in the remaining 9 folds are removed from the graph, so they are converted to non-observed links. The obtained graph is used to compute the similarities for links in test set (9 folds). For JNSL, first a model is built from links in one fold by using a kernel machine SVMs with the use of NSPDK kernel. Then it is used to return scores for links in the 9 remaining folds. We collect all scores and compute the area under the curve for the receiver operating characteristics (AUC-ROC). The final performance is obtained by taking the average over (10-folds $\times$ 10) trials.

Model selections

The hyper-parameter of different methods are set by using a 3-fold cross-validation in which one fold is used for training and two other folds are used for validating. We try the values for diffusion parameter of LEDK and MEDK in $\{0.01,\ 0.05,\ 0.1\}$, time steps in MDK in $\{3,\ 5,\ 10\}$ and RLK parameter in $\{0.01,\ 0.1,\ 1\}$, free parameter in LPI in $\{0.1,\ 0.3,\ 0.5\}$, time step in LRWI and SRWI in $\{5,\ 7\}$. For NSPDK, we try for maximum radius in $\{1,\ 2\}$, maximum distance in $\{2,\ 3,\ 4\}$. Finally, the fraction of training errors nu for One-class SVM is chosen in $\{0.01,\ 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5\}$ and regularization trade-off C for binary SVM in $\{10^{-4},\ 10^{-3},\ 10^{-2},\ 10^{-1},\ 0,\ 10,\ 10^{2},\ 10^{3},\ 10^{4}\}$.

5 Results and discussion

Table 1: The performance in AUC-ROC (%) of different link prediction methods on graphs induced from three datasets: Biogridgen, Omim and HPRD.

	Datasets		
Methods	Biogridgen	Omim	HPRD
LEDK	50.6±0.4	66.2±1.6	58.0±1.0
MEDK	50.0±0.0	50.1±0.0	50.0±0.0
MDK	50.7±0.6	68.1±1.8	64.9±0.4
RLK	50.7±0.6	68.1±1.8	63.5±0.5
LPI	50.2±0.3	60.6±1.1	52.8±0.2
LRWI	50.2±0.3	67.2±1.7	54.8±0.3
SRWI	50.2±0.3	66.2±1.6	56.7±1.0
JNSL1	73.4±2.7	69.9±1.4	72.8±0.5
JNSL2	**79.9±1.9**	**72.5±1.3**	**78.6±0.3**

Table 1 shows the synthesis results of our experiments. In the table, rows prepresent different link prediction methods, meanwhile columns illustrate different datasets (Biogridgen, Omim and HPRD) used to construct corresponding graphs. Here we report aggregated results which have averaged across a random choice of negative sets. Overall, our proposed method outperforms all compared link prediction methods on all data sets. In particular, it shows a remarkable difference (about 30%) comparing other methods on Biogridgen data set. In addition, it performs about 14.7% and 4.4% higher than the second best one in HPRD and Omim, respectively.

6 Conclusion and future work

In this paper, we have proposed an effective, novel link prediction method. Our method owns two strong points that help it to be a promissing link prediction method. The first point is the presentation of links. Each link is represented by a joint neighborhood subgraph which efficiently reflects its context in the graph. The second point is the adoption of a learning process before making predictions. Empirical experimental results shows that our proposed method is the state of the art in link prediction.

References

[1] Van Driel, Marc A., et al.: A text-mining analysis of the human phenome." European journal of human genetics 14.5 (2006): 535-542.

[2] Lu, L., and Tao Z.: Link prediction in complex networks: A survey. Physica A: Statistical Mechanics and its Applications 390.6 (2011): 1150-1170.

[3] Leicht, E., et al: Vertex similarity in networks. Physical Review E 73.2 (2006): 026120.

[4] Redner, S.: Networks: teasing out the missing links. Nature 453.7191 (2008): 47-48.

[5] White, H.C., et al.: Social structure from multiple networks. I. Blockmodels of roles and positions. American journal of sociology 81.4 (1976): 730-780.

[6] Neville, J.: Statistical models and analysis techniques for learning in relational data. Diss. University of Massachusetts Amherst, 2006.

[7] Liu, W., and Linyuan L.: Link prediction based on local random walk. EPL (Europhysics Letters) 89.5 (2010): 58007.

[8] Linyuan, L., et al: Similarity index based on local paths for link prediction of complex networks. Physical Review E 80.4 (2009): 046122.

[9] Zhou, T., et al.: Predicting missing links via local information. The European Physical Journal B-Condensed Matter and Complex Systems 71.4 (2009): 623-630.

[10] Linyuan, L., et al.: Similarity index based on local paths for link prediction of complex networks. Physical Review E 80.4 (2009): 046122.

[11] Newman, M.E.: Clustering and preferential attachment in growing networks. Physical review E 64.2 (2001): 025102.

[12] Kondor, Risi Imre, and John Lafferty, Diffusion kernels on graphs and other discrete input spaces. ICML. Vol. 2. 2002.

[13] Chen. B, et al. Disease gene identification by using graph kernels and Markov random fields. Science China Life Sciences 57.11 (2014): 1054-1063.

[14] Fouss F, et al. An experimental investigation of graph kernels on a collaborative recommendation task. Proceedings of the 6th international conference on data mining 2006. ICDM 2006, 863-868.

[15] Chebotarev P and Shamis E. The matrix forest therem and measuring relations in small social groups. Automation and Remote Control 1997, 58(9):1505-1514

[16] C. Fabrizio, and K. De Grave, Fast neighborhood subgraph pairwise distance kernel. Proceedings of the 26th, International Conference on Machine Learning. Omnipress, 2010.

[17] Haussler, David. Convolution kernels on discrete structures. Vol. 646. Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.

[18] Gene Ontology Consortium. The Gene Ontology (GO) database and informatics resource. Nucleic acids research,(2004) 32(suppl 1), D258-D261.

[19] Tran-Van, D., Sperduti, A., and Costa, F.: Conjunctive disjunctive node kernel. Proceedings of 25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2017.

[20] McKusick, Victor A.: Mendelian Inheritance in Man and its online version, OMIM. The American Journal of Human Genetics 80.4 (2007): 588-604.