AOEGK - A New Efficient Graph Node Kernel

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

1- Department of Mathematics, Padova University
Trieste, 63, 35121 Padova, Italy

2- Bioinformatics Group, Department of Computer Science, Freiburg University
Georges-Kohler-Allee 106, 79110 Freiburg, Germany

Abstract. Type your 100 words abstract here. Please do not modify the style of the paper. In particular, keep the text offsets to zero when possible (see above in this 'ESANNV2.tex' sample file). You may *slightly* modify it when necessary, but strictly respecting the margin requirements is mandatory (see the instructions to authors for more details).

1 Introduction

In genetic predictive systems, the definition of gene-gene similarity is one of the key aspects that determines the performance of the systems. A popular approach is to use a domain specific notion of relatedness between genes and materialize a graph where nodes are genes and edges are relationships that satisfy a stringent criterion (e.g. if the products of the two genes interact in a known pathway). Once the graph is materialized, one can define the node similarities by exploiting the graph properties.

Concerning graph node similarity measurement, graph node kernels (kernels) are known as the most effective paradigm to use. In the last decades, a number of kernels have been proposed. Most kernels are based on the natural transitive property (i.e. connecting two nodes via a path). For instance, diffusion kernel takes into account all the paths connecting two nodes to compute the similarity. On the positive side, these approaches can be employed on networks that: (1) are dense, (2) have nodes with high maximal degrees, (3) have cliques. On the negative side, these approaches do not have a high discriminative capacity. There are two reasons for this: (1) they are based on considering in an additive and independent fashion the contribution of each information source and, (2) they employ a representation that is aware of the distance between nodes but cannot model the precise configuration of the context (i.e. the neighboring sub-graph).

In order to increase the modelling discriminative capacity we propose to employ decomposition graph kernels (DGK). On one side, these approaches are computationally efficient (quasi linear in the network size) and can adequately represent contextual information. On the other side, they can be applied on networks that are sparse and have a small maximal node degree. In this paper, we employ the idea mentioned in [2] that originally proposes for graph similarities to pose a new graph node kernel named AOLGK that can fully exploit the graph structure to measure the node similarity in both sparse and dense graphs.

This paper is organized as follows: first we describe some of the most famous graph node kernel in Section 2. Then the proposed kernel is introduced in the

Section 3. In Section 4, we evaluate and compare the performance of our kernel with some others. The obtained results are showed and discussed in Section 5. Last, we make the conclusion and plan for the future work in Section 6.

2 Related Work

Graph node proximity is a topic that has been attracted many research. As a consequence, a considered number of graph node kernels have been proposed and applied in a wide range of applications. Most famous kernels are based on random walk paradigm, in which the similarity is a function of the paths connecting two nodes. In the following paragraphs, we briefly describe some of the most popular graph node kernels.

One of the most well-known kernels for graph nodes named Diffusion kernel (DK) is proposed in [1]. This kernel is based on the heat diffusion phenomenon. First, a given amount of heat is put on each node and diffused though the edges of the graph in an arbitrary time interval. Then the similarity between the node couple (v_i, v_j) is measured as the amount of heat starting from v_i and reaching v_j within the given time. Formally, DK is computed by using formula $K_{DK} = e^{-\beta \mathbf{L}}$, in which β is diffusion parameter that controls the rate of diffusion and L is the Laplacian matrix. On the one side, if β is too small, the local information cannot be diffused effectively. On the other side, if it is too large, the local information will be lost. Therefore, DK can capture the long range relationship between nodes of a graph to define the global similarities.

In DK, the similarity between the high degree nodes are generally higher compared to that between the 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 DK is introduced in [2] called Markov exponential diffusion kernel (MEDK). It is given by the formula $K_{MED} = e^{-\beta M}$. The difference with respect to DK is the replacement of $\mathbf{L}$ by the matrix $M = (D - A - nI)/n$ where n is the total number of vertices in graph and D is diagonal matrix.

The original Markov diffusion kernel (MDK) is introduced in [**?**]. This kernel 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 $K_{MDK} = Z(t)Z^{\top}(t)$.

Another popular graph node kernel used in graph mining is the regularized Laplacian kernel 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 $K_{RL} = \sum_{n=0}^{\infty} \beta^n (-L)^n = (I + \alpha L)^{-1}$, where $\alpha > 0$ is a parameter. K_{RL} 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_{RL} remains a relatedness measure even when diffusion factor is large, by virtue of the negative weights assigned to self-loops.

As discussed in the section 1, the above kernels face with the discrimination problem especially in the graphs that have many disconnected components. To overcome that limitation, in this study, we leverage the idea from Fast Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) [2] to propose a new efficient graph node kernel. Originally NSPDK is designed to measure the similarity between molecular graphs. First, each graph is decomposed into a finite vector of pairwise neighborhood subgraphs that are encoded as strings. The obtained finite sting vector is then transformed into integer vector by using a hash function. The similarity between any graph couple is computed by taking the dot product of two corresponding vectors. This kernel takes advantage from the idea of decomposition and the method to solve graph isomorphic problem. However, our aim is to create an efficient kernel to measure the proximity between nodes in a given graph. Here we introduce the new graph node kernel that first decomposes the input graph and then modify the vectorization method of NSPDK to form node similarity measure.

3 Method

3.1 Notation and Definitions

We intently follow the notations used in [2]. A graph *G = (V,E)* is a structure that consists of two sets: a node set *V* and an edge set *E*. The notation *V(G)* and *E(G)* are used to refer to the node set and edge set of *G*. A subgraph *G(W)* is a graph induced from G with the node set *W* and the edge set containing every edge in *G* whose endpoints are in *W*. The neighborhood set of v is denoted as *N(v)*. We notate the cardinality of a set A as $|A|$. The degree of a node v is the cardinality of its neighborhood set and is denoted as *d(v)*. A *clique* of the graph *G* is a fully connected subgraph of G and *clique(G)* indicates the list of all cliques in *G*.

3.2 Kernel Definition

Give the definition of kernel and its properties here.

3.3 Proposed Kernel

In this part, we propose the new kernel named AND-OR edge graph node kernel (AOEGK). The proposed kernel consists of two main steps: decomposition and vectorization. Following, we illustrate each step in detail.

- *Decomposition:* The objective of the decomposition process is to make the input graph sparser to use in the next step. We utilize kcore and clique techniques described in [4] and [5] respectively to decompose. Besides, we

propose to use two kinds of edges referred as AND and OR edges. For kcore decomposition, the algorithm is presented in Algorithm 1. Given a graph *G=(V,E)* and a maximal node degree, T, (i.e maximal degree of graph nodes after decomposing), we recursively apply kcore decomposition until none of the node having degree greater or equal to T.

Algorithm 1: Kcore decomposition algorithm

Data:

- $G = (V, E)$
- T: Maximum degree threshold

Result: Decomposed graph

$V_H \leftarrow V$; $G_D \leftarrow \emptyset$; $G_H \leftarrow G$;
while $|V_H > 0|$ **do**
  $V_{HT} \leftarrow \{v \in V(G_H) \mid d(v) \geq T\}$;
  $V_{LT} \leftarrow \{v \in V(G_H) \mid d(v) < T\}$;
  $G_H \leftarrow G_H(V_{HT})$;
  $G_L \leftarrow G_H(V_{LT})$;
  $G_D \leftarrow G_D \cup G_L$;
end
$E_0 \leftarrow E(G) - E(G_D)$;
Add And-edges in E_0 to G_D;
return G_D

Algorithm 2: Clique decomposition algorithm

Data:

- *G = (V,E)*
- *T: Minimum clique size*
- *New-ID: New node ID*

Result: Decomposed graph

```
LC ← {clique ∈ S(G)| |V(clique)| ≥ T};
for clique ∈ LC do
    Add New − ID node to G ;
    for n ∈ V(clique) do
        Add OR-edge (n, New − ID) to G ;
        Remove E(clique) from G ;
        for v ∈ N(n) do
            if v ∉ V(clique) then
                add AND-edge (v, n) to G
            end
        end
        New-ID = New-ID + 1
    end
end
return G_D
```

- *Vectorization:*

4 Evaluation

In this section, we aim at evaluating the performance of our proposed kernel and comparing it with some of the most popular graph node kernels.

4.1 Dataset

BioGPS: is a gene co-expression network, which includes 79 tissues in duplicates, measured with the Affymetrix U133A array. The gene co-expression between two genes represents their tendency to be expressed in the same amounts across different tissues or cell types. Its value can be defined by the Pearson correlation coefficient (PCC) of the expression profile vectors $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ and $x_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$, where x_{lk} stands for the expression of the l_{th} gene in the k_{th} tissue. The co-expression is so given by the following formula:

$$PCC(g_i, g_j) = \frac{\sum_{k=1}^{n}(x_{ik}-\bar{x}_i)(x_{jk}-\bar{x}_j)}{\sqrt{\sum_{k=1}^{n}(x_{ik}-\bar{x}_i)^2}\sqrt{\sum_{k=1}^{n}(x_{jk}-\bar{x}_j)^2}},$$

where $\bar{x}_i = \frac{1}{n}\sum_{k=1}^{n} x_{ik}$ and $\bar{x}_j = \frac{1}{n}\sum_{k=1}^{n} x_{jk}$. In BioGPS, a pair of genes is joined by an edge (g_i, g_j) of unitary weight if the corresponding correlation value

$PCC(g_i, g_j)$ is larger than 0.5.

BIOGRIDphys: This layer 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. In the adjacency matrix, the entries of coordinates (i, j) and (j, i) are equal to 1 if any product of gene i interacts with any product of gene j. Otherwise, they are equal to 0.

BIOGRIDgen: 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. In the adjacency matrix, the entries of coordinates (i, j) and (j, i) are equal to 1 if gene i and j interact. Otherwise, they are equal to 0.

4.2 Experimental Setup

4.3 Evaluation Methods

5 Results and Discussion

Disease	AOLGK	LEDK	MDK	RLK	MEDK
Cardiovascular	0.70/3	0.62/5	0.63/4	0.72/1.5	0.72/1.5
Connective	0.60/1	0.58/2	0.57/4	0.57/4	0.57/4
Dermatological	0.76/2	0.68/5	0.79/1	0.72/3	0.71/4
Developmental	0.89/1	0.88/2	0.74/5	0.84/3.5	0.84/3.5
Endocrine	0.65/5	0.73/2	0.74/1	0.70/3	0.69/4
Hematological	0.87/1	0.84/4	0.83/5	0.85/2.5	0.85/2.5
Immunological	0.76/5	0.78/3	0.77/4	0.80/1.5	0.80/1.5
Metabolic	0.79/2	0.76/5	0.78/4	0.79/2	0.79/2
Muscular	0.69/5	0.74/4	0.78/1	0.77/2	0.76/3
Ophthalmological	0.79/1	0.75/4	0.75/4	0.77/2.5	0.77/2.5
Renal	0.76/1	0.70/5	0.75/2	0.72/4	0.73/3
Skeletal	0.71/3	0.75/2	0.77/1	0.66/4.5	0.66/4.5
Average	0.75/2.5	0.73/3.6	0.74/3.0	0.74/2.8	0.74/3.0

Table 1: Performance of kernels on different genetic diseases

6 Conclusions

References

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

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

[3] Goh KI et al, The human disease network. Proceedings of the National Academy of Sciences 104.21 (2007): 8685-8690.

[4] A. Hamelin, et al, K-core decomposition: A tool for the visualization of large scale networks. arXiv preprint cs/0504107 (2005).

[5] R. E. Tarjan, Decomposition by clique separators, Discrete Math, vol. 55, no. 2, pp. 221-232, July. 1985.