Graph Kernels: Crossing Information from Different Patterns using Graph Edit Distance

Benoit Gaüzère, Luc Brun, Didier Villemin

Graph Kernels: Crossing Information from Different Patterns using Graph Edit Distance

Benoit Gaüzère[1], Luc Brun[1], and Didier Villemin[2]

[1] GREYC CNRS UMR 6072, Caen, France,
benoit.gauzere@ensicaen.fr
[2] LCMT CNRS UMR 6507, Caen, France

Abstract. Graph kernels allow to define metrics on graph space and constitute thus an efficient tool to combine advantages of structural and statistical pattern recognition fields. Within the chemoinformatics framework, kernels are usually defined by comparing number of occurences of patterns extracted from two different graphs. Such a graph kernel construction scheme neglects the fact that similar but not identical patterns may lead to close properties. We propose in this paper to overcome this drawback by defining our kernel as a weighted sum of comparisons between all couples of patterns. In addition, we propose an efficient computation of the optimal edit distance on a limited set of finite trees. This extension has been tested on two chemoinformatics problems.

1 Introduction

Chemoinformatics aims to predict molecule's properties from their structural similarity. Most of existing methods are based on fingerprints defined as collections of descriptors such as 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 provide thus a natural connection between structural and statistical pattern recognition fields. A large family of kernels is based on bags of patterns. These methods extract a bag of patterns from each graph and deduce graph's similarity from bag's similarity by comparing the number of occurrences of each pattern within both graphs. Most of existing methods are defined on linear patterns [6]. Such methods have generally 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 based on non linear patterns, such as the tree-pattern kernel [7]. This last method is

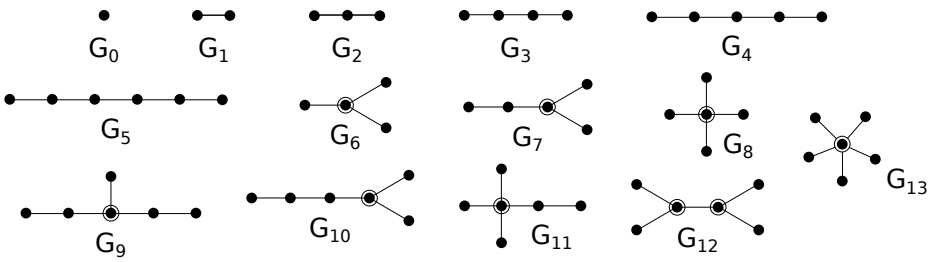


Fig. 1. Set of sub structures enumerated from the graphs.

based on an implicit enumeration of tree patterns, ie. trees where a node can appear more than once.

Another approach, called treelet kernel [4], computes an explicit enumeration of a limited set of subtrees. Treelet kernel is a graph kernel defined as a kernel between two sets of patterns extracted from both graphs to be compared. The set of extracted patterns, called treelets and denoted $\mathcal{T}$, is composed of all labeled trees with a number of nodes lower than or equals to 6 (Figure 1). Based on the enumeration of this set of substructures, each graph G is associated to a vector $f(G)$. Each component of this vector $f_t(G)$ is equals to the number of occurrences of a given treelet t in G:

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

where $\mathcal{T}(G)$ denotes the set of treelets extracted from G and $\trianglelefteq$ the sub graph isomorphism relationship. Using this vector representation, similarity between treelet distributions is computed using a sum of sub kernels between treelet's frequencies:

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

where $k(.,.)$ defines any positive definite kernel between real numbers such as linear kernel, Gaussian kernel or intersection kernel. Unfortunately, similarity of occurrences is only computed between isomorphic patterns and not between similar patterns. From a mathematical point of view, computing similarities only between isomorphic patterns relies to consider that each axis encoding a pattern is orthogonal with all other axis. This assumption is dubious since large patterns are composed by smaller ones, hence encoding partially the same information. Moreover, from a chemical point of view, two sub structures may have a similar influence on a chemical property if they slightly differ, hence showing the interest of crossing information collected from differents treelets.

In order to capture this similarity, we propose to extend treelet kernel by adding comparisons of non isomorphic treelets. In Section 2, we propose to weight the influence of any pair of treelets by their edit distance. In Section 2.1, we propose an efficient way to compute an exact edit distance between treelets. Then, in Section 3, this treelet kernel extension is tested and discussed on an experimental comparison involving two chemoinformatics problems.

2 Inter Treelet Kernel based on Edit Distance

Haussler's convolution kernels [5] are defined on objects $x \in \mathcal{X}$ which can be associated to a decomposition into finite sets $\mathcal{X}_x$. Considering a sub kernel $k : \mathcal{X}_x \times \mathcal{X}_x \to \mathbb{R}$, Haussler's convolution kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is defined as follows:

$$K(x, y) = \sum_{(x', y') \in \mathcal{X}_x \times \mathcal{X}_y} k(x', y') \tag{3}$$

By considering a decomposition $\mathcal{X}_G = \{(t, f_t(G)) | t \trianglelefteq G\}$ of each graph and a tensor product $(k \otimes k')$ of two kernels $k' : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ and $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, treelet kernel (Equation 2) can be reformulated as a convolution kernel:

$$K(G, G') = \sum_{\substack{(t, f_t(G)) \in \mathcal{X}_G \\ (t', f_{t'}(G')) \in \mathcal{X}_{G'}}} (k' \otimes k)(t, f_t(G), t', f_{t'}(G'))$$

$$K(G, G') = \sum_{\substack{(t, f_t(G)) \in \mathcal{X}_G \\ (t', f_{t'}(G')) \in \mathcal{X}_{G'}}} k'(t, t') k(f_t(G), f_{t'}(G')) \tag{4}$$

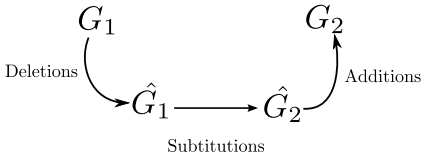
where $k(f_t(G), f_{t'}(G'))$ is defined as in Equation 2 and $k'(t, t') = 1 \iff t \simeq t', 0$ otherwise. Note that $k'(t, t')$ is equal to 1 only if treelet t in $\mathcal{T}(G)$ is isomorphic to t' and thus belongs simultaneously to $\mathcal{T}(G)$ and $\mathcal{T}(G')$ (Equation 2). Such a definition of $k'(t, t')$ restricts comparison of occurrences to isomorphic treelets. In order to relax this restriction and based on the assumption that similar structures should have a similar chemical activity, we propose to define $k'(t, t')$ in Equation 4 as a measure of similarity between t and t'. This similarity measure is based on the graph edit distance defined as the sequence of operations transforming G into G' with a minimal cost [8]. Such a sequence, called an edit path, may include vertex or edge addition, removal and relabeling. Given a cost function $c(.)$ associated to each operation, the cost of a sequence of operations is defined as the sum of each elementary operation's costs. A high edit distance indicates a low similarity between two graphs while a small one indicates a strong similarity. Unfortunately, trivial kernels defined on graph edit distance are not always semi definite positive and thus does not define valid kernels. In order to define semi definite positive kernels, we apply a regularization scheme as defined by [4, 8]. According to [8], the computational cost of the exact edit distance grows exponentially with the size of graphs. To overcome this problem, Fankhauser and al. [3] propose a method to compute an approximate edit distance in $O(n^3)$ where n is equals to the number of nodes and to the maximal degree of both graphs. Such an edit distance computation provides an efficient way to compute an approximate edit distance between graphs at the cost of a lower precision.

2.1 Exact Treelet Edit Distance

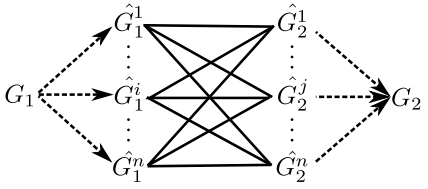
Exact edit distance is hard to compute when considering the whole set of possible graphs. Given a finite set of n structures $B = \{(V_1, E_1), \ldots, (V_n, E_n)\}$, we thus

restrict our study to sets of graphs D such that for any $G = (V, E, \mu, \nu) \in D$ we have $(V, E) \in B$. We show in the remaining of this section that within this framework, exact edit distance may be computed within a reasonable computational time using ad hoc methods. In order to present such methods, let us introduce some common definitions. A graph $G' = (V', E', \mu', \nu')$ is a structural sub graph of $G = (V, E, \mu, \nu)$, denoted $G' \trianglelefteq_s G$, iff $V' \subseteq V$ and $E' \subseteq E \cap (V' \times V')$. In addition, if $\mu'_{|V'} = \mu$ and $\nu'_{|E'} = \nu$, $f_|$ denoting the restriction of function f to a particular domain, then G' is a sub graph of G, denoted $G' \trianglelefteq G$. A graph $G = (V, E, \mu, \nu)$ is structurally isomorphic to a graph $G' = (V', E', \mu', \nu')$, denoted $G \simeq_s G'$ iff there exists a bijective function $f : V \to V'$ such that $(u, v) \in E \Leftrightarrow (f(u), f(v)) \in E'$. If $\mu' \circ f = \mu$ and $\nu' \circ f = \nu$, then G is isomorphic to G', denoted $G \simeq G'$. If $G = G'$ then f is called an automorphism. If f is only injective then it exists a sub graph isomorphism between G and G'. A graph $\hat{G}$ is a maximal common sub graph of G_1 and G_2 if it is a sub graph of G_1 and G_2 and if it is not a sub graph of any other common sub graph of G_1 and G_2. A graph $\hat{G}$ is called a maximum common sub graph of G_1 and G_2 if it is a common sub graph of G_1 and G_2 with a maximal number of nodes.The notions of maximal structural sub graph and maximum structural sub graph are defined the same way using the notion of structural sub graph.

Under mild assumptions [1], the sequence of edit operations encoding an edit path can be ordered into a sequence of deletions, substitutions and additions as illustrated in Figure 2(a). The first sequence transforms the initial graph G_1 into one of its sub graphs $\hat{G}_1$ by deleting a set of nodes corresponding to $V_1 - \hat{V}_1$ and a set of edges corresponding to $E_1 - \hat{E}_1$. The second sequence represents the set of substitutions transforming $\hat{G}_1$ into $\hat{G}_2$. This set of substitutions defines a one to one matching between $\hat{V}_1$ and $\hat{V}_2$ on the one hand and between $\hat{E}_1$ and $\hat{E}_2$ on the other. Substitutions matching two elements having a same label are denoted as identical substitutions. Finally, the last sequence corresponds to the addition of a set of nodes and edges in order to transform $\hat{G}_2$ into G_2. Note that the set of operations transforming $\hat{G}_1$ into $\hat{G}_2$ is only composed of substitutions



(a) Different steps describing an edit path.



(b) Possible edit paths passing through maximum common structural sub graphs $\{\hat{G}_1^1, \ldots, \hat{G}_1^n\}$ and $\{\hat{G}_2^1, \ldots, \hat{G}_2^n\}$. Dashed lines correspond to structural operations, other to substitutions.

Fig. 2. General edit path scheme and edit paths passing through maximum common structural sub graphs.

which do not modify the structure of graphs. Therefore, $\hat{G_1}$ and $\hat{G_2}$ have a same structure and correspond to two structurally isomorphic sub graphs of G_1 and G_2. We define costs on edit operations as non negative constant functions for edges (c_{e*}) and vertex (c_{v*}) deletions (c_{*d}), insertions (c_{*i}) or substitutions (c_{*s}). In addition, the cost associated to an identical substitution is equals to 0 since such an operation does not modify the graph. Using the representation described in Figure 2(a) and cost functions previously defined, the cost of any edit path is equals to:

$$\gamma(P) = |V_1 - \hat{V_1}|c_{vd} + |E_1 - \hat{E_1}|c_{ed} + V_f c_{vs} + E_f c_{es} + |V_2 - \hat{V_2}|c_{vi} + |E_2 - \hat{E_2}|c_{ei} \quad (5)$$

with V_f, resp. E_f, denoting the number of non identical substitutions on nodes, resp. edges, required to transform $\hat{G_1}$ into $\hat{G_2}$. Bunke have shown that under some slightly different conditions on edge operations, constraining the costs to $c_{vd} + c_{vi} < c_{vs}$ and $c_{es} < c_{vs}$ induces that $\hat{G_1} \simeq \hat{G_2}$ correspond to a maximum common sub graph of G_1 and G_2 [2]. However, maximum common sub graph of two graphs depends both on structure and labels. This last point does not allow us to use efficiently our assumption that the number of different structures of any set of graphs is bounded and known a priori. We propose to study if different conditions between costs can lead to a possible efficient algorithm to compute an exact edit distance.

Proposition 1. *Given two graphs G_1, G_2, let us denote by δ_v the number of vertices of their maximum structural common sub graph and by δ_e, the maximal number of edges, of their structural common sub graphs. If $\frac{c_{vd}+c_{vi}}{c_{vs}} \geq \delta_v + \frac{c_{es}}{c_{vs}}\delta_e$ and $\frac{c_{ed}+c_{ei}}{c_{es}} \geq \delta_e + \frac{c_{vs}}{c_{es}}\delta_v$, then $\hat{G_1}$ is a maximal common structural sub graph of G_1 and G_2.*

Proof. [1]

Considering two graphs G_1 and G_2, this first proposition ensures that sequences of structural operations transform G_1 and G_2 into one of their maximal common structural sub graphs. Since maximal common structural sub graph does not depend on labeling information, the set of maximal common structural sub graphs may be pre computed between any pair of structure belonging to B. However, this number may be large hence forbidding an efficient pre computation of the exact edit distance. By restricting conditions on costs, we obtain a relationship leading to a reduced set of sub structures:

Proposition 2. *Let us suppose that $c_{ed} = c_{ei} = 0$ and $c_{es} \leq c_{vs}$. Given two graphs G_1 and G_2, let us further denotes by δ_v the number of vertices of their maximum common structural sub graphs and by δ_e the maximal number of edges of all maximum common structural sub graphs. Then if $\frac{c_{vd}+c_{vi}}{c_{vs}} \geq \delta_v + \delta_e$, $\hat{G_1}$ is a maximum common structural sub graph of G_1 and G_2.*

Proof. [1]

Proposition 2 states that under some hypothesis on the costs c_{*d}, c_{*i} and c_{*s} any optimal edit path between two graphs G_1 and G_2 should pass through one of their maximum common structural sub graphs. Let us consider two graphs G_1 and G_2 and without loss of generality let us suppose that these two graphs share only one maximum common structural sub graph $\hat{G} = (\hat{V}, \hat{E})$. Let us denote as $\{\hat{G_1^0}, \ldots, \hat{G_1^i}, \ldots, \hat{G_1^{n_1}}\}$ and $\{\hat{G_2^0}, \ldots, \hat{G_2^i}, \ldots, \hat{G_2^{n_2}}\}$ the sets of sub graphs of G_1 and G_2 structurally isomorphic to G (Figure 2(b)). By Proposition 2, any optimal edit path P between G_1 and G_2 should pass through one G_1^i and G_2^j. The cost associated to P can be decomposed into two parts: a structural cost $\gamma_{struc}(P)$, corresponding to insertion and deletion operations, and a substitution cost $\gamma_{label}(P)$, corresponding to the label substitutions required to transform $\hat{G_1^i}$ into $\hat{G_2^j}$:

$$\gamma(P) = \gamma_{struc}(P) + \gamma_{label}(P) \tag{6}$$

Following Equation 5, we have:

$$\begin{cases} \gamma_{struc}(P) = |V_1 - \hat{V_1}|c_{vd} + |E_1 - \hat{E_1}|c_{ed} + |V_2 - \hat{V_2}|c_{vi} + |E_2 - \hat{E_2}|c_{ei} \\ \gamma_{label}(P) = V_f c_{vs} + E_f c_{es} \end{cases} \tag{7}$$

For any $i \in \{1, \ldots, n_1\}$, since $\hat{G_1^i} \trianglelefteq G_1$, we have $\hat{V_1^i} \subseteq V_1$ and $\hat{E_1^i} \subseteq E_1$ and thus:

$$\begin{cases} |\hat{V_1^i} - V_1| = |V_1| - |\hat{V_1^i}| = |V_1| - |\hat{V}| \\ |\hat{E_1^i} - E_1| = |E_1| - |\hat{E_1^i}| = |E_1| - |\hat{E}| \end{cases} \tag{8}$$

Similarly, the same holds for G_2 and $\hat{G_2^j}$ for any $j \in \{1, \ldots, n_2\}$. Structural cost corresponding to edit path P is thus equals to:

$$\begin{aligned} \gamma_{struct}(P) = & |V_1|c_{vd} + |V_2|c_{vi} + |E_1|c_{ed} + |E_2|c_{ei} \\ & - |\hat{V}|(c_{vd} + c_{vi}) - |\hat{E}|(c_{ed} + c_{ei}) \end{aligned} \tag{9}$$

Computing substitution cost $\gamma_{label}(P)$ (Equation 7) relies on computing the number of non identical node substitutions V_f and edge substitutions E_f transforming $\hat{G_1^i}$ into $\hat{G_2^j}$. Let $\Phi(\hat{G})$ denotes the set of structural automorphisms of $\hat{G}$. Given both sub graphs $\hat{G_1^i}$ and $\hat{G_2^j}$, each automorphism $\phi \in \Phi(\hat{G})$ induces a mapping of $\hat{G_1^i}$ onto $\hat{G_2^j}$ and thus a substitution of the label of each vertex v (resp. edge e) of G_1^i onto the label of $\phi(v)$ (resp. $\phi(e)$) in $\hat{G_2^j}$. More precisely, let us denote by $P_{i,j,\phi}$ the edit path associated to the triplet $(\hat{G_1^i}, \hat{G_2^j}, \phi)$. the number of non identical substitutions V_f and E_f induced by $P_{i,j,\phi}$ is equals to:

$$\begin{aligned} V_f(P_{i,j,\phi}) &= |\{v \in \hat{V_1} \mid \hat{\mu_1^i}(v) \neq \hat{\mu_2^j}(\phi(v))\}| \\ E_f(P_{i,j,\phi}) &= |\{(v, v') \in \hat{E_1} \mid \hat{\nu_1^i}(v, v')) \neq \hat{\nu_2^j}(\phi(v), \phi(v'))\}| \end{aligned} \tag{10}$$

Substitution cost of edit path $P_{i,j,\phi}$ is thus equals to $\gamma_{label}(P_{i,j,\phi}) = V_f(P_{i,j,\phi})c_{ns} + E_f(P_{i,j,\phi})c_{es}$. Let us denotes by P_{opt} the edit path minimizing the substitution

cost:

$$P_{opt} = P_{i_0,j_0,\phi_0} \text{ with } (i_0, j_0, \phi_0) = \underset{(i,j,\phi)\in\{1,\ldots,n_1\}\times\{1,\ldots,n_2\}\times\Phi(\hat{G})}{\operatorname{argmin}} \gamma_{label}(P_{i,j,\phi}) \tag{11}$$

Since $\gamma_{struct}(P_{i,j,\phi})$ is the same for any $(i, j, \phi) \in \{1, \ldots, n_1\}\times\{1, \ldots, n_2\}\times\Phi(\hat{G})$ (Equation 9), P_{opt} is an edit path having a minimal cost. Therefore, under our assumptions, the edit path associated to the edit distance is the one which passes through the pair of maximum common structural sub graphs and which minimizes the number of substitutions (Equation 11). This exact edit distance computation algorithm can be applied to treelets since the set of treelets is composed of 14 different structures. In addition, by restricting the set of edit paths to the ones which preserve the connectedness of intermediate graphs [1], we can obtain a lower bound on the ratio between substitutions and insertion/deletion costs.

Proposition 3. *Considering edit paths preserving connectedness and given two trees T_1, $T_2 \in \mathcal{T}$, if $\frac{c_{vd}+c_{vi}}{c_{vs}} \geq \delta_v$ and $\frac{c_{ed}+c_{ei}}{c_{es}} \geq \delta_v - 1$, then $\hat{G}_1$ is a maximum common structural sub tree of T_1 and T_2.*

Proof. [1]

When computing tree edit distance on the set of treelets, δ_v is bounded by 6 and if we define costs as symmetric, i.e. $c_{vd} = c_{vi}$ and $c_{ed} = c_{ei}$, bounds on costs lead to: $c_{vd} > 3c_{vs}$ and $c_{ed} > 2.5c_{es}$. Since the set of treelets represents all trees having a size lower than or equals to 6, the maximum common structural sub tree of two treelets T_1 and T_2 is a treelet. The set of possible sub graphs and automorphisms for any pair of treelets can be easily pre computed since we have to consider only 14 patterns. Therefore, computing exact edit distance between two treelets consists in comparing at most $\max_{i,j\in\{0,\ldots,13\}}(n_i * n_j * |\Phi_{ij}|)$ label sequences where Φ_{ij} denotes the set of automorphisms of the maximum common structural subtree $\hat{T}$ of treelets T_i and T_j and n_i, n_j the numbers of sub trees of T_i and T_j isomorphic to $\hat{T}$. The value of this product on the set of treelets is bounded by 120, hence inducing a constant time complexity for the computation of the exact tree edit distance. Note that, without our restriction to a set of specific tree structures, the complexity of the edit distance calculation between labeled unordered unrooted trees is NP-Complete [9]. In addition, given a trainset D, our kernel is defined as the 0-extension of the kernel defined by matrix $\left(e^{-d(t_i,t_j)}\right)_{(i,j)\in\{1,\ldots,n\}^2}$, where n is the number of different treelets extracted from D. Note that this regularisation has to be performed only once since this kernel only operates on treelets and not directly on graphs.

3 Experiments

Our first experiment evaluates our inter treelet kernel on a regression problem which consists in predicting molecule's boiling points[3]. This dataset is composed

[3] All databases are available on the IAPR TC15 Web page: http://www.greyc.ensicaen.fr/iapr-tc15/links.html#chemistry

Table 1. Boiling point prediction.

Method	RMSE ($^\circ C$)
1 Random Walks Kernel	18.72
2 Gaussian edit distance	10.27
3 Tree Pattern Kernel	11.02
4 Treelet Kernel	8.10
5 Treelet Kernel with backward selection	6.75
6 Inter Treelet Kernel with approximate edit distance	6.09
7 Inter Treelet Kernel with exact edit distance	5.89

of 183 acyclic molecules and prediction is performed using a 10-fold cross validation. The first line of Table 1 shows results obtained by random walks kernel [6]. Due to the limited expressivity of linear patterns, this method does not permit to predict correctly molecule's boiling points. Line 2 shows results obtained by a Gaussian kernel applied on graph edit distance [8]. This last method based on global similarity of graphs obtains a better result than kernel based on linear patterns. In the same way, tree pattern [7] and treelet kernels (Table 1, Lines 3 and 4) improve the accuracy of prediction model based on linear patterns by including information encoded by non linear patterns. Then, Line 5 shows results obtained by combining treelet kernel with a variable selection step [4] which leads to a better prediction accuracy (Table 1, Line 5) on this dataset, at the price of an high computation time. Lines 6 and 7 show results obtained using our inter treelet kernel. First, inter treelet kernel obtains a better prediction accuracy than using treelet kernel restricted to the comparison of similar treelets, hence showing the relevance of including pairs of non isomorphic treelets within kernel computation. Second, we can note that the use of an exact edit distance provides a slightly more accurate weighting than using an approximate edit distance (Table 1, Lines 6 and 7).

Our second experiment is defined as a classification problem on the monoamine oxidase (MAO) dataset which is composed of 68 molecules divided into two classes: 38 molecules inhibit the monoamine oxidase (antidepressant drugs) and 30 do not. Classification accuracy is measured for each method using a leave one out procedure with a two-class SVM. This classification scheme is made for each of the 68 molecules of the dataset. In this experiment, best results are obtained

Table 2. Classification accuracy on the monoamine oxidase (MAO) dataset.

Method	Classification Accuracy
1 Random Walks Kernel	82% (56/68)
2 Gaussian edit distance	90% (61/68)
3 Tree Pattern Kernel	96% (65/68)
4 Treelet Kernel	91% (62/68)
5 Inter Treelet Kernel with approximate edit distance	93% (63/68)
6 Inter Treelet Kernel with exact edit distance	94% (64/68)

using a Tree Pattern Kernel (Table 2, Line 3). Methods based on non linear patterns (Table 2, Lines 3 to 6) outperform methods based on linear patterns (Table 2, Line 1) and graph edit distance (Table 2, Line 2). In addition, the better accuracy obtained by methods crossing information from differents patterns (Table 2, Lines 5 and 6) shows the relevance of the proposed extension. As highlighted on our first experiment, difference between the two methods may be explained by the better accuracy provided by the exact edit distance.

4 Conclusion

In this article, we have presented an extension of the Treelet Kernel which consists in crossing information encoded by non isomorphic treelets according to their structural similarities. In addition, we have defined a new relation between edit distance and maximum common structural sub graphs which leads to an efficient computation of edit distance between treelets. The relevance of this extension has been validated by obtaining a better prediction accuracy than original Treelet Kernel on two chemoinformatics problems. One major perspective of this work is to define the weighting of non isomorphic treelet pairs using their relevance according to a property to predict and no more by an a priori similarity measure such as edit distance.

References

1. L. Brun, B. Gaüzére, and S. Fourey. Relationships between graph edit distance and maximal common unlabeled subgraph. Technical report, CNRS UMR 6072 GREYC, 2012.
2. H. Bunke. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):917–922, 1999.
3. S. Fankhauser, K. Riesen, and H. Bunke. Speeding up graph edit distance computation through fast bipartite matching. In *Graph-Based Representations in Pattern Recognition*, volume 6658, pages 102–111, 2011.
4. B. Gaüzère, L. Brun, and D. Villemin. Two new graph kernels and applications to chemoinformatics. *Pattern Recognition Lett. (In Press)*, 2012.
5. D. Haussler. Convolution kernels on discrete structures. Technical report, Dept. of Computer Science, University of California at Santa Cruz, 1999.
6. H. Kashima, K. Tsuda, and A. Inokuchi. *Kernels for graphs*, chapter 7, pages 155–170. MIT Press, 2004.
7. P. Mahé and J.-P. Vert. Graph kernels based on tree patterns for molecules. *Machine Learning*, 75(1)(September 2008):3–35, 2009.
8. M. Neuhaus and H. Bunke. *Bridging the gap between graph edit distance and kernel machines*. World Scientific Pub Co Inc, 2007.
9. K. Zhang, R. Statman, and D. Shasha. On the editing distance between unordered labeled trees. *Information Processing Letters*, 42(3):133 – 139, 1992.