ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

DEEP: Decomposition Feature Enhancement Procedure for Graphs

Dinh Van Tran[1], Nicolò Navarin[1] and Alessandro Sperduti[1] *

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

Abstract. When dealing with machine learning on graphs, one of the most successfully approaches is the one of kernel methods. Depending if one is interested in predicting properties of graphs (e.g. graph classification) or to predict properties of nodes in a single graph (e.g. graph node classification), different kernel functions should be adopted. In the last few years, several kernels for graphs have been defined in literature that extract local features from the input graphs, obtaining both efficiency and state-of-the-art predictive performances. Recently, some work has been done in this direction also regarding graph node kernels, but the majority of the graph node kernels available in literature consider only global information, that can be not optimal for many tasks. In this paper, we propose a procedure that allows to transform a local graph kernel in a kernel for nodes in a single, huge graph. We apply a specific instantiation to the task of disease gene prioritization from the bioinformatics domain, improving the state of the art in many diseases.

1 Introduction

The abundance of relational datasets has led to the fast increase of large-scale graph-based inference systems. Examples of such systems are ranging from Biomedicine [1] to Social networks [2], to Recommendations [3]. The inference in these cases consists in the prediction of properties associated to the nodes in the graph. When dealing with graph data, one of the most successful approaches is the one of kernel methods, whose performances are strongly affected by the choice of the kernel function. However, the task of designing kernels for graph nodes that show high performance in a wide range of different domains is not trivial and it normally faces the trade-off between expressiveness and efficiency.

In the last two decades, many graph node kernels have been proposed and applied in different systems from various fields. They can be classified in two classes: diffusion-based and subgraph (or decomposition)-based. The former is in general fast to compute, while the latter tend to show higher predictive performance, while being more computationally demanding and more difficult to define. However, decomposition-based kernels are common when considering kernels between pairs of graphs (referred to as graph kernels).

In this paper, we propose a procedure to define graph node kernels starting from decomposition-based graph kernels. We start from the Weisfeiler-Leman

*This work was supported by the University of Padova, Department of Mathematics, DEEPer project.

(WL) graph kernel [4], and we define a corresponding graph node kernel, considering a richer set of features. The empirical evaluation on several disease gene prioritization tasks shows that our kernel achieves state-of-the-art performances.

2 Background

In this section, we start providing some basic definitions and notation. We then revise the state-of-the-art concerning graph node kernels, and finally describe the WL subtree kernel, which is later used to develop a specific instance of our proposed graph node kernel framework.

We consider a graph as a triplet $G = (V, E, \lambda(\cdot))$, in which V is the set of nodes (or vertices), E is the edge set, and $\lambda : V \longrightarrow \mathcal{L}$ is the node labelling function that assigns a discrete label in $\mathcal{L}$ to each node in the graph. Given a vertex v, the d-neighbors of as all the nodes with *shortest path* distance exactly d from v, i.e. $\mathcal{N}_v^d = \{u|\ |shortest\_path(v,u)| = d\}$. A kernel $k(\cdot,\cdot)$ is a positive semi-definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that corresponds to a dot product in a *Reproducing Kernel Hilbert Space*, i.e. $k(x,y) = \langle \phi(x), \phi(y) \rangle$, where $\phi : \mathcal{X} \to \mathbb{R}^z$ is a feature map projecting nonlinearly any $x \in \mathcal{X}$ into a real-valued vector space of dimension $z \gg 1$. With $\phi_i(x)$ we denote the i-th entry of $\phi(x)$. Note that $\mathcal{X}$ can be any space.

In this paper, we consider $\mathcal{X}$ to be the space of nodes in a huge graph G, and we refer to the kernel as a *graph node kernel*. Existing graph node kernels can be classified into two categories: diffusion-based kernels and subgraph (or decomposition) based kernels. Diffusion-based graph node kernels measure the similarity between two nodes by taking into account global measures, based on random walks, related to two nodes. The most commonly used diffusion-based graph node kernel is the Laplacian exponential diffusion kernel (LED) [5] which is based on the heat diffusion phenomenon. In LED, the similarity between two nodes is proportional to the number of paths connecting them. Therefore, the similarities between high degree nodes are normally higher than the ones between nodes with low degree. This problem is solved in the Markov Exponential Diffusion kernel (MED) [6], a modification of LED which introduces a normalization that replaces the Laplacian matrix by the Markov matrix. A similar graph node kernel is the Markov Diffusion kernel (MD) [7] that defines the similarity between two nodes by measuring how similar the patterns of their heat diffusion are, i.e. it expresses how much nodes "influence" each other. The Regularized Laplacian kernel (RL) [8] counts the number of walks connecting two nodes on the graph induced by taking the negative Laplacian matrix as the adjacency matrix. Diffusion-based graph kernels are relatively fast to compute. However, they are not able to effectively exploit the local connectivity of nodes, so they do not show a high discriminative capacity.

Decomposition-based graph node kernels consider local subgraphs which are associated to two nodes when computing their similarities. As an example, the Conjuctive Disjunctive Node Kernel (CDNK) [9] is a graph node kernel that is based on NSPDK, a kernel for graphs [10]. First, the input graph is transformed into a set of linked connected components in which two types of links, "conjunctive" and "disjunctive", are introduced and treated in different manners. Nodes linked by conjunctive links are used jointly to define the notion of context, while nodes linked by disjunctive edges are instead only used to define features. Second, the features of a node u are defined as the subset of NSPDK features that have the node u as one of the roots. Finally, the kernel computes the number of identical features between two nodes. CDNK has the problem of depending on a large number of parameters that can be problematic in the model selection process. Moreover, its graph preprocessing step could be computationally expensive. Intuitively, subgraph-based kernels are able to capture the local relationships in the graph, that can be beneficial for some tasks. However, the definition of such kernels is not straightforward. The WL subtree kernel [4] is a kernel defined between pairs of graphs. Given a graph G the WL algorithm computes, at each iteration $i = 0, \dots, h^\star$ (maximum number of iterations), a new labelling function $\lambda_i : V \to \mathcal{L}_i$ defined as: $\lambda_i(v) = f_\#(\lambda_{i-1}(v), sort(\{\lambda_{i-1}(u) | u \in N_v^1\}))$, where $f_\#$ is a hashing function, $sort$ returns a sorted list of labels, $\lambda_0 = \lambda$ (the original set of labels for all graphs), $\mathcal{L}_i$ is the set of all labels generable at iteration i by all graphs, and $\mathcal{L}_i \cap \mathcal{L}_j = \emptyset$ if $i \neq j$. Each label generated at the i-th iteration by $f_\#$ can be interpreted as a subtree-walk of depth i. Moreover, each $\lambda_i(v)$ is associated to (rooted in) a specific node v. Let $\mathcal{L}_{WL}^{h^\star} = \bigcup_{i=0}^{h^\star} \mathcal{L}_i$. The WL kernel measures the similarity between two graphs by counting the number of identical labels (subtree walks) between two graphs at the different WL iterations. We can define the WL feature mapping $\phi^{WL}(G)$ as: $\phi_j^{WL}(G) = \sum_{v \in V} \sum_{i=0}^{h^*} \delta(\lambda_i(v), \sigma_j)$, where δ is the Kronecker's delta function, and σ_j is the j-th element[1] of $\mathcal{L}_{WL}^{h^\star}$. The WL kernel is then defined as $k_{WL}^{h^\star}(G, G') = \langle \phi^{WL}(G), \phi^{WL}(G') \rangle$.

3 Proposed approach

In this section, we detail our proposed procedure. We can define decomposition features as features that depend on local substructures of the graph. There are many proposals in literature that define graph kernels based on decomposition features. Among the others, the NSPDK kernel [10], the ODD$_{ST}$ kernel [11, 12], and the WL kernel [4], described in Section 2. In the original formulations, decomposition features are exploited to define graph kernels. Our proposal is a procedure to define a graph node kernel out of them.

We achieve this goal with the following steps: *i*) we obtain graph node features (that are rooted decomposition features) from a slight modification of the existing graph kernels (in this paper, we focus on WL); *ii*) we define a procedure to combine such features in order to obtain a discriminative graph node kernel.

Weisfeiler-Lehman node features. We can easily define a WL feature mapping for each node v in a graph as follows (we drop the WL subscript for ease of notation): $\phi_j(v) = \sum_{i=0}^{h^*} \delta(\lambda_i(v), \sigma_j)$.

[1] Any predetermined enumeration of the elements of $\mathcal{L}_{WL}^{h^\star}$ can be used.

Feature Enhancement Procedure. Our proposed feature enhancement procedure takes inspiration from the NSPDK kernel: we generate a new feature space composed of pairs of the original WL features, i.e. $\mathcal{L}_{WL}^{h^\star} \times \mathcal{L}_{WL}^{h^\star}$. However, we restrict our features to pairs where the generating nodes are no more distant than a predefined maximum distance $d^\star$.

Thus, given nodes u and v, we can define the DEEP graph node kernel as:

$$DEEP_{WL,h^*}^{d^*}(u,v) = \phi(u)^\top \phi(v) \left(\sum_{0 \leq d \leq d^*} \sum_{s \in \mathcal{N}_u^d} \sum_{t \in \mathcal{N}_v^d} \phi(s)^\top \phi(t) \right), \qquad (1)$$

where we recall $\mathcal{N}_v^d$ is the set of nodes at shortest-path distance d from v. Note that, since the dot product is linear, it is possible to pre-compute $\sum_{s \in \mathcal{N}_v^d} \phi(s)$ for each node v in the graph and for each d. The feature space of this kernel, fixed a node v, is composed by pairs of WL features from dimension 0 up to $h^\star$, rooted in the vertices v and $s \in \mathcal{N}_v^d$. The resulting kernel combines the features of the base kernel, allowing for the modeling of complex relationships. Eq. 1 is positive demi-definite kernel since it is an instance of convolution kernels [13].

4 Experimental setup

We evaluate our proposed node kernel in the context of disease gene prioritization. Given a genetic graph, a set of genes known to be associated to a disease and a set of candidate genes, first a graph node kernel is adopted to compute a gram-matrix which encodes similarities between any couple of genes. Then this gram-matrix is fed into a kernel machine to build a model. Finally, the obtained model is used to rank the candidate genes. For the considered datasets, the nodes represent genes ($7,311$ genes in total). We considered two separate networks derived from the *BioGPS* and the *Pathways* datasets. *BioGPS* is a gene co-expression network including 79 tissues, measured with the Affymetrix U133A array. A link is formed between two genes when their pairwise Pearson correlation coefficient (PCC) is larger than 0.5. The total number of links for this dataset is $911,294$. *Pathways* is a dataset retrieved from KEGG, Reactome, PharmGKB and the Pathway Interaction databases. Two genes are linked, if their corresponding proteins co-participate in any pathway. The number of edges for this graph is $2,254,822$. To label the nodes in the graphs, we adopt the node labeling method proposed in [9]. Each gene is represented by a binary vector where each element shows whether a Gene Ontology [14] term is associated with the gene or not. The gene representations are then clustered (hyper-parameter L) and genes belonging to the same cluster are labeled with their cluster identifier. We followed the experimental procedure in [9] and [6], in which 12 diseases [15] with at least 30 confirmed genes are used. For each disease, a dataset including a positive set $\mathcal{P}$ and a negative set $\mathcal{N}$ is constructed such that $(|\mathcal{N}| = \frac{1}{2}|\mathcal{P}|)$. The set $\mathcal{P}$ contains all confirmed disease genes, while the set $\mathcal{N}$ contains random genes associated at least to one disease and not related to the considered disease. Leave-one-out cross validation is adopted to evaluate the performance of each method. We compute a decision score q_i for each test gene g_i following [6]. We collect the decision scores for every gene in the dataset to form a global ranking on which we compute the area under the ROC curve (AUC). The hyper-parameters of the proposed method were tuned using a 10-fold CV on the dataset 0 in the following ranges: $i^* \in \{1,\ 2,\ 3\}$, $d^* \in \{1,\ 2,\ 3\}$, the SVM C parameter in $\{10^{-5},\ 10^{-4},\ 10^{-3},\ 10^{-2},\ 10^{-1}, 1,\ 10,\ 10^{2}\}$, and the number of clusters L in $\{10,\ 15,\ 20,\ 25\}$. For the hyper-parameters of the other kernels please refer to [9].

5 Results and discussion

DG	1	2	3	4	5	6	7	8	9	10	11	$\overline{AUC}$	$\overline{Rank}$
BioGPS													
DK	51.9/6	81.7/2	64.3/5	65.3/5	64.0/6	74.6/3	73.0/4	74.4/5	71.5/2	54.0/4	58.2/5	66.6	4.27
MD	57.4/5	78.5/3	59.6/6	58.2/6	64.1/5	70.2/6	66.7/6	76.8/1	65.6/6	50.3/6	51.3/6	63.5	5.09
ME	58.8/4	75.2/4	71.6/2	67.8/4	66.5/3	71.0/5	75.4/3	76.2/3	67.7/4	56.1/3	59.3/4	67.8	3.55
RL	59.2/3	75.0/5	71.8/1	67.8/3	66.2/4	71.2/4	75.6/2	76.4/2	69.9/3	51.1/5	59.3/3	67.6	3.18
CD	65.1/2	88.3/1	66.5/4	71.9/2	75.9/1	79.3/2	68.8/5	74.7/4	66.8/5	77.6/1	71.8/2	73.3	2.64
DE	74.4/1	71.2/6	69.5/3	81.9/1	67.4/2	79.4/1	79.1/1	65.5/6	72.0/1	71.9/2	82.5/1	74.1	2.27
Pathways													
DK	74.7/6	55.1/6	55.0/6	54.3/6	52.9/6	83.4/6	84.6/6	53.7/6	52.5/6	68.8/4	53.7/6	62.6	5.8
MD	76.4/4	64.9/5	62.7/5	65.2/5	55.7/5	92.7/4	88.3/4	65.6/5	64.9/3	65.4/6	69.2/5	70.1	4.6
ME	78.7/3	76.6/3	64.1/4	73.7/2	62.7/3	96.5/2	89.4/2	72.0/4	64.2/5	74.4/3	74.6/3	75.2	3.1
RL	78.8/2	76.6/2	65.6/3	73.7/3	62.7/4	96.5/1	89.5/1	72.3/3	64.2/4	74.4/2	74.1/4	75.3	2.6
CD	80.2/1	81.1/1	67.1/2	66.1/4	68.3/2	93.0/3	88.5/3	72.5/2	81.3/1	66.9/5	76.9/2	76.6	2.36
DE	76.4/4	75.3/4	70.1/1	77.9/1	75.2/1	84.5/5	86.4/5	73.4/1	72.2/2	77.8/1	86.3/1	77.8	2.36

Table 1: AUC/Rank of the considered kernels on 11 gene-disease associations using networks induced by the *BioGPS* and the *Pathway* databases. Best results are underlined. ME = MED, CD = CDNK, DE = DEEP (our proposal), DG = Disease gene association, $\overline{AUC}$= average AUC, $\overline{Rank}$=average Rank.

Table 1 reports the performance in AUC and rank of the different graph node kernels on the two considered networks. The table shows that, in general, subgraph-based kernels (CDNK and DEEP) perform better with respect to diffusion-based ones (DK, MD, MED and RL). In particular, CDNK and DEEP kernels show an average improvement in AUC of 5.5% and 6.3% with respect to the best diffusion-based kernel on the *BioGPS* dataset, respectively. As for the *Pathways* dataset, the improvement is AUC is 1.3% and 2.5%, respectively. The proposed DEEP kernel performs slightly better than CDNK with respect to the mean AUC in both datasets (improvement of 0.8% and 1.2%, respectively). In both datasets, our proposed DEEP kernel is ranked at the first place in 6 out of 11 diseases, with the second best performing kernel being CDNK with 3 first places. Moreover, our proposed DEEP kernel is the one with lower average rank in both the datasets. Looking at the single diseases, in some cases, such as disease 4 and 11 on *BioGPS* and disease 11 on *Pathways*, DEEP shows an AUC around 10% higher than the best competing method. In general, our proposed

DEEP node kernel shows good predictive performance, performing better than all the other considered kernels in the majority of our experiments.

6 Conclusions and future work

In this paper, we have proposed a procedure for defining graph node kernels starting from decomposition-based graph kernels. An instantiation of such procedure have been evaluated on the task of disease gene prioritization, showing state-of-the-art predictive performance. In future, we plan to apply such procedure to other graph kernels, and to define an approximated version of the procedure with improved efficiency.

References

[1] Giorgio Valentini, Alberto Paccanaro, Horacio Caniza, Alfonso E. Romero, and Matteo Re. An extensive analysis of disease-gene associations using network integration and fast kernel-based gene prioritization methods. *Artificial Intelligence in Medicine*, 61(2):63–78, 2014.

[2] A. Anil, N. Sett, and S.R. Singh. Modeling evolution of a social network using temporal graph kernels. In *SIGIR*, 2014.

[3] Zan Huang, Wingyan Chung, Thian-Huat Ong, and Hsinchun Chen. A graph-based recommender system for digital library. In *proceedings of JCDL*, page 65, 2002.

[4] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-Lehman Graph Kernels. *JMLR*, 12:2539–2561, 2011.

[5] Risi Imre Kondor and John Lafferty. Diffusion Kernels on Graphs and Other Discrete Input Spaces. In *ICML*, pages 315–322, 2002.

[6] BoLin Chen, Min Li, JianXin Wang, and Fang-Xiang Wu. Disease gene identification by using graph kernels and markov random fields. *Science China. Life Sciences*, 57(11):1054, 2014.

[7] Francois F Fouss, Luh Yen, Alain Pirotte, and Marco Saerens. An Experimental Investigation of Graph Kernels on a Collaborative Recommendation Task. In *ICDM*, pages 863–868. IEEE, 2006.

[8] Pavel Chebotarev and Elena Shamis. The Matrix-Forest Theorem and Measuring Relations in Small Social Groups. *Automation and Remote Control*, 58(9):10, 1997.

[9] Dinh Van Tran, Alessandro Sperduti, and Fabrizio Costa. The Conjunctive Disjunctive Node Kernel. In *ESANN*, 2017.

[10] Fabrizio Costa and Kurt De Grave. Fast neighborhood subgraph pairwise distance kernel. In *ICML*, pages 255–262. Omnipress, 2010.

[11] Giovanni Da San Martino, Nicolò Navarin, and Alessandro Sperduti. A Tree-Based Kernel for Graphs. In *SDM*, pages 975–986, 2012.

[12] Giovanni Da San Martino, Nicolò Navarin, and Alessandro Sperduti. Ordered Decompositional DAG Kernels Enhancements. *Neurocomputing*, 192:92–103, 2016.

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

[14] Gene Ontology Consortium et al. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl 1):D258–D261, 2004.

[15] Kwang-Il Goh, Michael E Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-László Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, 2007.