Noyau de Treelets Appliqué aux Graphes Étiquetés

Benoit Gaüzère, Luc Brun, Didier Villemin

Noyau de Treelets Appliqué aux Graphes Étiquetés.

B. Gaüzère[1] L. Brun[1] D. Villemin[2]

[1] GREYC, UMR 6072 CNRS, Caen, France
[2] LCMT, UMR 6507 CNRS, Caen, France

6 Boulevard Maréchal Juin
14 050 CAEN CEDEX
{benoit.gauzere,luc.brun,didier.villemin}@ensicaen.fr

Résumé

La chémoinformatique utilise des méthodes issues de l'informatique, plus particulièrement la théorie des graphes et l'apprentissage automatique, afin de classifier ou prédire les propriétés de bases de molécules. Dans ce contexte, les noyaux sur graphes fournissent une approche intéressante en combinant les méthodes d'apprentissage automatique et la représentation naturelle des molécules par graphes. Plusieurs méthodes basées sur les noyaux sur graphes ont été proposées pour résoudre des problèmes en chémoinformatique. La décomposition du graphe en sous structures représente une importante famille de noyau. Dans cet article, nous présentons une extension d'un noyau précédemment basé sur les sous structures non étiquetées à l'énumération de sous structures étiquetées. Nous proposons également deux méthodes itératives permettant de sélectionner un ensemble de sous structures afin d'améliorer la précision de la prédiction. Le noyau a été validé sur deux jeux de données impliquant des graphes étiquetés.

Mots Clef

chémoinformatique, noyau sur graphe, treelets, sacs de motifs.

Abstract

Chemoinformatics consists to discover or predict molecule's properties through informational techniques.Computer science's research fields mainly concerned by chemoinformatics are machine learning and graph theory. From this point of view, graph kernels provide a nice framework combining machine learning and graph theory techniques. Several methods based on graph kernels have been proposed to resolve chemoinformatics problems. An major family of kernels is based on a decomposition of a graph into substructures. In this paper, we present an extension of a kernel previously based on unlabeled sub structures to labeled substructures. We also propose two selection methods which allow us to reduce the set of considered sub structures in order to improve prediction accuracy. The proposed extension has been validated on two datasets of labeled graphs.

Keywords

chemoinformatics, graph kernel, treelets, bags of patterns.

1 Introduction

L'objet de la chémoinformatique est de prédire ou analyser les propriétés des molécules à l'aide de méthodes informatiques. Une des bases de ce domaine est le *principe de similarité* qui stipule que deux molécules ayant une structure similaire possèdent des activités et/ou des propriétés similaires. Une molécule peut être naturellement représentée par un graphe moléculaire $G = (V, E, \mu, \nu)$, où le graphe non étiqueté (V, E) code la structure de la molécule tandis que μ assigne à chaque noeud son élément chimique correspondant et ν encode le type de la liaison entre deux atomes. Une majorité de méthodes est basée sur la corrélation entre un ensemble de descripteurs associés à la molécule et une propriété à prédire. La liste de descripteurs peut être calculée à partir de la structure, des propriétés physiques ou bien encore de l'activité biologique de la molécule [17].
L'ensemble de ces descripteurs est regroupé au sein d'un vecteur de taille fixe. Cette dernière représentation permet d'appliquer à la chémoinformatique un vaste ensemble de méthodes numériques définies dans le cadre de l'analyse de données et des méthodes d'apprentissage. Cependant, la définition d'un vecteur à partir d'un choix de descripteurs implique une sélection a priori de l'information pertinente. De plus, pour certaines applications, la définition du vecteur de caractéristiques reste heuristique. Une seconde famille de méthodes, reposant sur la théorie des graphes, peut être décomposée en deux sous-familles. La première sous-famille [12], issue de l'analyse de données, consiste à trouver des sous-graphes ayant une importante différence de fréquence d'apparition entre deux ensembles d'exemples positifs et négatifs. La seconde sous-famille [1], reliée à l'apprentissage automatique, construit une description structurelle de chaque classe de molécules, de façon à ce que la classification soit effectuée par un appa-

riement structurel entre la molécule à traiter et chacun des prototypes. Cette famille est toutefois essentiellement restreinte aux problèmes de classification.
Les noyaux sur graphes peuvent être vus comme une mesure de similarité symétrique entre deux graphes. Si $\mathcal{G}$ est un espace de graphe, un noyau k de $\mathcal{G} \times \mathcal{G}$ dans $\mathbb{R}$ est dit semi défini positif si il vérifie la propriété suivante :

$$\forall\, G_1,\, \ldots,\, G_n, \forall\, (c_1, \ldots, c_n) \in \mathbb{R}^n,$$
$$\sum_i^n \sum_j^n c_i\, k(G_i, G_j)\, c_j \geq 0.$$

Pour tout noyau semi défini positif k, la valeur de $k(G, G')$, où G et G' désignent deux graphes, correspond à un produit scalaire entre deux vecteurs $\psi(G)$ et $\psi(G')$, la fonction $\psi(.)$ encodant une représentation des graphes dans l'espace de Hilbert $\mathcal{H}$ associé à k. L'astuce du noyau (*Kernel Trick*) permet d'utiliser la fonction noyau dans n'importe quel algorithme d'apprentissage automatique pouvant s'exprimer à l'aide de produits scalaires sans avoir à calculer explicitement la représentation des graphes dans $\mathcal{H}$. Les noyaux sur graphes et l'astuce noyau fournissent donc une connexion naturelle entre les approches structurelle et statistique de la reconnaissance de formes.
Une importante famille de méthodes à noyaux sur graphes est basée sur la construction d'un sac de sous structures pour chaque graphe, la similarité entre deux graphes étant déduite de la similarité entre leurs sacs. La plupart de ces méthodes sont basées sur des sous structures linéaires [7, 13, 18, 16]. Bien que l'utilisation de sous structures linéaires permet de limiter la complexité des algorithmes, elle ne permet de prendre en compte que très partiellement la topologie du voisinage de chaque sommet. D'autres méthodes [10, 14] définissent des noyaux basés sur un ensemble infini d'arbres au lieu de structures linéaires. Ces méthodes corrigent ainsi le manque d'expressivité des structures linéaires et, par conséquent, améliorent la pertinence de la mesure de similarité. Au lieu de décomposer les graphes en un ensemble infini de sous structures, le noyau peut être défini à partir de la distribution d'un ensemble prédéfini de sous structures non linéaires [15, 5]. Toutefois, ces méthodes n'utilisent pas l'étiquetage des noeuds et des arêtes, ce qui limite leur application aux jeux de données composés de graphes non étiquetés. Dans cet article, nous proposons d'étendre la méthode définie dans [5] aux sous structures étiquetées et donc d'étendre le domaine d'application de la méthode. Dans la Section 2, nous présentons un processus permettant d'obtenir une clé canonique identifiant l'étiquetage de chacune des structures énumérées par [5]. Cette clé canonique permet de distinguer deux structures ayant un même étiquetage. Dans la Section 3, nous proposons deux approches permettant de réduire le nombre de sous structures prises en compte dans le noyau afin d'améliorer le résultat de la prédiction. Ce noyau est ensuite comparé à diverses méthodes de l'état de l'art dans la Section 4 sur un problème de classification et un problème de régression. Ces deux expériences mettent en relief les avantages apportés par l'énumération de sous structures non linéaires ainsi que la pertinence de la sélection d'un sous ensemble de structures.

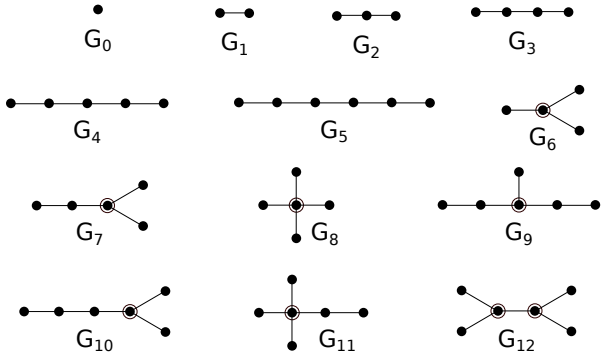


FIGURE 1 – Ensemble des arbres énumérés grâce à la méthode définie dans [5]. Les noeuds entourés représentent les *n-étoiles*.

2 Noyaux de treelets étiquetés

2.1 Énumération des treelets

La méthode décrite dans [5] permet d'énumérer et compter le nombre d'occurrences d'un ensemble fini de sous structures au sein d'un graphe. Les sous structures énumérées, appelées treelets, sont l'ensemble des arbres non étiquetés ayant un nombre de noeuds inférieur ou égal à 6, chaque noeud ayant au plus 4 voisins. Les arbres ayant un degré maximal supérieur à 4 ne sont pas énumérés car ils ne représentent pas une structure moléculaire faisable dans les jeux de molécules testés. L'ensemble des treelets est représenté dans la Figure 1. L'énumération de cet ensemble de sous graphes est principalement effectuée en deux étapes. La première étape consiste à énumérer les treelets linéaires en utilisant une recherche en profondeur. La seconde étape énumère l'ensemble des motifs non linéaires en analysant le voisinage des treelets spéciaux *3-étoile* ou *4-étoile*. Ces treelets correspondent à des arbres composés d'un noeud central de degré 3 ou 4 ainsi que leurs noeuds incidents. Dans le cas de graphes non étiquetés, un noyau peut être défini en confrontant la fréquence d'apparition de chaque treelet dans les deux graphes à comparer. Toutefois, lors de l'utilisation de cette méthode avec des graphes étiquetés, des treelets ayant un étiquetage différent peuvent être associés à une même structure. Afin de différencier ces treelets, il est possible d'utiliser une des méthodes d'étiquetage canonique de molécules [11, 8, 4]. Ces méthodes permettent d'identifier une molécule en se basant sur la structure et l'étiquetage du graphe moléculaire mais ne permettent pas de séparer clairement l'information structurelle extraite par [5] de l'information portée par l'étiquetage.
Dans cet article, nous proposons d'identifier chaque treelet par un code composé de deux parties : une première partie encodant l'information structurelle et définie par l'index de la structure du treelet (G_0, G_1, etc.) ainsi qu'une seconde partie encodant l'étiquetage du treelet. Cette seconde par-

tie est définie par une clé canonique correspondant à une suite d'étiquettes de noeuds et d'arêtes. Cette séquence est spécifique à chaque structure et est définie de manière à ce que deux treelets ayant un même code soient isomorphes.

La définition de la clé est triviale pour les structures linéaires, i.e. les chemins. Chaque chemin peut être associé à deux séquences composées alternativement des étiquettes de noeuds et d'arêtes, chacun encodant les deux parcours possibles du chemin. Par convention, la clé associée à une structure linéaire est définie comme la séquence ayant le plus faible ordre lexicographique.

Soit un treelet non linéaire $t = (V, E, \mu_t, \nu_t)$, où μ_t et ν_t correspondent respectivement aux fonctions d'étiquetage des noeuds et des arêtes. La clé canonique définie pour ces structures non linéaires est basée sur le concept de connectivité étendue, tel que défini dans [11]. Ce concept est basé sur une fonction d'étiquetage de noeuds λ de V sur $\mathbb{N}$, appelée étiquetage étendu. Cette fonction est définie par un processus itératif qui initialise chaque étiquette étendue $\lambda(v)$ par le degré de v. Cette étiquette initiale est ensuite étendue en assignant à chaque noeud la somme des étiquettes étendues λ de son voisinage. Ce processus est itéré tant que le nombre d'étiquettes étendues distinctes augmente. L'ensemble des étiquettes étendues obtenu est le même pour deux graphes isomorphes et est unique pour chaque structure d'arbre. La Figure 2 montre l'étiquetage étendu calculé sur les structures non linéaires. Cet étiquetage étendu correspond au degré des noeuds pour tous les treelets sauf G_9 où une itération de l'algorithme permet de différencier v_1 de v_3 et v_5.

Puisque deux noeuds adjacents v et v' d'un treelet peuvent être comparés suivant $\lambda(v)$ et $\lambda(v')$, l'étiquetage étendu définit un ordre partiel entre des noeuds adjacents au sein d'un même treelet. Ce tri partiel peut être représenté par un arbre enraciné. Les treelets G_6 à G_{11} ont un étiquetage étendu avec un seul maximum local et les arbres associés sont donc enracinés sur ce noeud (Figure 3(a,b)). Le treelet G_{12} possède deux maxima locaux situés sur les noeuds v_0 et v_3. Ce treelet est donc associé à deux arbres enracinés, chacun ayant pour racine v_0 ou v_3 (Figure 3(c-e)).

Notre processus de construction de la clé canonique d'un treelet est basé sur un parcours de l'arbre enraciné associé. La conception de la clé nécessite de trier les noeuds enfants de chaque noeud interne de l'arbre afin de définir un parcours unique de l'arbre et donc une clé unique. Cette étape de tri est effectuée par la récursion suivante : La clé de chaque feuille v, dénotée $clé(v)$, est définie par une étiquette vide. Pour chaque noeud v interne de l'arbre, considérons son ensemble de noeuds fils $\{v_1, \ldots, v_n\}$. Cet ensemble est premièrement trié selon $\lambda(v_i)$ et ensuite par la chaîne de caractère définie comme la concaténation de $\mu_t(v_i)$, $\nu_t(v, v_i)$ et $clé(v_i)$. Considérant ce tri sur $\{v_1, \ldots, v_n\}$, la clé associée au noeud v est définie par :

$$clé(v) = \left(\bigodot_{i=1}^{n} \mu_t(v_i).\nu_t(v, v_i) \right) . \bigodot_{i=1}^{n} clé(v_i) \quad (1)$$

où $\bigodot$ représente l'opérateur de concaténation. En utilisant cette récursion, l'étiquette de chaque noeud est encodée par la clé de son père. Afin de prendre en compte l'étiquette de la racine, la clé d'un arbre enraciné sur le noeud r est définie comme $\mu_t(r).clé(r)$.

Les treelets G_6 à G_{11} sont encodés par un seul arbre enraciné, et leurs codes canoniques sont définis comme l'index de la structure, concaténé avec la clé calculée à partir du parcours de l'arbre. Puisque le treelet G_{12} est associé à deux arbres enracinés et donc deux clés, son code canonique est défini comme l'index de la structure (G_{12}) concaténé avec la clé ayant le plus faible ordre lexicographique.

La clé de chaque treelet correspond à la suite d'étiquettes d'arêtes et de sommets rencontrés durant un parcours en profondeur effectué en suivant les index associés à chaque noeud par la méthode de Morgan [11]. Toutefois, à la différence de la représentation de Morgan, notre clé n'inclut pas d'information structurelle, qui est encodée par l'index de la structure associée au treelet.

Notre clé canonique est basée sur l'étiquetage étendu qui est lui même basé sur la structure du treelet ainsi que sur les fonctions d'étiquetage définies sur les noeuds et les arêtes. Par conséquent, deux treelets isomorphes sont associés à une même clé canonique. À l'inverse, puisque il est possible de construire un treelet linéaire à partir de sa clé canonique, deux treelets linéaires ayant la même clé doivent être isomorphes. Les treelets correspondant aux structures G_0 à G_5 peuvent donc être uniquement déterminés par leur clé canonique.

Notre processus de construction de la clé triant en priorité les noeuds en fonction de leur clé étendue, l'étiquette d'un noeud ayant une étiquette étendue unique sera située à une position fixe dans la clé associée à ce treelet. L'étiquette d'un tel noeud peut donc être retrouvée sans ambiguïté depuis la clé canonique. Toutefois, un ensemble de noeuds fils $\{v_1, \ldots, v_n\}$ possédant les mêmes étiquettes étendues et ayant le même noeud parent v seront triés selon l'ordre lexicographique de la suite d'étiquettes de noeuds et d'arêtes $\mu_t(v_i)\nu_t(v, v_i)clé(v_i)$. Ce tri permet d'obtenir une clé unique pour deux treelets isomorphes mais ne permet pas de différencier entre les permutations des noeuds $\{v_1, \ldots, v_n\}$. Il est donc nécessaire de vérifier, pour chaque treelet, que les permutations de noeuds autorisées par notre code correspondent à un treelet isomorphe.

Les noeuds ayant une même étiquette étendue dans les structures G_6, G_7, G_8, G_{10}, G_{11} et G_{12} sont représentées dans la Figure 2 par des carrés noirs (■). Pour chaque structure, ces noeuds de degré un sont les noeuds fils de l'unique noeud auxquels ils sont connectés. Par conséquent, notre clé ne différencie pas les permutations parmi ces noeuds. Puisque ces noeuds ont un degré égal à un et qu'ils sont connectés à un même noeud, n'importe quelle permutation échangeant deux de ces noeuds conduit à un treelet isomorphe.

L'arbre enraciné associé au treelet G_9 ne permet pas de différencier les deux branches v_2v_3 et v_4v_5 (Figure 3(b))

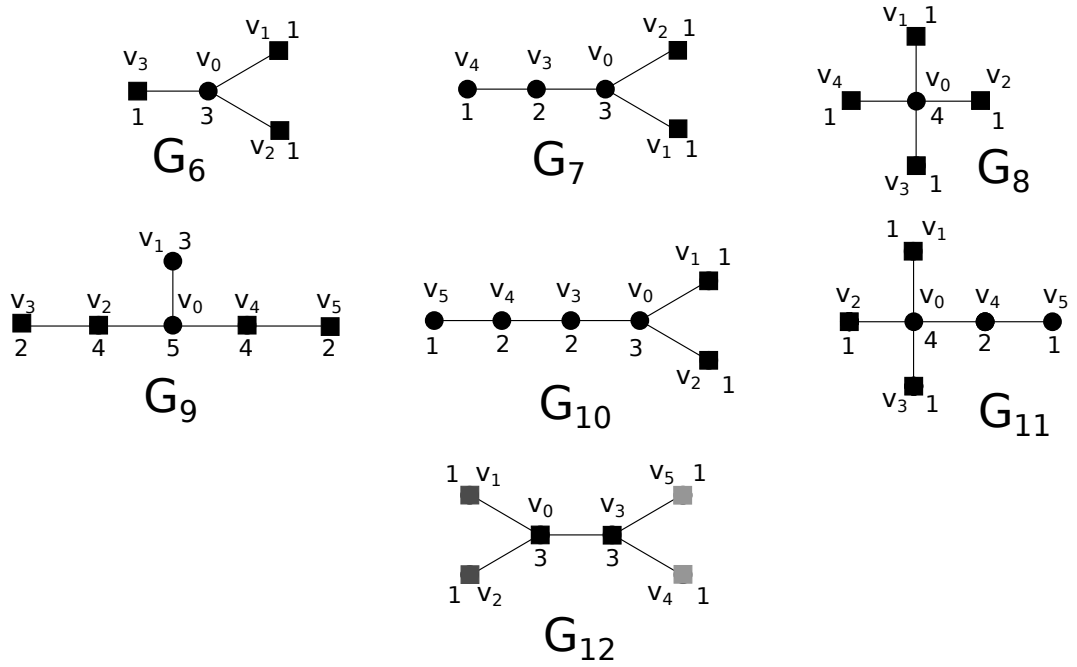


FIGURE 2 – Structures non linéaires avec l'étiquetage étendu de Morgan. La valeur de l'étiquette étendue est indiquée à proximité de chaque noeud. Les permutations possibles sont représentées par des noeuds en forme de carrés. Les différentes permutations possibles dans une même structure sont représentées par des niveaux de gris différents.

puisque v_2 et v_4 possèdent les mêmes étiquettes étendues. Toutefois, l'échange simultané de v_2 avec v_4 et v_3 avec v_5 donne un graphe isomorphe (Figure 3(a)). De la même manière, la clé canonique de G_{12} (Figure 3(c)) ne permet pas de différencier les permutations entre v_0 et v_3 ou bien entre v_1, v_2 d'une part et v_4, v_5 d'autre part. Toutefois, comme illustré dans la Figure 3(c), ces trois permutations conduisent à des treelets isomorphes.

Lorsque tous les treelets d'un graphe G ont été énumérés, un vecteur représentant la distribution des treelets dans G est calculé. Chaque élément de ce vecteur, dénoté le *spectrum* de G, est égal au nombre d'occurrences d'un treelet dans G :

$$f_i(G) = |(G_i \subset G)| \tag{2}$$

2.2 Définition du noyau de treelets

Une première idée pour définir un noyau à partir des treelets consiste à calculer le produit scalaire des deux vecteurs représentant le spectre des graphes. Cependant, ce type de noyau peut être biaisé par des treelets simples tels que G_0 qui sont présents un grand nombre de fois mais qui portent peu d'information. Nous utilisons donc un noyau Gaussien afin de mieux prendre en compte les différences entre chaque élément des deux spectres :

$$k_{Treelet}(G, G') = \sum_{k=0}^{T} e^{-\frac{(f_k(G)-f_k(G'))^2}{\sigma}} \tag{3}$$

où σ est une variable d'ajustement utilisée pour pondérer les différences entre le nombre d'occurrences de deux treelets et N est le nombre de treelets trouvés. Ce noyau peut donc être considéré comme une somme de noyaux Gaussiens entre deux vecteurs et est donc défini positif.

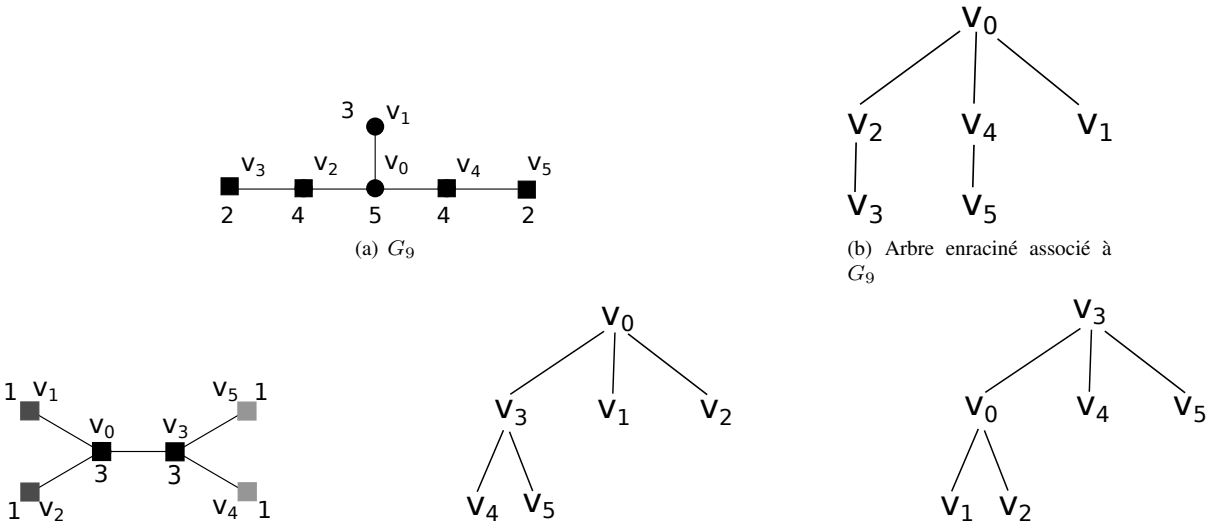
3 Sélection de treelets

Parmi l'ensemble des treelets trouvés dans un jeu de données, certains d'entre eux ne sont pas pertinents pour expliquer la propriété recherchée. La prise en compte de ces treelets entraîne des calculs superflus et dégrade le résultat de la prédiction. Une première approche pour éliminer ces treelets est de considérer tous les sous ensembles possibles de treelets. Une telle étude exhaustive implique de tester 2^p ensembles de treelets, avec p représentant le nombre de treelets trouvés dans un jeu de données. Une telle approche est impossible à réaliser, même en considérant un petit nombre de treelets.

Algorithme 1 Approche additive.

$P = Treelets$
$S = \varnothing$
$nb\_treelets = |P|$
for $i = 0 \rightarrow nb\_treelets$ **do**
  $t = \arg min_t RSS(S \cup t), t \in P$
  $P = P - t$
  $S_{i+1} = S_i \cup t$
end for
return $\arg min_{S_i} RSS(S_i), i \in [0, nb\_treelets]$

Pour définir un ensemble de treelets pertinents dans un problème de régression, nous proposons d'appliquer deux



(a) G_9

(b) Arbre enraciné associé à G_9

(c) G_{12}

(d) Premier arbre enraciné associé à G_{12}

(e) Second arbre enraciné associé à G_{12}

FIGURE 3 – Exemples de treelets non linéaires avec leur arbre enraciné associé. Les valeurs de λ sont indiquées à coté de chaque sommet sur les figures (a) et (c).

Algorithme 2 Approche Soustractive.

$S = Treelets$
$nb\_treelets = |S|$
for $i = 0 \rightarrow nb\_treelets$ **do**
  $t = \arg min_t RSS(S - t), t \in S$
  $S_{i+1} = S_i - t$
end for
return $\arg min_{S_i} RSS(S_i), i \in [0, nb\_treelets]$

approches itératives. La première, appelée approche additive [6], consiste à ajouter à chaque itération un nouveau treelet à l'ensemble des treelets utilisés pour calculer le noyau. Partant d'un ensemble de treelets vide, le treelet ajouté à chaque itération est celui qui donne le meilleur résultat de régression (Alg. 1). La qualité du résultat de la régression est évaluée en calculant la RSS (*Residual Sum of Squares*) définie comme la somme des carrés des erreurs de prédiction commises pour chaque molécule d'un jeu de test. La seconde approche [6] consiste à partir d'un ensemble composé de tous les treelets trouvés et de supprimer un treelet à chaque étape. Cette seconde approche est appelée approche soustractive (Alg. 2). Ces deux méthodes impliquent de tester $\frac{p(p+1)}{2}$ ensembles de treelets.

Une différence importante entre ces deux approches concerne les sous ensembles de treelets associés à une propriété donnée lorsqu'ils sont considérés simultanément. Pour de tels sous ensembles, la suppression d'un treelet par l'approche soustractive entraîne une forte augmentation de la RSS. Ces sous ensembles particuliers sont donc préservés par l'approche soustractive. À l'inverse, ces sous ensembles ont peu de chances d'être sélectionnés par l'approche additive puisque l'ajout de l'un de ces treelets n'améliorera pas de manière significative la RSS.

Méthode	Précision de la classification
(1) KMean [16]	80% (55/68)
(2) KWMean [3]	88% (60/68)
(3) Random Walks [18]	82% (56/68)
(4) Noyau Tree-Pattern [10]	96% (65/68)
(5) Noyau de treelets	94% (64/68)

TABLE 1 – Comparaison de différents noyaux sur graphes, combinés avec des SVM, sur un problème de classification.

4 Expérimentations

La première expérience évalue différents noyaux sur graphes sur un problème de classification. Ce problème est défini sur la monoamine oxidase (MAO) [1] et le jeu de données est composé de 68 molécules divisées en deux classes : 38 molécules inhibent la monoamine oxidase (médicament antidépresseur) et 30 ne l'inhibent pas. Ces molécules sont composées de différents éléments chimiques et sont donc représentées par des graphes étiquetés. La classification est effectuée par SVM en utilisant une validation croisée. Le Tableau 1 montre les résultats obtenus par les différentes méthodes à noyaux testées. Les trois premières lignes correspondent à des méthodes basées sur des motifs linéaires. La première ligne [16] déduit la similarité entre deux graphes de la similarité moyenne entre chaque paire de chemins extraits des graphes. La deuxième ligne [3]

1. Tous les jeux de données dans cette section sont disponibles sur la page Internet du TC15 : http ://www.greyc.ensicaen.fr/iapr-tc15/links.html#chemistry

Méthode	Erreur Moyenne ($°C$)	RMSE ($°C$)
(1) Réseaux de Neurones [2]	3.01	5.10
(2) KMean [16]	7.28	12.37
(3) Random Walks [18]	14.64	18.95
(4) Noyau Tree-Pattern [10]	5.53	9.50
(5) Noyau de treelet (TK)	4.90	7.80
(6) TK approche additive	3.23	4.36
(7) TK approche soustractive	2.63	3.70

TABLE 2 – Prédiction de la température d'ébullition sur des molécules acycliques. RMSE dénote la racine de l'erreur quadratique moyenne.

est une extension à la méthode précédente où l'importance des chemins non représentatifs est atténuée. La troisième ligne [18] est basée sur le nombre de marches aléatoires en commun dans deux graphes à comparer. Les deux dernières lignes sont des méthodes basées sur des motifs non linéaires. Le noyau Tree-Pattern (Ligne 4) calcule la similarité entre deux graphes à partir du nombre de *tree-patterns* que les deux graphes ont en commun. Enfin, la dernière ligne est le noyau défini dans la Section 2. Les noyaux sur graphes utilisant des motifs linéaires ne permettent pas de dépasser les 88% de bonne classification (Tableau 1, lignes 1 à 3). À l'inverse, les méthodes basées sur des motifs non linéaires permettent d'extraire plus d'information structurelle et obtiennent une meilleure précision de classification (Tableau 1, lignes 4 et 5). Cette expérience permet de mettre en relief l'importance de la prise en compte des motifs non linéaires lors de la comparaison de graphes moléculaires.

Le noyau défini dans cet article a été testé sur un second jeu de données composé de 185 molécules acycliques [2]. Le problème de régression proposé ici consiste à prédire la température d'ébullition des molécules. Les molécules présentes dans ce jeu de données ont la particularité d'être acycliques et composées d'hétéroatomes et sont donc représentées par des graphes acycliques étiquetés. Afin de pouvoir comparer les résultats obtenus, nous avons utilisé la procédure de validation croisée décrite dans [2] qui consiste à prédire la température d'ébullition d'une molécule en utilisant le reste du jeu de données comme base d'apprentissage.

La première ligne du Tableau 2 montre le résultat obtenu en utilisant un réseau de neurones basé sur le comptage de 20 sous structures définies par un expert en chimie. Les lignes suivantes correspondent à des méthodes utilisant des noyaux sur graphes et la température d'ébullition a été prédite en utilisant une *kernel ridge regression* [9]. Les lignes 2 et 3 du Tableau 2 correspondent à des méthodes basées sur des motifs linéaires. Ces deux méthodes obtiennent les erreurs de prédiction les plus élevées parmi les méthodes testées. En effet, la représentation du graphe moléculaire par des structures linéaires ne permet pas de prendre en compte assez d'information structurelle. Les méthodes basées sur des motifs non linéaires (Tableau 2, lignes 4 et 5) obtiennent de meilleurs résultats que les noyaux basés sur des motifs linéaires, mais sans atteindre la précision de la méthode basée sur les réseaux de neurones. Ce manque d'efficacité peut être expliqué par le nombre de treelets différents contenus dans le jeu de données. En effet, le noyau de treelets énumère 142 treelets différents dans ce jeu de données et certains d'entre eux apportent peu d'information vis à vis de la propriété à prédire. En conséquence, l'information apportée par ces treelets peut être assimilée à du bruit et dégrade la qualité du résultat. Le résultat de prédiction peut être amélioré en appliquant une des deux méthodes de sélection de treelets décrites dans la Section 3. L'utilisation de ces méthodes (cf. Tableau 2, lignes 6 et 7) permettent de réduire l'erreur de prédiction commise et obtiennent une plus faible $RMSE$ que celle obtenue par [2]. L'approche additive réduit l'ensemble des treelets utilisés à 26 treelets et l'approche soustractive réduit cet ensemble à 56 treelets. L'approche soustractive obtient les meilleurs résultats grâce au fait qu'elle prend mieux en compte l'information apportée par certaines combinaisons de treelets (Section 3). Une différence notable entre notre approche et celle proposée par Pierre Mahé [10] réside dans le fait que l'on calcule explicitement la distribution de chaque treelet dans une molécule. Cette énumération explicite permet de pondérer chaque treelet indépendamment tandis que la méthode décrite dans [10] permet de pondérer chaque sous structure seulement en fonction de sa profondeur ou du nombre de branchements.

5 Conclusion

Notre noyau est défini sur des structures non linéaires (les treelets) qui permettent de prendre en compte l'information structurelle présente dans les graphes. L'énumération explicite des treelets permet de pondérer chaque treelet de manière indépendante. Cette pondération permet de limiter l'impact des structures non associées avec la propriété à prédire. Les deux expériences ont montré les avantages apportés par la prise en compte de structures non linéaires ainsi que par le choix d'un sous ensemble de treelets pertinents.

Les futures extensions de ce noyau viseront à élargir l'ensemble de treelets et à inclure la présence de cycles. L'existence de cycles et les relations d'adjacence entre ces cycles induisent des propriétés spécifiques et doivent donc être détectés et encodés séparément du reste de la molécule. De plus, tous les cycles ne sont pas pertinents et le choix des cycles à énumérer est à étudier. Une autre extension importante est d'étendre les méthodes de sélection de treelets aux problèmes de classification.

Références

[1] L. Brun, D. Conte, P. Foggia, M. Vento, and D. Villemin. Symbolic learning vs. graph kernels : An experimental comparison in a chemical application. In

Proceedings of the 14th Conference on Advances in Databases and Information Systems (ADBIS 2010), pages 31–40, 2010.

[2] D. Cherqaoui, D. Villemin, A. Mesbah, J. M. Cense, and V. Kvasnicka. Use of a Neural Network to Determine the Normal Boiling Points of Acyclic Ethers, Peroxides, Acetals and their Sulfur Analogues. *J. Chem. Soc. Faraday Trans.*, 90 :2015–2019, 1994.

[3] F.-X. Dupé and L. Brun. Tree covering within a graph kernel framework for shape classification. In *Proceeding of 15 th International Conference on Image Analysis and Processing (ICIAP 2009)*, pages 278–287, 2009.

[4] J. L. Faulon, M. Collins, and R. D. Carr. The signature molecular descriptor. 4. Canonizing molecules using extended valence sequences. *Journal of chemical information and computer sciences*, 44(2) :427–36, 2004.

[5] B. Gaüzère, L. Brun, and D. Villemin. Two new graph kernels and applications to chemoinformatics. In *8th IAPR - TC-15 Workshop on Graph-based Representations in Pattern Recognition (GBR'11)*, volume 6658 of *Lecture Notes in Computer Science*, pages 112–121. Springer, 2011.

[6] R.R. Hocking. A Biometrics Invited Paper. The Analysis and Selection of Variables in Linear Regression. *Biometrics*, 32(1) :1–49, 1976.

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

[8] M. Kuramochi and G. Karypis. An efficient algorithm for discovering frequent subgraphs. *Knowledge and Data Engineering, IEEE Transactions on*, 16(9) :1038–1051, 2004.

[9] C. H. Lampert. *Kernel Methods in Computer Vision*. Now Publishers Inc., Hanover, MA, USA, 2009.

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

[11] H. L. Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *J. of Chem. Doc.*, 5(2) :107–113, 1965.

[12] G. Poezevara, B. Cuissart, and B. Crémilleux. Discovering emerging graph patterns from chemicals. In *Proceedings of the 18th International Symposium on Methodologies for Intelligent Systems (ISMIS 2009)*, pages 45–55, Prague, 2009. LNCS.

[13] L. Ralaivola, S. J Swamidass, H. Saigo, and P. Baldi. Graph kernels for chemical informatics. *Neural networks, special issue on Neural Networks and Kernel Methods for Structured Domains*, 18(8) :1093–1110, 2005.

[14] J. Ramon and T. Gärtner. Expressivity versus efficiency of graph kernels. In *1st Int. Workshop on Mining Graphs, Trees and Sequences*, pages 65–74, 2003.

[15] N. Shervashidze, S. V.N. Vishwanathan, T. H. Petri, K. Mehlhorn, and K. M. Borgwardt. Efficient graphlet kernels for large graph comparison. In *12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 488–495, 2009.

[16] F. Suard, A. Rakotomamonjy, and A. Bensrhair. Kernel on bag of paths for measuring similarity of shapes. In *European Symposium on Artificial Neural Networks*, pages 355–360, 2002.

[17] R. Todeschini and V. Consonni. *Handbook of Molecular Descriptors*. WILEY-VCH, Weinheim, 2000.

[18] S.V.N. Vishwanathan, K. M. Borgwardt, I. R. Kondor, and N. N. Schraudolph. Graph Kernels. *Journal of Machine Learning Research*, 11 :1201–1242, 2010.