The Conjunctive Disjunctive Node Kernel

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

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. Gene-disease associations are inferred on the basis of similarities between genes. Biological relationships that are exploited to define similarities range from interacting proteins, proteins that participate in pathways and gene expression profiles. Though graph kernel methods have become a prominent approach for association prediction, most solutions are based on a notion of information diffusion that does not capture the specificity of different network parts. Here we propose a graph kernel method that explicitly models the configuration of each gene's context. An empirical evaluation on several biological databases show that our proposal is competitive w.r.t. state-of-the-art kernel approaches.

1 Introduction and Related Work

Predictive systems for gene-disease associations are often based on a notion of similarity between genes. A common strategy is to encode relations between genes as a network and use graph based techniques to make useful inferences. In the last decades, a number of graph kernel methods have been proposed that directly exploit transitive properties in biological networks. The prototypical method is the Diffusion kernel (DK) [2] inspired by the heat diffusion phenomenon. The key idea is to allow a given amount of *heat* placed on nodes to *diffuse* through the edges. The similarity between two nodes v_i, v_j is then defined as the amount of heat starting from v_i and reaching v_j within a given time interval. In DK the heat flow is proportional to the number of paths connecting two vertices, which introduces an undesired bias that penalize peripheral nodes w.r.t. central ones. This problem is tackled by a modified version of DK called Markov exponential diffusion kernel (MED) [3] where a Markov matrix replaces the Laplacian matrix. Another kernel called Markov diffusion kernel (MD) [4], exploits instead the notion of *diffusion distances*, a measure of similarity between patterns of heat diffusion. The Regularized Laplacian kernel (RL) [5] represents instead a normalized version of the random walk with restart model and defines the node similarity as the number of paths connecting two nodes with different lengths. All these approaches can be applied to dense networks with high degree nodes. A drawback of these approaches is however their relatively low discriminative capacity. This is in part due to the fact that information is processed in an additive and independent fashion which prevents them from accurately modeling the configuration of each gene's context. To address this issue here we propose to employ a *decompositional* graph kernel (DGK) [1] technique. To exploit its higher discriminative capacity we first decompose the network in a

collection of connected sparse graphs and then we develop a suitable kernel, that we call Conjunctive Disjunctive Node Kernel (CDNK).

2 Method

We start from the type of similarity notion computed by decomposition kernels between graph instances and adapt it to express the similarity between nodes in a single network. In this work we use three key ideas: 1) genes are labeled using their functional profile, 2) we transform the network to distinguish highly connected components from sparsely connected ones, and 3) we transform the neighborhood of each gene in a sparse high dimensional vector that can be easily processed by standard machine learning techniques such as SVMs.

2.1 Gene Labeling

The type of networks created to model gene-disease associations problems typically represent genes as nodes labeled with a gene identifier. Here we take a different approach: since we want to build a predictive system based on the configuration of each gene's context, we use as labels a discretized functional annotation based on the *gene ontology* [10]. More precisely, we use the gene ontology to construct binary vectors representing the bag-of- words encoding for each gene. The resulting representations are then clustered to finally yield a discretized label for each gene. In this way genes that have similar functional profiles will receive the same coarse identifier as label.

2.2 Network Decomposition

Gene-disease associations networks typically have nodes with a high degree. These networks cannot be meaningfully processed by decomposition kernels based on exact neighborhood matches. Instead we propose to decompose the network in a linked collection of sparse sub-networks where each node has a reduced connectivity.

Notation and Definitions. A graph $G = (V, E)$ is a structure that consists of a node set $V(G)$ and an edge set $E(G)$. The *distance* between two vertices 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 vertex v is the set of vertices at a distance no greater than r from v and denoted by $N_r(v)$. The *neighborhood subgraph* with radius r of vertex 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 $N_r(v)$.

K-core Decomposition: the K-core decomposition [8] intend to decompose a given graph to have a graph without any node degree greater than D. Kcore contains an iterative process in which each round consists of a procedure to alternately extract a high degree and a low degree subgraph from the input graph. The high degree subgraph (G_H) is the induced subgraph on the set of nodes with degree bigger than D, meanwhile the low degree subgraph (G_L) is the one on the set of nodes with degree smaller then D. At the first round, we

takes G as the input. At the step i, the input graph is $G_{H_{i-1}}$ taken from the output of the step $(i-1)$. The iteration stops when the G_{H_i} of the output is empty. When the decomposition process is done, we form a decomposed graph by making the union of all G_{L_i} taken from all rounds. We then add the set of edges from G that are not present in any G_{L_i} as the *disjunctive* edges of the decomposed graph.

Clique Decomposition: The clique decomposition begins with finding the set of cliques L in G that have size no less than C. For each clique in L, first, we add a new node to G. We then connect the new node to clique's nodes with disjunctive edges and to all neighborhood of clique's nodes at distance 1 with conjunctive edges. Finally, we remove all conjunctive edges that have end points at one of the clique nodes.

2.3 Node Graph Kernels

The principle is: neighborhoods counts. We will modify from graph to nodes in this way: xx

2.3.1 The Original Decomposition Kernel

The The Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) is an instance of convolution kernel which is designed for measuring the similarity between graphs.

Given a graph $G \in \mathcal{G}$ ($\mathcal{G}$ is the graph domain) and two rooted graphs A_u, B_v, the relation $R_{r,d}(A_u, B_v, G)$ is defined to be true iff A_u and B_v are in $\{\mathcal{N}_r^v : v \in V(G)\}$, where A_u (B_v) needs to be isomorphic with some $\mathcal{N}_r$ and $\mathcal{D}(u,v) = d$. We denote R^{-1} as the inverse relation that returns subgraphs of G, $R_{r,d}^{-1}(G) = \{A_u, B_v | R_{r,d}(A_u, B_v, G)\}$. The kernel $\kappa_{r,d}$ over $\mathcal{G} \times \mathcal{G}$ takes into account the number of identical neighboring graph pairs with radius r at distance d between two graph and is formulated as:

$$\kappa_{r,d}(G, G') = \sum_{\substack{A_v, B_u \in R_{r,d}^{-1}(G) \\ A'_{v'}, B'_{u'} \in R_{r,d}^{-1}(G')}} \mathbf{1}_{A_v \cong A'_{v'}} \cdot \mathbf{1}_{B_u \cong B'_{u'}},$$

where $\mathbf{1}_{A \cong B}$ is the *exact matching function* that returns 1 if A is isomorphic to B and 0 otherwise. In order to solve the graph isomorphism problem, an efficient approximate algorithm is also proposed in [6]. Finally, the NSPDK is defined as $K(G, G') = \sum_r \sum_d \kappa_{r,d}(G, G')$. For efficiency issue, we limit the values of r and d with the upper bounds r^* and d^*, respectively.

2.3.2 The Conjunctive Disjunctive Node Kernel

In this section, we describe our proposed kernel, CDNK, which is a modification of NSPDK to measure the node similarity in the graph which consists of conjunctive and disjunctive edges. In our kernel, we consider only conjunctive edges when computing the distance between nodes and extracting neighborhood

subgraphs. We define two relations: the *conjunctive relation* $R^{\wedge}_{r,d}(A_u, B_v, G)$ to be true iff (i) A_u and B_v are in $\{\mathcal{N}^v_r : v \in V(G)\}$, where A_u (B_v) needs to be isomorphic with some $\mathcal{N}_r$, (ii) $\mathcal{D}(u, v) = d$; and the *disjunctive relation* $R^{\vee}_{r,d}(A_u, B_v, G)$ to be true iff (i) A_u and B_v are in $\{\mathcal{N}^v_r : v \in V(G)\}$, where A_u (B_v) needs to be isomorphic with some $\mathcal{N}_r$, (ii) there exists a vertex w such that $\mathcal{D}(u, w) = d$, (iii) (w, v) is a disjunctive edge.

We define $\kappa_{r,d}$, an instance of the DGK on the relation $R^{\wedge}_{r,d}$ and $R^{\vee}_{r,d}$ as

$$\kappa_{r,d}(G_u, G_v) = \sum_{\substack{A_u,\ A'_{u'} \in ({R^{\wedge}_{r,d}}^{-1}(G) \cup {R^{\vee}_{r,d}}^{-1}(G)) \\ B_v,\ B'_{v'} \in ({R^{\wedge}_{r,d}}^{-1}(G) \cup {R^{\vee}_{r,d}}^{-1}(G))}} \mathbf{1}_{A_u \cong B_v} . \mathbf{1}_{A'_{u'} \cong B'_{v'}},$$

$\kappa_{r,d}$ counts the number of identical pairs of neighboring graphs of radius r at a distance d between two vertices. The CDNK is finally defined as

$$K(G_u, G_v) = \sum_r \sum_d \kappa_{r,d}(G_u, G_v).$$

3 Evaluation

We perform an empirical evaluation of the predictive performance of several kernel based methods on two of the datasets used in [3].

BioGPS: A gene co-expression network is constructed from BioGPS dataset, which contains 79 tissues in duplicates, measured with the Affymetrix U133A array. Pairwise Pearson correlation coefficients (PCC) are calculated and a pair of genes are linked by an edge if the PCC value is larger than 0.5.

Pathways: Pathway information is retrieved from KEGG, Reactome, PharmGKB and the Pathway Interaction Database. If a couple of proteins co-participate in any pathway, the two corresponding genes are linked.

3.1 Evaluation Methods

We evaluate the performance of graph node kernels in the gene prioritization problem. Given a set of genes known to be associated to a given disease, gene prioritization is a task that aims 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 12 diseases in which each one contains at least 30 confirmed genes. For each disease, we construct a positive set $\mathcal{P}$ and a negative set $\mathcal{N}$. The set $\mathcal{P}$ consists of all disease gene members. The set $\mathcal{N}$ is built by randomly picking genes from known disease genes, genes associated at least to one disease class, but not related to the current class, such that $|\mathcal{N}| = \frac{1}{2}|\mathcal{P}|$. After that, leave-one-out cross validation is used to evaluate the performance of the algorithm. Each turn one gene is out to be the test gene and the rest are used to train the model using SVM. Starting from the output scores, we compute a decision score q_i for the test gene g_i as the top percentage value of score s_i among all scores. $q_i = \frac{|\{j|s_i \geq s_j\}|}{N}, i = 1, 2, \ldots, N$, where s_i, s_j are scores, N is the length of gene list. We collect all decision scores

	BioGPS					**Pathways**				
Disease	K1	K2	K3	K4	K5	K1	K2	K3	K4	K5
1	52/5	57/4	59/3	59/2	**65/1**	75/5	76/4	79/3	79/2	**80/1**
2	82/2	79/3	75/4	75/5	**88/1**	55/5	65/4	77/3	77/2	**81/1**
3	64/4	60/5	72/2	72/1	66/3	55/5	63/4	64/3	66/2	67/1
4	65/4	58/5	68/3	68/2	**72/1**	54/5	65/4	**74/1**	74/2	66/3
5	64/5	64/4	67/2	66/3	**76/1**	53/5	56/4	63/2	63/3	**68/1**
6	75/2	70/5	71/4	71/3	**79/1**	83/5	93/4	97/2	**97/1**	93/3
7	73/3	67/5	75/2	**76/1**	69/4	85/5	88/4	89/2	**90/1**	89/3
8	74/5	**77/1**	76/3	76/2	75/4	54/5	66/4	72/3	72/2	**73/1**
9	**72/1**	66/5	68/3	70/2	67/4	53/5	65/2	64/4	64/3	**81/1**
10	54/3	50/5	56/2	51/4	**78/1**	69/3	65/5	74/2	**74/1**	67/4
11	58/4	51/5	59/3	59/2	**72/1**	54/5	69/4	75/2	74/3	**77/1**
$\overline{AUC}$	66.6	63.5	67.8	67.5	**73.3**	62.7	70.1	75.3	75.5	**76.5**
$\overline{Rank}$	3.5	4.3	2.8	2.5	**2.0**	4.8	3.9	2.5	2.0	**1.8**

Table 1: *The performance of kernels on different genetic diseases using BioGPS and Pathway dataset. Each element in the table shows the AUC in percentage and the order of kernel comparing to the rest (AUC/Rank). K1 = DK, K2 = MD, K3 = MED, K4 = RL, K5 = CDNK.*

for every gene in the training set of a disease to form a global decision score list. The performance of the algorithm is measured by using AUC calculated that list.

3.2 Parameter Selection

In order to select the optimal parameter values for each kernel, we use one disease gene set for parameter selection and use Kfold with k equals to 3. We set the values for diffusion parameter in DK and MED as $\{10^{-3}, 10^{-3}, 10^{-2}, 10^{-1}\}$, for time steps in MD as $\{1, 10, 100\}$ and for RL parameter as $\{1, 4, 7\}$. For CDNK, we set values for degree threshold 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 set as $\{10^{-5},\ 10^{-4},\ 10^{-3},\ 10^{-2},\ 10^{-1},\ 1,\ 10,\ 10^{2}\}$.

4 Results and Discussion

Table 1 shows the AUC performance of the models trained by using different graph node kernels on 11 genetic diseases using BioGPS and Pathways datasets. In the table, the best result on each disease is marked in bold. By observing the results, we note that the kernel CDNK perform the best results comparing with other considered kernels. Particularly, the CDNK is ranked at the first order in seven out of 11 diseases on both datasets. It also illustrates the highest results in

average AUC and rank with 73.3/2.0, 76.5/1.8, and the AUC difference with the second best ones are 5.5% and 1% on BioGPS and Pathways, respectively. The MED and RL show similar and moderate results with small gap between them. Last, DK and MD demonstrate modest performance in average comparing with other adopted kernels. They are ranked in the last position in many diseases, especially 7 times out of 11 for MD in BioGPS and 10 out of 11 for DK in Pathways. While DK shows better performance than MD in BioGPS, it presents worse in Pathways. In conclusion, CDNK outperforms all employed graph node kernels in term of both average rank and AUC measure.

The CDNK shows the state of the art results. However, in the case that the input graph has high average node degree and they are uniformly distributed, the decomposed graph is too sparse and it can lead our kernel to the poor performance.

5 Conclusions

We have shown how decomposing a network in a set of connected sparse graphs allows us to take advantage of the greater discriminative power of CDNK, a novel decomposition kernel, and achieve state- of-the-art results. In future work we will investigate how to extend the CDNK approach to gene-disease association problems exploiting multiple heterogeneous information sources.

References

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

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

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

[4] 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.

[5] 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.

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

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

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

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

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