Link Enrichment for Strengthening Diffusion-based Graph Node Kernels

Dinh Tran-Van, Alessandro Sperduti, and Fabrizio Costa

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

Abstract. Node similarity 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 node similarity. However, they only return state-of-the-art results in the case of dense graphs. In this paper, we propose a method employing link enrichment that aims to strengthen diffusion-based kernels when working with sparse graphs. The empirical assessment shows that our method considerably improves the power of diffusion-based graph node kernels in the case of sparse graphs.

Keywords: Graph node kernels, diffusion-based kernels, strenghtening 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 this huge amount of data, a number of learning systems have been introduced. Some of these systems are tailored for specific types of data. Graph is a widely used data representation and it is employed by many systems in different domains [1], [8]. Learning systems that take graphs as their input are referred to as graph-based learning systems.

In graph-based systems, the measurement of the proximity between nodes of a graph is one of the key factors that determines the performance of the system. The most common paradigm used to capture similarity between nodes is to resort to graph node kernels. In fact, many graph node kernels have been proposed and applied in several real-world applications and domains. Among them, diffusion-based kernels [2] are the most commonly employed[1], very often returning state-of-the-art results. However, these node kernels usually show good performance only when dealing with dense graphs, i.e., graphs with a high value of average node degree. Vice versa in the case of sparse graphs, i.e. graphs with a low value of average node degree, they usually lead to poor performance.

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

This is due to: *i)* the number of links in the graph is very limited compared to the situation encountered in a complete graph, so the information cannot be spreaded properly through the whole graph; *ii)* the lack of links also causes the fragmentation of the graph into isolated components. It is important to stress that, in case of fragmentation, information cannot be diffused between isolated components. Therefore, the similarity between nodes located in different isolated components, as measured by diffusion-based graph node kernels, is equal to zero. As a consequence, the performances of these type of kernels hinders the possibility 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 of a graph. Many link prediction methods have been proposed. In [9], a quite exhaustive study of link prediction methods is presented in which methods proposed in literature are classified into different groups. The most widely used framework is the similarity-based one because of its effectiveness and ease of use. In this group of methods, to each pair of nodes is assigned a score which is directly used 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 goes along this direction. The experimental assessment on different real-world datasets confirms the efficacy of our proposed method.

2 Notation and Background

Let us consider an undirected graph $G = (V, E)$ in which V represents 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. $D_{ii} = \sum_j A_{ij}$.

Graph Node Kernels. 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. Most graph node kernels belong to one of two popular frameworks: diffusion-based graph node kernels and decomposition graph node kernels.

Diffusion-based kernels can be considered as modifications of the 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 state-of-the-art performance when dealing with dense graphs because of their ability to capture a global similarity measure. However, they perform poorly

when facing sparse graphs with a low number of links and a high number of disconnected components. In the 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. The formula to compute the LEDK kernel matrix is:

$$K_{LEDK} = e^{-\beta L} \,, \tag{1}$$

  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 [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]. This kernel is dubbed Markov exponential diffusion kernel (MEDK) and its kernel matrix is obtained by the following formula:

$$K_{MEDK} = e^{-\beta M} \,. \tag{2}$$

  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 and I is the identity matrix. 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 between 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. The corresponding kernel matrix is computed starting from the transition matrix P ($P = D^{-1}A$) and by defining $Z(t) = \frac{1}{t} \sum_{\tau=1}^{t} P^{\tau}$, as follows:

$$K_{MDK} = Z(t)Z^{\top}(t) \,. \tag{3}$$

– *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. Its kernel matrix is defined as follows:

$$K_{RLK} = \sum_{n=0}^{\infty} \beta^n (-L)^n \ , \tag{4}$$

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 the 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, the graph matching problem, or equivalently the graph isomorphic problem, need to be solved, which is not known to be solvable in polynomial time nor to belong to the NP-complete complexity class. An advantage of using decomposition kernels is the possibility to have non-zero similarity values for node couples located in distinct disconnected components of a graph. A recent and effective decomposition graph node kernel is the Conjunctive and Disjunctive Node Kernel (CDNK), 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, the CDNK kernel defines the similarity between them by taking into account the common pairwise neighborhood subgraphs rooted at u and v.

Link Enrichment. Link enrichment is a task that intends to add the most likely non-observed links into a graph. This task can be performed by first using a 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 have been proposed in the literature. These methods can be classified into different categories as discussed in [9]: *similarity-based algorithms*, *maximum likelihood methods*, and *probabilistic models*. Similarity-based methods assign for each non-observed link a score and this score is then directly used as the proximity between starting and ending nodes of that link. In maximum likelihood methods, some organizing principles of the network structure are assumed. Then, the likelihood of any non-observed link can be calculated according to corresponding rules and parameters. Probabilistic models aim at abstracting the underlying structure from the observed network, and to predict the missing links by using a 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 which can best fit the observed data of the target network.

In this paper, we employ five graph node kernels described in the previous section: LEDK, MEDK, MDK, RLK and CDNK for link prediction since they belong to global similarity-based group. There are two reasons for the use of global similarity-based methods. Firstly, among similarity-based algorithms, global ones show, in general, better results than local and semi-local similarity-based algorithms. Secondly, similarity-based algorithms are much simpler to deal with (and computationally less demanding) than maximum likelihood methods and probabilistic models.

3 Method

Information encoded in data is usually incomplete. This usually leads to the sparsity issue when using graphs to represent data. As a consequence, graph-based systems which use diffusion-based kernels show limited performances. Therefore, in this section, we describe our proposed method, based on 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 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 obtain a new enriched graph $G^{'}$.
- Kernel computation: in the second phase, we apply the chosen diffusion-based graph node kernel to the achieved graph $G^{'}$ to compute the kernel matrix which encodes the similarities between any couple of nodes. This kernel matrix can then be used into graph kernel-based learning systems to make inference.

4 Evaluation

We have assessed our approach on four different datasets. In the following, we first describe the datasets and then we present the obtained results.

4.1 Datasets

The proposed method aims to strengthen the power of diffusion-based kernels when dealing with sparse graphs. Therefore, we employ genetic-related data which typically lead to sparse graphs. Hereafter, we briefly describe them.

BioGPS: a gene co-expression network (7311 nodes and 911294 edges) is constructed from the 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 9,465 vertices and 37,039 edges. We employ the HPRD version used in [13] in which they remove some vertices so to have 7311 nodes and 30503 edges remaining. HPRD, and BioGPS, 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 converted 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 considered genes. We set the similarity cut-off by following [14] with a similarity score greater than 0.3. Finally, we obtain a network with 3393 nodes and 144739 edges.

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 introduce a link between two genes if their products interact. As a result, the achieved network consists of 15389 nodes and 155333 edges.

4.2 Evaluation Methods

To evaluate the performance of the considered kernels, we use the *gene prioritization* task, i.e. given a set of genes known to be associated to a given disease, gene prioritization consists in ranking 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 involved genes. For each disease, we first construct a positive set $\mathcal{P}$ with all confirmed disease genes, and a negative set $\mathcal{N}$ contains random genes associated at least to one disease class, but not related to the class which defines the positive set such that $|\mathcal{N}| = \frac{1}{2}|\mathcal{P}|$. We then repeat this procedure five times. Each time, we keep the positive set and only change the negative set. As a result, we have five different training sets for each disease class. We assess the performance of kernels through a paradigm similar to 3-fold CV: each $(\mathcal{P} + \mathcal{U})$ set is partitioned into three folds, where one fold is used to train the model (via a linear SVM) and the two folds are used to test. For each test gene g_i, model returns a score s_i showing its likelihood to be associated to the disease. Next a dicision score q_i is computed as the top percentage value of s_i among all candidate gene scores. We collect all decision scores for every test genes to compute AUC-ROC. The final performance on the disease class is obtained by taking average over 3×5 trials.

Model Selection: The hyper parameters of the various methods are set using a 3-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%} over total number of existing links. Finally, the C of SVM is searched in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4\}$.

5 Results and Discussion

Table 1. *Predictive performance on 14 gene-disease associations using four different networks induced by the BioGPS, Biogridphys, Hprd and Omim. We report the average AUC-ROC (%) and standard deviations for all difussion-based kernels with (B) and without (A) using link enrichment.*

	BioGPS		**Biogridphys**		**Hprd**		**Omim**	
Disease	A	B	A	B	A	B	A	B
1	60.3±1.5	63.4±1.0	73.1±4.1	77.1±2.9	75.5±0.2	77.5±0.9	85.3±1.1	86.9±1.5
2	53.7±1.4	63.4±3.8	56.6±3.4	61.3±4.1	57.1±0.9	60.2±1.8	75.0±2.2	76.5±2.4
3	50.2±0.4	58.6±3.0	58.9±5.9	67.5±7.7	61.8±3.6	70.7±3.8	77.3±1.8	83.1±0.9
4	61.5±0.9	72.2±2.2	65.7±4.1	74.6±4.2	67.3±1.1	71.9±2.2	90.2±1.2	92.1±1.2
5	55.1±0.4	61.7±0.9	54.2±4.8	60.7±4.0	57.7±1.6	67.0±1.8	76.4±0.8	81.9±1.5
6	60.8±0.9	67.9±2.2	60.6±3.6	65.9±3.5	66.8±1.3	71.9±2.3	79.9±2.4	83.3±1.2
7	68.1±1.4	73.4±0.7	57.7±3.2	63.7±4.0	68.9±2.1	72.5±1.2	81.0±1.2	84.1±1.0
8	69.2±2.3	74.0±2.2	68.1±3.6	72.6±2.5	76.6±2.2	80.3±2.8	85.4±2.2	91.0±1.0
9	62.0±1.6	64.5±1.4	68.7±4.6	71.7±4.3	68.4±2.5	75.0±3.2	78.5±0.2	80.6±0.6
10	67.5±2.9	72.9±1.8	58.8±3.2	66.1±3.8	65.8±3.4	74.4±2.6	86.1±0.6	87.8±0.3
11	58.7±1.8	62.3±1.5	58.2±1.2	61.6±1.7	60.1±1.1	64.2±1.5	82.0±1.4	83.6±0.9
12	64.0±1.3	73.6±1.7	59.3±2.1	67.0±2.8	60.8±1.1	68.8±2.8	82.0±1.8	85.9±1.7
13	56.5±0.9	63.3±2.4	55.8±1.1	65.1±4.2	66.4±1.3	71.8±1.7	83.1±2.8	87.5±2.5
14	55.2±0.3	62.3±1.2	55.6±1.6	63.5±4.0	66.3±2.3	71.1±2.8	97.4±0.1	99.0±0.4
$\overline{AUC}$	60.2±0.3	66.7±1.2	60.8±1.6	67.0±4.0	65.7±2.3	71.2±2.8	82.8±0.1	86.0±0.4

Table 1 shows the average predictive performances of diease gene prioritization system on 14 genetic diseases and four genetic networks. In each experiment, we test performance of system with the use kernels with and without using link enrichment. Overall, the performance of system with link enrichment are remarkably higher than not using link enrichment in every disease and dataset. In particular, the use of link enrichment helps to improve in average around 6% on BioGPS, Biogridphys and Hprd, and 3% in Omim. The detail of all experimental results can be found in the *appendix* [2]. Considering a specific kernel,

[2] https://github.com/

the performance of the system get higher with the use of any link enrichment methods comparing to it 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.

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. 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.
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.