Link Enrichment to Strengthen Diffusion-based Graph Node Kernels

Anonymous Authors[1]

Abstract

The 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 by employing link enrichment. The empirical experiments show that our method considerably improves the power of diffusion-based graph node kernels.

1. Introduction

Recently, with the fast development of science and technology, we have witnessed the rapid growth of data in terms of 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 to representate data. Graph-based learning systems are learning systems which consider graphs as their input and they are widely used in different domains (**?**).

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 using 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 application, domains. Among them, diffusion-based kernel [1] are the most commonly employed and show promissing results. However, those node kernels usually show good performance when dealing with dense graphs - graphs with high value of an average high node degree node. And, vice versa, they usually lead to poor performance when working with sparse graphs.

Data collected from domains are normally in complete. This raises challenge for graph-based learning systems. One of the reasons to explain for that is the lack of links connecting nodes. To overcome this problem, a potential solution is to use link enrichment - link enrichment is a task that aims at recovering the missing links (or future links in evolution networks). There exist many link enrichment methods and they can be classified into different groups (**?**). The simplist and most widely used framework is the similarity-based algorithms where 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. The experimental results confirm 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, weighted graph $G = (V, E)$ representing entities and relationships among them. The adjacency matrix $\mathbf{A}$ is a symmetric matrix used to characterize the direct links between vertices v_i and v_j in the graph. Any entry A_{ij} is equal to w_{ij} when there exists an edge of weight $w_{ij} > 0$ connecting v_i and v_j, and is 0 otherwise. The Laplacian matrix $\mathbf{L}$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ 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 following graph node kernels are described under this notation convention.

[*]Equal contribution [1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

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

2.1. Diffusion-based Graph Node Kernels

Laplacian exponential diffusion kernel
One of the most well-known kernels for graphs is the Laplacian exponential diffusion kernel $\mathbf{K}_{LED}$, 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 $\mathbf{K}_{LED}$ in (**?**): 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, $\mathbf{K}_{LED}$ can capture the long range relationship between vertices of a graph to define the global similarities. Below is the formula to compute $\mathbf{K}_{LED}$ values:

$$\mathbf{K}_{LED} = e^{-\beta \mathbf{L}} = \mathbf{I} - \beta \mathbf{L} + \frac{\beta \mathbf{L}^2}{2!} - ... \tag{1}$$

where β is the diffusion parameter and is used to control the rate of diffusion and $\mathbf{I}$ is the identity matrix. 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. $\mathbf{K}_{LED}$ is positive semi-definite as proved in (**?**).

Markov exponential diffusion kernel
In $\mathbf{K}_{LED}$, 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 $\mathbf{K}_{LED}$ was introduced by Chen et al in (**?**), called Markov exponential diffusion kernel $\mathbf{K}_{MED}$ and given by the following formula:

$$\mathbf{K}_{MED} = e^{-\beta \mathbf{M}} \tag{2}$$

The difference with respect to the Laplacian diffusion kernel is the replacement of $\mathbf{L}$ by the matrix $\mathbf{M} = (\mathbf{D} - \mathbf{A} - n\mathbf{I})/n$ where n is the total number of vertices in graph. The role of β is the same as for $\mathbf{K}_{LED}$.

Markov diffusion kernel
The original Markov diffusion kernel $\mathbf{K}_{MD}$ was introduced by Fouss et al. (**?**) and 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 $\mathbf{P}$ and by defining $\mathbf{Z}(t) = \frac{1}{t}\sum_{\tau=1}^{t} \mathbf{P}^{\tau}$, as follows:

$$\mathbf{K}_{MD} = \mathbf{Z}(t)\mathbf{Z}^{\top}(t) \tag{3}$$

Regularized Laplacian kernel
Another popular graph node kernel function used in graph mining is the regularized Laplacian kernel $\mathbf{K}_{RL}$. This kernel function was introduced by Chebotarev and Shamis in (**?**) and represents a normalized version of the random walk with restart model. It is defined as follows:

$$\mathbf{K}_{RL} = \sum_{n=0}^{\infty} \beta^n (-\mathbf{L})^n = (\mathbf{I} + \beta \mathbf{L})^{-1} \tag{4}$$

where the parameter β is again the diffusion parameter. $\mathbf{K}_{RL}$ counts the paths connecting two nodes on the graph induced by taking $-\mathbf{L}$ 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. $\mathbf{K}_{RL}$ remains a relatedness measure even when diffusion factor is large, by virtue of the negative weights assigned to self-loops.

2.2. Link Enrichment

Graphs are widely used to represent data in different domains. Since information encoded in data normally are not complete, it impacts on the performance of graph-based learning systems. Link enrichment (link prediction) is a task that intends to add the most likely non-observed links into graphs. There is a number of link prediction methods which have been proposed. These methods can be classified into different categories as presented in (**?**). The simpliest framwork for link prediction is Similarity-Based Algorithms where each pair of nodes is assigned a score which is directly defined as the similarity. In this paper, we employ global similarity methods which belong to Similarity-Based Algorithms to predict the most likely of non-observed links to add into graphs.

3. Method

In this section, we present the link enrichment method for strengthening diffusion-based graph node kernels.

Given a graph $G = (V, E)$, our method consists of two pharses:

- Link enrichment: in the first phase, Starting from the graph G, we apply one of the link prediction method on G to compute a score for each of non-observed link. This score represents for its probability to be considered as a link in the graph. The non-observed link list is then sorted by their corresponding scores. The top links in the sorted list are added into G to have new graph $G^{'}$.

- Kernel computation: in the second phase, we apply diffusion-based graph node kernels to the graph $G^{'}$ to compute kernel matrix which encodes the similarities between any couple of nodes. This

4. Evaluation

We perform an empirical evaluation of the predictive performance of several kernel based methods on two of the databases used in (**?**).

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 (**?**) with 9465 vertices and 37039 edges. We enploy the HPRD version used in (**?**) in which they remove some vertices to have 7311 vertices at the end.

To evaluate the performance of graph node 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 (**?**), we choose 12 diseases with at least 30 confirmed genes. For each disease, we construct a positive set $\mathcal{P}$ with all confirmed disease genes, and a negative set $\mathcal{N}$ which contains random genes associated at least to one disease class which is not related to the class that is defining the positive set. In (**?**) the ratio between the dataset sizes is chosen as $|\mathcal{N}| = \frac{1}{2}|\mathcal{P}|$. The predictive performance of each method is evaluated via a leave-one-out cross validation: one gene is kept out in turn and the rest are used to train an SVM model. 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 in the training set to form a global decision score list on which we compute the AUC ROC.

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 estimate. We try the values for diffusion parameter in DK and MED in $\{10^{-3}, 10^{-3}, 10^{-2}, 10^{-1}\}$, time steps in MD in $\{1, 10, 100\}$ and RL parameter in $\{1, 4, 7\}$. For CDNK, we try for the degree threshold values in $\{10,\ 15,\ 20\}$, clique size threshold in $\{4,\ 5\}$, maximum radius in $\{1,\ 2\}$, maximum distance in $\{2,\ 3,\ 4\}$. Finally, the C of SVM is searched in $\{10^{-5},\ 10^{-4},\ 10^{-3},\ 10^{-2},\ 10^{-1}, 1,\ 10,\ 10^{2}\}$.

5. Results and Discussion

6. Conclusion

HPRD Dataset								
	LEDK		MEDK		MDK		RLK	
Disease	No_Link	Link	No_Link	Link	No_Link	Link	No_Link	Link
0	73/2	74/1	71/1	71/2	71/1	69/2	73/2	73/1
1	68/2	69/1	54/2	55/1	82/2	82/1	80/2	80/1
2	77/1	75/2	72/1	70/2	79/1	77/2	81/1	76/2
3	60/2	62/1	62/2	63/1	63/2	66/1	65/2	67/1
4	67/2	68/1	65/2	67/1	68/2	70/1	68/1	68/2
5	67/1	67/2	69/1	68/2	65/2	68/1	66/2	70/1
6	87/2	88/1	87/2	87/1	87/2	89/1	87/2	88/1
7	79/2	80/1	78/2	79/1	81/1	78/2	79/1	77/2
8	78/2	79/1	72/2	72/1	79/1	75/2	80/2	80/1
9	73/2	76/1	71/2	73/1	70/2	72/1	70/2	77/1
10	80/1	79/2	82/1	81/2	77/1	75/2	80/2	80/1
11	76/1	75/2	75/2	75/1	83/2	87/1	84/1	83/2
$\overline{AUC}$	74/1.67	74/1.33	71/1.67	72/1.33	75/1.58	76/1.42	76/1.67	77/1.33
		0.76		0.73		0.76		0.77

Table 1. Predictive performance on 12 gene-disease associations using network induced by the HPRD. We report the AUC-ROC (%) and the rank for each kernel method.

HPRD Dataset								
	LEDK		MEDK		MDK		RLK	
Disease	No_Link	Link	No_Link	Link	No_Link	Link	No_Link	Link
0	73/2	76/1	76/2	79/1	70/1	69/2	70/1	67/2
1	60/1	61/2	59/1	58/2	60/2	61/1	60/2	72/1
2	85/1	84/2	85/1	84/2	82/1	82/2	82/2	84/1
3	66/2	67/1	56/2	60/1	66/2	69/1	66/2	67/1
4	61/2	62/1	63/1	62/2	58/2	58/1	58/2	58/1
5	70/1	69/2	70/1	69/2	67/2	68/1	67/2	68/1
6	73/1	71/2	68/1	68/2	69/1	69/2	69/2	76/1
7	69/2	69/1	69/2	69/1	67/2	67/1	67/1	65/2
8	73/2	74/1	70/2	71/1	65/2	68/1	65/2	67/1
9	69/1	67/2	65/2	66/1	64/2	65/1	64/1	64/2
10	58/2	60/1	52/2	56/1	60/2	61/1	60/2	70/1
11	69/2	73/1	70/2	70/1	60/2	61/1	60/2	62/1
$\frac{\overline{AUC}}{\overline{Rank}}$	69/1.58	69/1.42	67/1.58	68/1.42	66/1.75	67/1.25	66/1.75	68/1.25
		72		69		68		71

Table 2. Predictive performance on 12 gene-disease associations using network induced by the BioGPS. We report the AUC-ROC (%) and the rank for each kernel method.