Neurocomputing 000 (2018) 1–10

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

The conjunctive disjunctive graph node kernel for disease gene prioritization

Dinh Tran Van[a], Alessandro Sperduti[a], Fabrizio Costa[b,*]

[a] Department of Mathematics, Padova University, Trieste, 63, Padova 35121, Italy
[b] Department of Computer Science, University of Exeter, Exeter EX4 4QF, UK

ARTICLE INFO

Article history:
Received 2 July 2017
Revised 14 January 2018
Accepted 14 January 2018
Available online xxx

Keywords:
Graph node kernels
Graph decomposition
Disease gene prioritization

ABSTRACT

Disease gene prioritization plays an important role in disclosing the relation between genes and diseases and it has attracted much research. As a consequence, a high number of disease gene prioritization methods have been proposed. Among them, graph-based methods are the most promising paradigms due to their ability to naturally represent many types of relations using a graph representation. One key factor of success of graph-based learning methods is the definition of a proper graph node similarity measure normally measured by graph node kernels. However, most approaches share two common limitations: first, they are based on the diffusion phenomenon which does not effectively exploit the nodes' context; second, they are not able to process the auxiliary information associated to graph nodes.

In this paper, we propose an efficient graph node kernel, based on graph decompositions, that not only is able to effectively take into account nodes' context, but also to exploit additional information available on graph nodes. The key idea is to learn and generalize from small network fragments present in the neighborhood of genes of interest. An empirical evaluation on several biological databases shows that our proposal achieves state-of-the-art results.

1. Introduction

The advancement of experimental technologies allows ever larger amounts of data to be gathered from which powerful statistical analysis can be performed to gain deeper insights on natural phenomena. In Biological and Medical domains, high throughput techniques have exponentially lowered the cost to acquire information on cellular events. The abundance of data however needs to be matched by a proportional capacity of data elaboration. In the Biomedical field, disease-gene association recovery is a major goal that has received much attention. Despite significant progress in the last decades, the typical number of genes that can be associated to a genetic disease is quite limited. In order to find out the "missing" unknown set of related genes one can employ some form of reasoning by analogy and search in specific regions of the genome that contain large numbers of genes (candidate genes) known to be somehow related. In order to reduce the expensive empirical validation phase to as few candidates as possible, many gene prioritization methods, which automatically predict a list of candidate genes sorted according to the probability to be actually involved in the target disease, have been proposed in literature. A quite exhaustive survey on disease gene prioritization methods can be seen in [6] or in [7–9]. Methods for gene-disease association are often based on a notion of similarity between genes inferred from the available knowledge encoded in biological databases. Many solutions employ machine learning methods to robustly predict gene-disease associations. A common strategy is to encode relations between functionally related genes in a network and then employ graph based techniques to make useful inferences, as done in [1,2,10].

A key element in graph-based disease gene prioritization approaches is the definition of the node similarity measure. Node similarity is often measured by graph node kernels [2,11–13]. The state-of-the-art graph node kernels used to measure node similarity are based on the notion of "information diffusion", which depends on the number of paths connecting two nodes in the graph. These graph node kernels suffer however when working with sparse graphs (i.e., graphs with a low number of links) because of the following limitations. First, they are defined using a heat diffusion dynamics which sums up the contributions of all paths from one node to another one, disregarding the local topological context of the nodes and considering the single contribution of one path as independent with respect to the contributions of other paths. Second, they do not take into account auxiliary information (i.e. properties of a single node) associated to

* Corresponding author.
E-mail address: F.Costa@exeter.ac.uk (F. Costa).

https://doi.org/10.1016/j.neucom.2018.01.089
0925-2312/

individual nodes. These additional information can complement the information encoded in the graph topology and potentially significantly improve the expressiveness of the kernel.

In this paper we define an expressive kernel that can improve the performance of several graph-based prioritization approaches. We propose a decompostional graph node kernel, named *Conjunctive Disjunctive Node Kernel* (CDNK) which is able to: *i*) exploit the nodes' context and *ii*) make use of auxiliary node information. In particular, first the biological network is transformed into a set of linked sub-networks that still preserves all the available relational information. Then the similarity between nodes (genes) is computed on the basis of the two neighborhoods rooted in each node. Finally we integrate additional node information using a convolution operator between the information extracted by the graph topology and the auxiliary information encoded in a flexible way as real valued vectors.

2. Background

In this section, we first introduce definitions and notations that are used to define our proposed method. We then describe the state-of-the-art concerning graph node kernels.

2.1. Definitions and notations

A graph is a structure $G = (\mathbb{V}, \mathbb{E}, \mathcal{L}_1, \mathcal{L}_2)$ where $\mathbb{V}$, $\mathbb{E}$, $\mathcal{L}_1, \mathcal{L}_2$ are the vertex (node) set, link (edge) set, discrete labeling fuction and real vector labeling function, respectively. The functions $\mathcal{L}_1, \mathcal{L}_2$ are defined as:

- $\mathcal{L}_1 : \mathbb{V} \longmapsto \mathbb{L}$, where $\mathbb{L}$ is a set of discrete labels. $\mathcal{L}_1$ assigns a single discrete label $\ell \in \mathbb{L}$ for each node $v \in \mathbb{V}$, $\mathcal{L}_1(v) = \ell$.
- $\mathcal{L}_2 : \mathbb{V} \longmapsto \mathbb{R}^n$. $\mathcal{L}_2$ assigns a single real vector label $(v_1, v_2, \ldots, v_n) \in \mathbb{R}^n$ for each ndoe $v \in \mathbb{V}$, $\mathcal{L}_2(v) = (v_1, v_2, \ldots, v_n)$.

We define the length of a shortest path between u and v, denoted as $\mathcal{D}(u, v)$, as the number of edges on a shortest path between them. The *neighborhood* of a node u with radius r, $N_r(u) = \{v \mid \mathcal{D}(u, v) \leq r\}$, is the set of nodes at distance no greater than r from u. The corresponding *neighborhood subgraph* $\mathcal{N}_r^u$ is the subgraph induced by the neighborhood (i.e. considering all the edges with endpoints in $N_r(u)$). The *degree* of a node u, $deg(u) = |\mathcal{N}_1^u|$, is the cardinality of its neighborhood for $r = 1$. The maximum node degree in the graph G is $deg(G) = \max_{v \in \mathbb{V}} deg(v)$.

Definition 1. An adjacency matrix $\boldsymbol{A}$ 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 $w_{ij} \in \mathbb{R}$ when there exists a link connecting v_i and v_j, and is 0 otherwise.

Definition 2. The Laplacian matrix $\boldsymbol{L}$ is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D}$ is the diagonal matrix with non-null entries equal to the summation over the corresponding row of the adjacency matrix, i.e. $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}_{ij}$.

Definition 3. The Transition matrix of a graph G, denoted as $\boldsymbol{P}$, is a matrix with entries $P_{ij} = A_{ij} / \sum_i A_{ij}$. When considering a random walk in G, P_{ij} can be interpreted as proportional to the probability of moving from node v_i to node v_j.

2.2. Kernels on graphs

Kernel methods have emerged as one of the most powerful framework in machine learning. They have been successfully applied in various domains, due to their modularity, i.e. the definition of kernel functions is independent from the design of the learning algorithm.

A kernel function can be considered as the similarity measure between input instances, whatever the nature of the instances may be, e.g. vectors, sequences, trees, graphs. Formally, a kernel $k : \mathbb{X} \times \mathbb{X} \longmapsto \mathbb{R}$, where $\mathbb{X}$ is a set of entities, is a function satisfying the following properties: *i*) k is symmetric, i.e., $k(x_1, x_2) = k(x_2, x_1)$, where $x_1, x_2 \in \mathbb{X}$; *ii*) k is positive semi-definite, that is $\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j k(x_i, x_j) \geq 0$ for any $N > 0$, $c_i, c_j \in \mathbb{R}$, and $x_i, x_j \in \mathbb{X}$.

Canonical machine learning methods take vectorial data, i.e. numerical vectors that collect the results of measures on features of the input instances, as their input. However, there are many fields where data is naturally represented by structured forms, such as graphs, one of the most popular representations for structured data. Two interesting examples of domains involving graphs are Chemistry and the Web. In Chemistry, chemical compounds are represented via their molecular graphs and typical computational tasks consist in the prediction of their physicochemical properties. Thus, in this domain, the target function to learn is a mapping from one graph to a real value. The Web can be described as a huge graph, where nodes are web pages and edges are links from one page to another one. A typical task in this context is to automatically predict the topics covered by the textual content of a page on the basis of the characteristics of web pages connected with it. So, the target function to learn is a mapping from one node of the graph to a set of discrete values. Therefore, the definition of a kernel function for graphs has to take into account one of the two scenarios described above. In the first case, we talk about graph kernels, while in the second case we talk about graph node kernels. Both graph kernels and graph node kernels are widely applied to build graph-based learning systems for fields ranging from Social Sciences, to Recommendation Systems, and Biology.

In general, one important contribution with respect to the design of kernels for structured data, and in particular for graphs, has been given by Haussler, who proposed a convolution-based framework for the definition of decompositional kernels [14]. In the following, we shortly describe some of the state-of-the-art graph (node) kernels underpinning our proposed approach.

2.2.1. Graph kernels

The task of designing efficient and expressive graph kernels plays an important role in the development of graph-based predictive systems. Existing graph kernels are decompositional kernels and can be classified into two categories: sequence-based graph kernels and subgraph-based graph kernels. The sequence-based graph kernels decompose graphs into "parts" in sequence-based forms, such as paths and walks. Typical examples of sequence-based graph kernels are the product graph kernel [15], and the shortest path kernel [16]. The subgraph-based graph kernels decompose graphs into subgraphs. Examples include the Weisfeiler–Lehman kernel [17,18], and the Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) [19]. This latter category of kernels are generally more effective because sub-graphs are more expressive than walks and paths. Moreover, they can be computed quite efficiently thanks to sparsity of representation that allows an explicit encoding of the graph via hash functions. In the following, we describe NSPDK, since it is later adopted to develop the proposed graph node kernel (presented in Section 3).

The NSPDK [19] is an instance of convolution kernel [14] where a given graph G is decomposed in features (pairwise neighborhood subgraphs) constituted by couples of subgraphs of radius r rooted at nodes of G which are at distance d. More formally, given two rooted graphs A_u, B_v, where u and v are nodes of G, the relation $R_{r,d}(A_u, B_v, G)$ is true *iff* $\mathcal{D}(u, v) = d$ and $A_u \cong \mathcal{N}_r^u$ is (up to isomorphism $\cong$) a neighborhood subgraph of radius r of G as well as $B_v \cong \mathcal{N}_r^v$. Fig. 1 illustrates a pairwise neighborhood subgraph. We denote with R^{-1} the inverse relation that returns all pairs of neighborhoods of radius r at distance d in G, $R_{r,d}^{-1}(G) =$

Please cite this article as: D. Tran Van et al., The conjunctive disjunctive graph node kernel for disease gene prioritization, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.01.089