Link Enrichment for Strengthening Diffusion-based Graph Node Kernels

Dinh Tran-Van, Alessandro Sperduti, and Fabrizio Costa

Department of Mathematics, Padova University Trieste, 63, 35121 Padova, Italy
Department of Computer Science, University of Exeter Exeter EX4 4QF, UK
{dinh, sperduti}@math.unipd.it, f.costa@exeter.ac.uk
http://www.math.unipd.it

Abstract. Node similarity measurement is one of the key points which determines the performance of graph-based learning systems. Diffusion-based graph node kernels are commonly used in many applications to capture the node similarity. However, they only show promissing results in the case of using dense graphs. In this paper, we propose a method that aims to strengthen diffusion-based kernels when working with sparse graphs by employing link enrichment. The empirical experiments show that our method considerably improves the power of diffusion-based graph node kernels.

Keywords: Graph node kernels, diffusion-based kernels, strenghthening diffusion-based kernels, link enrichment

1 Introduction

Recently, with the fast development of science and technology, we have witnessed the rapid growth of data in terms of both volume and variety. In order to efficiently extract knowledge from those huge data, a number of learning systems have been introduced. Each system takes specific types of data to preceed. Graph is a widely used data representation and is employed in many systems of different domains [1], [8]. Learning systems used graphs as their input are referred as graph-based learning systems.

In graph-based systems, the measurement of node proximity is one of the key factor that determines the performace of the systems. The most common paradigm used to capture node similarity is graph node kernels. Graph node kernel is a paradigm which allows to define similarity between any couple of graph nodes in a normally high dimensional space. As a consequence, considerable graph node kernels have been proposed and applied in many applications, domains. Among them, diffusion-based kernels [1] are the most commonly employed and they show promissing results. However, those node kernels usually show good performance only when dealing with dense graphs - graphs with high

[1] A diffusion-based graph node kernel measures the proximity between any couple of nodes by taking into account paths connecting them.

value of average node degree node. And, vice versa in the case of working with sparse graphs - graphs with low value of average node degree node - they usually lead to poor performance of systems. This is due to (1) the number of links in the graph is very limitted comparing to the complete graph, so the information cannot be well diffused. (2) the lack of links also causes the fragment problem to the graph. In this case, the information cannot diffused between isolated components. Therefore, the similarity between nodes located in different isolated components measured by diffusion-based graph node kernels is equal to zero. As a consequence, it raises challenge for us if we desire to build good graph-based learning systems. To overcome this problem, we come up with the idea of using link enrichment. Link enrichment is a task that aims at predicting the most probable candidate links to be considered as missing links to add into graphs. Many link prediction methods have been proposed. In [9], a roughly exhaused study of link prediction methods is presented in which they classify methods into different groups. The most widely used framework is the similarity-based algorithms because of the ease and effectiveness. In those methods, each pair of nodes is assigned a score which is directly defined as the similarity between nodes.

To the best of our knowledge, there is no investigation that has been done to boost the performance of diffusion-based kernels by using link enrichment. Therefore, in this paper, we present a method that intends to strengthen the power of diffusion-based graph node kernels by employing link enrichment paradigm. An evaluation on different real datasets confirms that our proposed method is notable when using diffusion-based graph node kernels.

This paper is organized as follows: we first introduce the notaion and background in the section 2. We then describe our proposed method in section 3. The evaluation and results are presented in section 4 and section 5, respectively. Finally, the conclusion is writen in section 6.

2 Notation and Background

Let us consider an undirected graph $G = (V, E)$ in which V represents for a set of entities (vertices) and E characterizes the entity relationships (links). The adjacency matrix A is a symmetric matrix used to describe the direct links between vertices v_i and v_j in the graph. Any entry A_{ij} is equal to 1 when there exists a link connecting v_i and v_j, and is 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 rest of the paper are described under this notation convention.

2.1 Graph Node Kernels

As the desire of having a good node similarity measure for building graph-based leanring systems, many graph node kernels have been introduced and applied.

A graph node kernel is a kernel which defines the similarity between nodes in a graph. Formally, a graph node kernel, k, is defined as:

$$k(u, v) =< \phi(u), \phi(v) >, \tag{1}$$

where u, v are nodes in the graph and k needs to be symmetric positive semidefinite. Most graph node kernels belong to one of the two popular frameworks: Diffusion-based graph node kernels and decomposition graph node kernels.

Diffusion-based kernels can be considered as the modifications of laplacian diffusion kernel [2]. These kernels measure the node proximity between any couple of nodes by taking into account the paths connecting them. They normally show promissing performance in case of dealing with dense graphs because of their ability to capture the global similarity. However, they poorly demonstrate when working with sparse graphs which consist of a low number number of links and high number of disconnected components. Following, we briefly describe some of the most popular diffusion-based graph node kernels.

- *Laplacian exponential diffusion kernel:* One of the most well-known kernels for graphs is the Laplacian exponential diffusion kernel (LEDK), as it is widely used for exploiting discrete structures in general and graphs in particular. On the basis of the heat diffusion dynamics, Kondor and Lafferty proposed LEDK in [2]: 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 the given time. Therefore, LEDK can capture the long range relationship between vertices of a graph to define the global similarities. Below is the formula to compute LEDK values:

  $$K_{LEDK} = e^{-\beta L}, \tag{2}$$

  where β is the diffusion parameter and is used to control the rate of diffusion. 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 [2].
- *Markov exponential diffusion kernel:* In LEDK, similarity values between high degree vertices is generally higher compared to that between low degree ones. Intuitively, the more paths connect two vertices, the more heat can flow between them. This could be problematic since peripheral nodes have unbalanced similarities with respect to central nodes. In order to make the strength of individual vertices comparable, a modified version of LEDK is introduced by Chen et al in [3], called Markov exponential diffusion kernel MEDK and given by the following formula:

  $$K_{MEDK} = e^{-\beta M}. \tag{3}$$

  The difference with respect to the LEDK is the replacement of L by the matrix $M = (D - A - nI)/n$ where n is the total number of vertices in graph. The role of β is the same as for LEDK.

- *Markov diffusion kernel:* The original Markov diffusion kernel MDK is introduced by Fouss et al. [10]. It exploits the idea of diffusion distance, which is a measure of how similar the pattern of heat diffusion is among a pair of initialized nodes. In other words, it expresses how much nodes "influence" each other in a similar fashion. If their diffusion ways are alike, the similarity will be high and, vice versa, it will be low if they diffuse differently. This kernel is computed starting from the transition matrix P and by defining $Z(t) = \frac{1}{t}\sum_{\tau=1}^{t} P^{\tau}$, as follows:

$$K_{MDK} = Z(t)Z^{\top}(t) \tag{4}$$

- *Regularized Laplacian kernel:* Another popular graph node kernel function used in graph mining is the regularized Laplacian kernel (RLK). This kernel function was introduced by Chebotarev and Shamis in [4] and represents a normalized version of the random walk with restart model. It is defined as follows:

$$K_{RLK} = \sum_{n=0}^{\infty} \beta^n (-L)^n \tag{5}$$

  where the parameter 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. K_{RLK} remains a relatedness measure even when diffusion factor is large, by virtue of the negative weights assigned to self-loops.

Decomposition graph node kernels take the idea from [5] in which the similarity function between two graphs can be formed by decomposing each graph into subgraphs and by devising a valid local kernel between the subgraphs. This idea is then adjusted to measure graph node similarity by considering the neighborhood subgraph rooted at a vertex as its graph to compute. In order to form this kind of kernel, we need to face with graph matching problem or graph isomorphic problem which is not known to be solvable in polynomial time nor to be NP-complete. An advantage of using decomposition kernels is the possibility to have non-zero similarity value for node couples locate in distinct disconnected components of graph. One of the novel decomposition graph node kernel named Conjunctive and disjunctive node kernel (CDNK) which is proposed in [6]. CDNK is an extension of NSPDK [7], which is an instance of convolution kernel (decomposition kernel). Considering a couple of nodes u and v, this kernel defines the similarity between them by taking into acount the common pairwise neighborhood subgraphs rooted at u and v.

2.2 Link Enrichment

Link enrichment is a task that intends to add the most likely non-observed links into graphs. This tasked can be performed by first using link prediction method to make a ranking over all non-observed links based on their probabilities to be actual links, and then the top non-observed links are added into the graph. A considerable number of link prediction methods which have been proposed. These methods can be classified into different categories as presented in [9].

- *Similarity-based algorithms:* This is the simplist framework for link prediction. In these methods, each non-observed link of a graph is assigned a score. This score is then directly used as the proximity between starting and ending nodes of that link. In order to define similarity, one can use nodes' attributes. Unfortunately, nodes' attributes are normally hidden. Therefore, most methods are based on the structured similarity. Graph node kernels are also in this group.
- *Maximum likelihood methods:* These algorithms presuppose some organizing principles of the network structure, with the detailed rules and specific parameters obtained by maximizing the likelihood of the observed structure. Then, the likelihood of any non-observed link can be calculated according to those rules and parameters.
- *Probabilistic models:* Probabilistic models aim at abstracting the underlying structure from the observed network, and then predicting the missing links by using the learned model. Given a target graph, G, the probabilistic model will optimize a built target function to establish a model composed of a group of parameters θ, which can best fit the observed data of the target network. Then the probability of the existence of a nonexistent link (i, j) is estimated by the conditional probability $P(A_{ij} = 1|\theta)$.

In this paper, we intend to use the global similarity-based algorithms for two reasons. First, among similarity-based algorithms, global ones show better results in general, comparing with local and semi-local similarity-based algorithms. Second, it is more straightforward to use comparing with maximum likelihood methods probabilistic models.

3 Method

Information encoded in data are usually incomplete. This usually leads to the sparsity issue when using graphs to represent for data. As a consequence, the graph-based systems which use diffusion-based kernels show limitted performances. Therefore, in this section, we describe our proposed method using link enrichment to strengthen diffusion-based graph node kernels so that the performance of graph-based systems can be improved.

Given a sparse graph $G = (V, E)$ in which $|V| = n$ and $|E| = m$, the proposed method consists of two phases:

- Link enrichment: in the first phase, starting from the graph G, we utilize a link prediction method to compute scores for all $\frac{n(n-1)}{2} - m$ candidate links. These scores represent for their probabilities to be considered as a link in the graph. The candidate links are then sorted based on their corresponding scores. The top t links in the sorted link list are added into G to have new graph $G^{'}$.

– Kernel computation: in the second phase, we apply the chosen diffusion-based graph node kernel to the achieved graph $G^{'}$ to compute kernel matrix which encodes the similarities between any couple of nodes. This kernel matrix then can be fit into graph kernel-based learning systems to make inference.

4 Evaluation

4.1 Datasets

The proposed method aims to strengthen the power of diffusion-based kernel when dealing with sparse graphs. Therefore, we employ four genetic datasets which lead to sparse graphs for the evaluation. Hereafter, we briefly describe each these datasets.

BioGPS: A gene co-expression network is constructed from BioGPS dataset, which contains 79 tissues, measured with the Affymetrix U133A array. Edges are inserted when the pairwise Pearson correlation coefficient (PCC) between genes is larger than 0.5.

HPRD: a database of curated proteomic information pertaining to human proteins. It is derived from [14] with 9465 vertices and 37039 edges. We enploy the HPRD version used in [13] in which they remove some vertices to have 7311 vertices at the end. Both BioGPS, HPRD are used in [3].

Phenotype similarity: in order to capture the relatedness of genes from a phenotypic point of view, we resort to OMIM [11] data and the phenotype similarity conceived by Van Driel et al. [14]. They define a similarity among OMIM phenotypes based on the relevance and the frequency of the Medical Subject Headings (MeSH) vocabulary terms in the corresponding OMIM text documents. We convert this information into a graph by linking those genes whose associated phenotypes have a maximal phenotypic similarity greater than a fixed cut-off value. The weight of the link is the maximal similarity among the phenotypes relative to the two genes considered. We set the similarity cut-off by following [14], where it is found that biologically meaningful relationships are those between phenotypes with a similarity score greater than 0.3.

Biogridphys: This dataset represents the physical interactions among proteins. The idea is that mutations can affect physical interactions by changing proteins shape and their effect can propagate through protein networks. We form the link between two genes if their products interact.

4.2 Evaluation Methods

To evaluate the performance of kernels we analyze the *gene prioritization*, i.e. given a set of genes known to be associated to a given disease, gene prioritization is the task to rank the candidate genes based on their probabilities to be related to that disease. Similar to the evaluation process used in [3], we choose 14 diseases with at least 30 confirmed genes. For each disease, we construct a positive set $\mathcal{P}$

with all confirmed disease genes, and five negative sets such that each negative set $\mathcal{N}$ contains random genes associated at least to one disease class which is not related to the class that is defining the positive set. In [3] the ratio between the dataset sizes is chosen as $|\mathcal{N}| = \frac{1}{2}|\mathcal{P}|$. Then, we assess the performance of kernels through a paradigm similar to 3-fold CV: each (P + U) set is partitioned into three folds, where one fold are used to train the model (SVM) and the two folds are used to test. We compute a decision score q_i for the test gene g_i as the top percentage value of score s_i among all candidate gene scores. We collect all decision scores for every gene test set to compute AUC-ROC. The final performance for a kernel a disease class is obtained by averaging over 3×5 trials.

4.3 Model Selection

Model Selection The hyper parameters of the various methods are set using a k-fold on a dataset set that is then never used in the predictive performance estimation. We try the values for LEDK and MEDK in $\{0.01, 0.05, 0.1\}$, time steps in MDK in $\{3, 5, 10\}$ and RLK parameter in $\{0.01, 0.1, 1\}$. For CDNK, we try for the degree threshold value in $\{10,\ 15,\ 20\}$, clique size threshold in $\{4,\ 5\}$, maximum radius in $\{1,\ 2\}$, maximum distance in $\{2,\ 3,\ 4\}$. Number of added links are set in $\{40\%,\ 50\%,\ 60\%,\ 70\%\}$ total number of existing links. Finally, the C of SVM is searched in $\{10^{-4},\ 10^{-3},\ 10^{-2},\ 10^{-1},\ 10^{0}, 1,\ 10,\ 10^{2},\ 10^{3},\ 10^{4}\}$.

5 Results and Discussion

Table 1 and 2 show the empirical evaluation results of each kernel with and without link enrichment using 14 genetic diseases on four different datasets. It can be seen from the tables that in all cases, using link enrichment considerably improves the power of kernels. Particularly, on BioGPS, Biogridphys and HPRD, the average AUC over all diseases of all kernels using link enrichment are about 7% higher than themself without using link enrichment. And on the Omim dataset, the difference are still noticeable with around 3%. Importantly, we observe from the results that the performance of single kernel with all employed link enrichment methods are close and are all remarkably higher than its without using link enrichment. It illustrates that the method is stable for the use of link enrichment methods and is considerable when constructing graph-based learning systems using diffusion-based kernels.

6 Conclusion

In this paper, we have proposed a novel method to boost the power of diffusion-based graph node kernel by using link enrichment paradigm. The results achieved from empirical experiments illustrate that our proposed method is noticeable when using diffusion-based graph node kernels to build learning systems. For the coming work, we desire to apply this boosting method to improve the performance of systems using kernel integration.

Table 1. *Predictive performance on 14 gene-disease associations using network induced by the BIOGPS and BIOGRIDphys. We report the AUC-ROC (%) each kernel with (B) and without (A) using link enrichment.*

	BIOGPS								**BIOGRIDphys**							
	LEDK		**MEDK**		**MDK**		**RLK**		**LEDK**		**MEDK**		**MDK**		**RLK**	
D	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A	B
0	61.8	63.7	57.9	62.8	60.4	65.2	61.1	63.0	76.4	79.6	73.2	79.7	66.4	79.2	76.6	80.2
1	55.6	70.4	52.0	62.8	53.0	67.5	54.4	63.7	60.5	67.3	52.3	63.3	54.4	65.9	59.3	65.9
2	50.3	63.6	49.5	57.0	50.4	62.7	50.4	57.4	62.2	77.2	49.1	65.2	60.0	74.0	64.4	76.3
3	61.3	73.3	62.7	72.7	60.2	74.0	61.9	72.0	69.2	79.1	63.5	77.5	60.2	71.8	70.1	79.7
4	55.7	63.2	55.1	61.7	54.8	60.8	54.8	61.2	59.2	67.5	50.2	59.5	48.6	62.7	58.7	64.6
5	60.1	70.4	59.7	66.1	62.0	70.6	61.3	70.0	63.1	70.1	54.4	62.9	61.7	68.8	63.2	70.1
6	68.8	73.6	67.7	73.5	66.0	73.4	69.8	73.7	60.1	72.2	53.2	64.1	56.4	68.7	61.3	71.1
7	71.1	76.5	65.6	71.7	71.4	76.9	68.7	75.0	70.6	75.7	62.9	73.6	66.8	74.6	72.3	75.4
8	62.4	64.4	63.7	66.9	59.3	64.2	62.4	64.4	69.1	72.3	61.3	69.2	74.0	79.1	70.4	72.6
9	69.4	74.5	62.5	73.6	69.6	75.9	68.4	72.8	61.3	70.4	57.3	67.4	54.4	68.2	62.3	70.8
10	60.0	64.6	58.4	61.4	56.0	61.4	60.6	63.5	58.9	64.1	56.2	61.5	58.4	63.9	59.1	63.2
11	64.4	76.1	61.9	71.2	65.4	74.2	64.5	76.2	61.0	69.5	59.6	68.6	55.9	66.7	60.8	68.5
12	57.2	65.8	55.1	63.7	57.3	67.2	56.4	63.3	55.4	69.1	57.6	71.5	54.8	69.4	55.2	68.1
13	55.4	61.6	55.0	63.7	55.5	63.1	54.9	62.0	57.4	72.4	53.2	64.7	55.1	67.6	56.4	68.7
$\overline{AUC}$	61.0	68.7	59.1	66.3	60.1	68.4	60.7	67.0	63.2	71.9	57.4	67.8	59.1	70.0	63.6	71.1

Table 2. *Predictive performance on 14 gene-disease associations using network induced by the HPRD and OMIM. We report the AUC-ROC (%) each kernel with (B) and without (A) using link enrichment.*

	HPRD								**OMIM**							
	LEDK		**MEDK**		**MDK**		**RLK**		**LEDK**		**MEDK**		**MDK**		**RLK**	
D	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A	B
1	75.5	77.9	75.1	77.2	75.6	78.6	75.6	78.8	84.9	87.4	84.4	85.9	87.3	89.5	84.7	86.5
2	56.6	63.5	55.9	59.5	58.0	63.0	58.0	63.5	76.6	78.8	75.7	77.3	71.2	72.8	76.4	78.2
3	61.2	76.1	56.1	70.0	65.5	76.6	64.4	74.6	78.4	83.8	74.4	82.5	78.9	84.0	77.5	83.2
4	67.3	74.1	65.5	72.7	68.1	74.3	68.3	74.0	91.3	93.0	89.3	92.6	88.8	90.0	91.4	92.8
5	57.6	68.4	55.1	67.3	59.0	68.3	59.0	68.2	76.1	84.1	75.6	84.3	77.7	83.1	76.3	83.8
6	67.1	73.5	64.8	71.8	68.3	73.8	67.1	73.6	81.9	83.9	79.1	83.6	76.3	83.7	82.2	83.8
7	68.8	73.7	65.5	71.9	71.1	74.2	70.1	73.5	81.2	84.1	79.5	82.2	82.7	85.8	80.6	84.3
8	76.2	83.3	73.4	80.7	77.6	83.3	79.3	83.2	84.3	91.0	83.9	89.7	89.3	91.7	84.3	90.5
9	68.2	74.5	64.4	72.2	71.0	77.5	69.8	75.3	78.8	80.8	78.2	80.1	78.5	81.5	78.6	80.8
10	66.0	76.6	60.4	74.9	67.1	77.5	69.6	77.1	86.3	87.6	86.7	88.1	85.1	88.3	86.0	87.6
11	60.5	65.0	58.3	64.1	60.3	65.6	61.3	65.1	83.3	84.6	79.8	83.1	82.0	82.8	83.0	84.4
12	61.9	72.1	60.5	68.8	59.2	73.0	61.7	72.0	82.0	87.0	79.8	87.2	84.7	87.7	81.3	86.4
13	67.6	72.6	65.1	71.5	65.3	73.3	67.7	73.2	85.0	88.6	84.1	89.4	78.2	83.8	85.0	88.5
14	68.4	74.1	64.6	70.5	63.4	73.7	68.7	74.5	97.4	99.2	97.2	99.1	97.5	98.6	97.4	99.2
$\overline{AUC}$	65.9	73.2	63.2	70.9	66.4	73.8	67.2	73.3	83.4	86.7	82.0	86.1	82.7	86.0	83.2	86.4

References

1. Huang, Z., et al.: A graph-based recommender system for digital library. Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries. ACM, 2002.
2. Kondor, R. I., and Lafferty J.: Diffusion kernels on graphs and o ther discrete structures." Machine Learning, Proceedings of the 19th International Conference (ICML 2002). 2002.
3. Chen, B., et al.: Disease gene identification by using graph kernels and Markov random fields. Science China. Life Sciences 57.11 (2014): 1054.
4. Chebotarev P. and Shamis E.: The matrix-forest theorem and measuring relations in small social groups. Automation and Remote Control 1997, 58(9):15051514.
5. Haussler, D.: Convolution kernels on discrete structures. Technical Report UCS-CRL-99-10, UC Santa Cruz, 1999.
6. Dinh, T., Alessandro, S., and Fabrizio, C.: Conjunctive disjunctive node kernel. Proceedings of 25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2017.
7. Costa, F., and Kurt D.: Fast neighborhood subgraph pairwise distance kernel. Proceedings of the 26th International Conference on Machine Learning. Omnipress, 2010.
8. Ramadan, E., Sadiq A., and Rafiul H.: "Network topology measures for identifying disease-gene association in breast cancer." BMC bioinformatics 17.7 (2016): 274.
9. Lu, L., and Tao Z.: Link prediction in complex networks: A survey. Physica A: Statistical Mechanics and its Applications 390.6 (2011): 1150-1170.
10. Fouss, F., et al.: An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification. Neural networks 31 (2012): 53-72.
11. McKusick, Victor A.: Mendelian Inheritance in Man and its online version, OMIM. The American Journal of Human Genetics 80.4 (2007): 588-604.
12. Chatr-Aryamontri, Andrew, et al.: The BioGRID interaction database: 2015 update. Nucleic acids research 43.D1 (2015): D470-D478.
13. Prasad, TSK et al.: Human Protein Reference Database - 2009 Update. Nucleic Acids Res 2009, 37(Database):D767-72.
14. Van Driel, Marc A., et al.: A text-mining analysis of the human phenome." European journal of human genetics 14.5 (2006): 535-542.