Graph Kernels Based on Relevant Patterns and Cycle Information for Chemoinformatics

Benoit Gaüzère, Luc Brun, Didier Villemin, Myriam Mokhtari

Graph Kernels Based on Relevant Patterns and Cycle Information for Chemoinformatics

Benoit Gaüzère, Luc Brun, Didier Villemin, Myriam Brun
GREYC UMR CNRS 6072, LCMT UMR CNRS 6507, Université de Caen, Caen, France,
{*benoit.gauzere,luc.brun,didier.villemin,myriam.brun*}*@ensicaen.fr*

Abstract

Chemoinformatics aim to predict molecule's properties through informational methods. Computer science's research fields concerned with chemoinformatics are machine learning and graph theory. From this point of view, graph kernels provide a nice framework for combining these two fields. We present in this paper two contributions to this research field: a graph kernel based on an optimal linear combination of kernels applied to acyclic patterns and a new kernel on the cyclic system of two graphs. These two extensions are validated on two chemoinformatics datasets.

1 Introduction

Chemoinformatics aims to predict molecules properties from their structural similarity. Most of existing methods are based on fingerprints defined as collections of descriptors such as the boiling point, logP, molar refractivity, etc. An alternative strategy consists to extract a set of descriptors directly from the molecular graph $G = (V, E, \mu, \nu)$, where the unlabeled graph (V, E) encodes the structure of the molecule while μ maps each vertex to an atom's label and ν characterizes a type of bond between two atoms (single, double, triple or aromatic). Considering this representation, similarity between molecules can be deduced from the similarity of their molecular graphs. Graph kernels can be understood as symmetric graph similarity measures. Using a semi definite positive kernel, the value $k(G, G')$ where G and G' encode two graphs corresponds to a scalar product between two vectors $\psi(G)$ and $\psi(G')$ in an Hilbert space. Graph kernels thus provide a natural connection between structural and statistical pattern recognition fields. A large family of kernels is based on a bag of patterns. These methods extract a bag of patterns from the graphs and deduce similarity between graphs from similarity between their bags. Most of existing methods are defined on linear patterns [3]. Such methods generally have a low complexity but are limited by the lack of expressivity of linear patterns on graphs. In order to use more structural information, some methods are defined on non-linear patterns. For example, tree-pattern kernel [4] is based on an implicit enumeration of tree-patterns, ie. trees where a node can appear more than once. Another approach, described in Section 2 and called treelet kernel [1], computes an explicit enumeration of a limited set of subtrees.

The above methods don't take into account the cyclic information encoded within molecular graphs. Nonetheless, cycles have an impact on molecules behavior and must be taken into account. Horváth proposed to combine the tree pattern kernel with an intersection kernel defined on a set of simple cycles of a graph. Despite the high complexity of the enumeration of all simple cycles of a graph, this method can be efficiently used when a set of graphs has a low number of cycles. In order to tackle the complexity required by the enumeration of all simple cycles, Horváth proposed in [2] to use a subset of simple cycles. This set is first initialized using the set of relevant cycles [9]. Then, additional simple cycles are iteratively enumerated by combining relevant cycles and newly discovered cycles. Horváth showed that a low number of iterations is sufficient to obtain similar results than the ones obtained using all simple cycles.

This paper presents two contributions: We propose in Section 3 a treelet kernel based on cyclic information. Unlike existing methods based on cycles, this kernel encodes topological relationships between relevant cycles. Our second contribution (Section 4) allows us to associate a weight to each treelet found in a training set. These weights are incorporated within our treelet kernel and are optimal according to a given regression or classification task on the training set. Finally, Section 5 shows results obtained by these two contributions on different chemoinformatics problems.

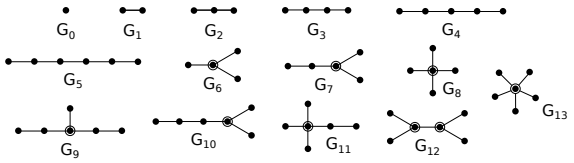

Figure 1. Set of treelet's structures

2 Treelet Kernel

Treelet kernel [1] is a graph kernel defined as a convolution kernel between bags of patterns extracted from graphs. The set of extracted patterns, called treelets, is composed of all labeled trees with a number of nodes lower than or equal to 6 (Figure 1). The first step of our bag construction scheme enumerates the set of tree structures from a graph. This structure identification step uses two different approaches: First, a depth first traversal is performed from each node of the graph in order to enumerate linear patterns. Then, each non linear pattern is enumerated using a neighborhood analysis of n-star nodes. Such nodes correspond to nodes having a degree equal to n.

Once this structure identification step is performed, a key encoding the labels of each treelet is computed. This key, based on Morgan numbering, provides a sequence of nodes and edges labels which is unique for two isomorphic treelets. Conversely, we have shown [1] that two treelets with a same index (a same structure) and a same key are isomorphic. The concatenation of treelet's index and treelet's key defines a unique code for each treelet which allows us to perform an explicit enumeration of all treelets included within a graph. Based on this enumeration, we define a function f which associates to each graph G a vector $f(G)$, each component of this vector being equal to the number of occurrences of a given treelet t in G:

$$f(G) = (f_t(G))_{t \in \mathcal{T}(G)} \text{ with } f_t(G) = |(t \trianglelefteq G)| \quad (1)$$

where $\mathcal{T}(G)$ denotes the set of treelets extracted from G and $\trianglelefteq$ the subgraph isomorphism relationship. Then, similarity between treelet distributions is computed using a sum of subkernels between treelet's frequencies:

$$K_{\mathcal{T}}(G, G') = \sum_{t \in \mathcal{T}(G) \cap \mathcal{T}(G')} k(f_t(G), f_t(G')) \quad (2)$$

where $k(.,.)$ defines any positive definite kernel between real numbers such as the linear kernel, Gaussian kernel or intersection kernel. Note that, unlike tree pattern methods, this method explicit enumerates sub trees by computing the number of occurrences of each pattern. This explicit enumeration allows us to apply a treelet weighting step, as defined in Section 4.

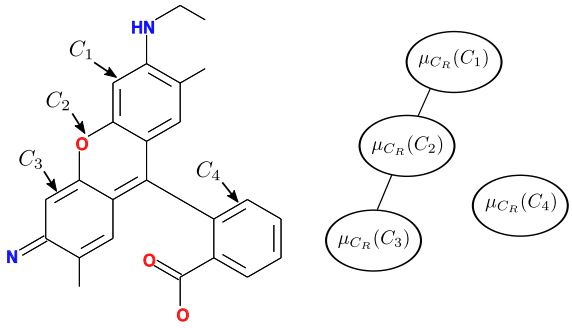

Figure 2. A cyclic system and its graph representation. Canonical key $\mu_{C_R}(C_2)$ is equals to C1C1C1O1C1C2C.

3 Kernel on Relevant Cycles

3.1 Relevant Cycle Graph

A simple cycle is defined as a subgraph $C = (V', E', \mu, \nu)$ of $G = (V, E)$ where each vertex $v \in V'$ has a degree equal to 2. Each cycle $C \subseteq G$ can be represented as a vector $\vec{C} \in \{0,1\}^{|E|}$ where $\vec{C}_i$ equals 1 if i is an edge of C and is 0 otherwise. The set of vectors encoding the cycles of G defines a vector space where the addition of two cycles C and C' corresponds to a XOR bitwise [9]. The set of relevant cycles, $\mathcal{C}_{\mathcal{R}}$, is then defined by the union of all bases of the vector space of minimum length. The length of a base being defined as the sum of lengths of its cycles. This first step allows us to compute a canonical set of cycles with a polynomial complexity according to the number of nodes of the graphs.

Topological relations between relevant cycles can be encoded by the relevant cycle graph [8]. This graph is defined as $G_{\mathcal{C}} = (\mathcal{C}_{\mathcal{R}}, E_{\mathcal{C}_{\mathcal{R}}}, \mu_{\mathcal{C}_{\mathcal{R}}}, \nu_{\mathcal{C}_{\mathcal{R}}})$ where each vertex encodes a relevant cycle and two vertices are connected by an edge if their cycles share at least one vertex of the initial graph (Figure 2). According to [8], a labeling function $\mu_{\mathcal{C}_{\mathcal{R}}}(v)$ is defined as the number of edges composing the cycle of v while $\nu_{\mathcal{C}_{\mathcal{R}}}(e)$ is defined as the couple $(|v(C_1) \cap v(C_2)|, (|e(C_1) \cap e(C_2)|)$ where $v(\mathcal{C}_{\mathcal{R}})$ and $e(\mathcal{C}_{\mathcal{R}})$ respectively denote the set of vertices and the set of edges of a cycle. These two labeling functions only encode the size of cycles and of their connections. In order to include more information within relevant cyclic graphs, we propose to redefine these two labeling functions as follows:

- $\mu_{\mathcal{C}_{\mathcal{R}}}(C)$: Each cycle C is defined by a sequence of edge and vertex labels encountered during the traversal of C. In order to obtain a sequence invari-

ant to cyclic permutations, $\mu_{\mathcal{C}_\mathcal{R}}(C)$ is defined as the sequence having the lowest lexicographic order.

- $\nu_{\mathcal{C}_\mathcal{R}}(e)$: An edge e in G_C encodes a path between two cycles and is described by a sequence of edge and vertex labels. Since such a path may be traversed from its two extremities, we define $\nu_{\mathcal{C}_\mathcal{R}}(e)$ as the sequence of lowest lexicographic order.

In order to encode both the similarities between the cycles of two graphs and the relationships between these cycles, we apply our treelet kernel defined in Section 2 on cyclic graphs using our edge and vertex labeling functions. This kernel is thus defined as follows:

$$K_\mathcal{C}(G, G') = \sum_{t_\mathcal{C} \in \mathcal{T}(G_\mathcal{C}) \cap \mathcal{T}(G'_\mathcal{C})} k(f_{G_\mathcal{C}}(t_\mathcal{C}), f_{G'_\mathcal{C}}(t_\mathcal{C})) \tag{3}$$

Unlike cyclic pattern kernel (Section 1) based on a comparison of set of simple cycles, this kernel encodes both the similarity between sets of relevant cycles (pattern G_0 in Figure 1) and relationships between these cycles (all remaining treelets in Figure 1). The proposed edge and vertex labels of the cyclic graph allows us to identify all acyclic combinations composed of at most 6 cycles. In addition, Horváth's method requires at most n^k operations to perform k iterations on n relevant cycles. On the other hand, our treelet kernel requires at most nd^5 operations, where d denotes the maximal degree of relevant cycle nodes. Our method thus has a linear complexity according to the number of relevant cycles of a graph. Such a kernel may be computed efficiently as soon as the degree of the vertices of G_C remains bounded.

4 Treelet Weighting

As shown in Section 5, the number of different treelets enumerated from the dataset can be huge and some of them may not be related to the property to be predicted. Therefore, considering these treelets in Equations 2 or 3 leads to inaccurate predictions. In order to tackle this drawback, we must weight each treelet according to its relevance for a given property. To this end, we propose to adapt a multiple kernel learning method called SimpleMKL [6] to variable selection. SimpleMKL defines an optimal weighting of any linear combination of N kernels according to a prediction task:

$$K_{\text{MKL}}(x, x') = \sum_{i=1}^{N} w_i * k_i(x, x') \tag{4}$$

where $w_i \in \mathbb{R}_+$ and $\sum_i^N w_i = 1$. The treelet kernel (Equations (2) and (3)) is defined as a sum on the intersection of two sets of treelets. Given the set of treelets $\mathcal{T}$ computed over a training set, we define the kernel $k_t(G, G')$ specific to each treelet $t \in \mathcal{T}$ as:

$$k_t(G, G') = \begin{cases} 0 \text{ if } f_t(G) \text{ or } f_t(G') = 0 \\ k\ (f_t(G), f_t(G')) \text{ otherwise} \end{cases} \tag{5}$$

Using Equation (5), Equations (2) and (3) may be written as:

$$K_\mathcal{T}(G, G') = \sum_{t \in \mathcal{T}} k_t(G, G') \tag{6}$$

where the sum is performed over the finite set $\mathcal{T}$. Using this definition, the treelet kernel can be adapted to the SimpleMKL formulation as follows:

$$K_W(G, G') = \sum_{t \in \mathcal{T}} w_t * k_t(G, G') \tag{7}$$

where w_t denotes the optimal weight defined using the SimpleMKL method. Since SimpleMKL induces a sparsity constraint on the objective function, only relevant treelets will be selected. Non zero weights may be interpreted as treelet influence measures on the property to predict.

5 Experiments

5.1 Regression Problem

Our first experiment evaluates the relevance of our treelet weighting procedure (Section 4) by a regression task on molecule's boiling points using a dataset[1] composed of 185 acyclic molecules. Prediction is performed using a 10-fold cross validation. The first line of Table 1 shows results obtained by the random walks kernel [3]. Due to the limited expressivity of linear patterns, this method doesn't correctly predict the boiling point property. Line 2 shows results obtained by a Gaussian kernel based on graph edit distance [5]. This method obtains a better result than a kernel based on linear patterns. In the same way, tree pattern [4] and treelet kernels improve the accuracy of prediction models (Table 1, Lines 3 and 4). Finally, the three last lines show results obtained by two treelet selection methods described in [1] and our treelet weighting method defined in Section 4. Using this dataset, 138 different treelets have been enumerated. As shown by Line 7, the lowest root mean squared error (RMSE) is obtained using the SimpleMKL algorithm. A major difference

[1] This database is available on the IAPR TC15 Web page: http://www.greyc.ensicaen.fr/iapr-tc15/links.html#chemistry

Table 1. Boiling point prediction.

Method	RMSE (°C)
1 Random Walks Kernel	18.72
2 Gaussian edit distance	10.27
3 Tree Pattern Kernel	11.02
4 Treelet Kernel (TK)	8.10
5 TK + Forward Selection	7.05
6 TK + Backward Elimination	6.75
7 TK + MKL	**5.24**

between the MKL method and forward selection and backward elimination methods, which may explain this result, is that the SimpleMKL algorithm provides an optimal real weighting on the training set while forward and backward eliminations only a set of selected treelets [1]. Treelet kernel on cycle graphs has not been tested on this experiment since the dataset is exclusively composed of acyclic molecules.

5.2 Classification Problems

The second experiment is a classification problem taken from the Predictive Toxicity Challenge [7] which aims to predict carcinogenicity of chemical compounds applied to female (F) and male (M) rats (R) and mice (M). This experiment is based on ten different datasets, each of them being composed of one trainset and one testset. Table 2 shows the number of correctly classified molecules over the ten testsets for each method and for each class of animal. As shown by Table 2, Lines 1 and 2, Horváth's cycle kernel [2] obtains better classification accuracy than our treelet kernel, hence showing the importance of cyclic information in this experiment. Our cyclic kernel, taking into account both relevant cycles and their relationships, outperforms the one of Horvárth's (Lines 2 and 3) which does not encode labels between cycles. As shown by Lines 4 and 5, the selection of treelets using the SimpleMKL algorithm allows us to improve the classification accuracy of both our treelet and cyclic kernels. Note that SimpleMKL allows us to reduce the number of treelets from about 3500 to 150, depending on dataset. Finally, a weighted sum of both kernels (Line 6), with both weights set by cross-validation, allows us to slightly improve the individual results of each kernel. This clear separation between cyclic and acyclic information is one reason which may explain the better classification accuracy of our final kernel compared to the one obtained by the Gaussian Edit Distance kernel (Line 7).

Table 2. Classification accuracy on PTC.

Method	MM	FM	MR	FR
1 Treelet Kernel (TK)	208	205	209	212
2 Horváth	209	207	202	228
3 TK on Cycles (TC)	211	210	203	232
4 TK + MKL	217	224	223	250
5 TC + MKL	216	213	212	237
6 (TK + αTC) + MKL	219	**226**	**226**	**251**
7 Gaussian Edit Distance	**223**	212	194	234

6 Conclusion

We have proposed a new kernel between cyclic systems of molecules. This kernel compares both the relevant cycles of two graphs and the adjacency relationships between these relevant cycles. This kernel may be computed with a linear complexity according to the number of relevant cycles. We also proposed to adapt a multiple kernel learning framework in order to select the most relevant patterns of our kernels for a prediction task. Further works will aim to improve the definition of our graph of relevant cycles in order to take into account more information about cycles within our kernel.

References

[1] B. Gaüzère, L. Brun, and D. Villemin. Two new graph kernels and applications to chemoinformatics. *Pattern Recognition Letters (to be published)*, 2012.

[2] T. Horváth. Cyclic pattern kernels revisited. *PAKDD 2005*, pages 791–801, 2005.

[3] H. Kashima, K. Tsuda, and A. Inokuchi. *Kernels for graphs*, chapter 7, pages 155–170. MIT Press, 2004.

[4] P. Mahé and J.-P. Vert. Graph kernels based on tree patterns for molecules. *Machine Learning*, 75(1)(September 2008):3–35, 2009.

[5] M. Neuhaus and H. Bunke. *Bridging the gap between graph edit distance and kernel machines*. World Scientific Pub Co Inc, 2007.

[6] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[7] H. Toivonen, A. Srinivasan, R. King, S. Kramer, and C. Helma. Statistical evaluation of the predictive toxicology challenge 2000-2001. *Bioinformatics*, 19(10):1183–1193, 2003.

[8] P. Vismara. *Reconnaissance et représentation d'éléments structuraux pour la description d'objets complexes. Application à l'élaboration de stratégies de synthèse en chimie organique*. PhD thesis, Université Montpellier II, 1995.

[9] P. Vismara. Union of all the minimum cycle bases of a graph. *The Electronic Journal of Combinatorics*, 4(1):73–87, 1997.