Deux nouveaux noyaux sur graphes et leurs applications en chimioinformatique

Benoit Gaüzère, Luc Brun, Didier Villemin

Deux nouveaux noyaux sur graphes et leurs applications en chimioinformatique

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

[1] GREYC, UMR 6072 CNRS,
Caen, France
{benoit.gauzere,didier.villemin}@ensicaen.fr,

[2] LCMT, UMR 6072 CNRS,
Caen, France
luc.brun@greyc.ensicaen.fr

Résumé : La chimioinformatique utilise des méthodes issues de l'informatique, plus particulièrement la théorie des graphes et l'apprentissage automatique, afin de traiter d'importants volumes de données fournis par l'étude de familles de molécules. Dans ce contexte, les noyaux sur graphes fournissent une approche intéressante en combinant les méthodes d'apprentissage automatique avec la théorie des graphes. Précédemment, les noyaux sur graphes ont été appliqués avec succès sur quelques problèmes proposés par la chimioinformatique. Dans cet article, nous présentons deux nouveaux noyaux sur graphes appliqués sur des problèmes de régression et de classification. Le premier noyau est basé sur la notion de distance d'édition entre deux graphes tandis que le second se base sur l'énumération des sous arbres d'un graphe. La dernière partie montre la complémentarité des deux approches grâce à plusieurs expériences.

Mots-clés : apprentissage de graphes, méthodes à noyaux, distance d'édition

1. Introduction

Une des applications de la chimioinformatique consiste à prédire ou à analyser des propriétés moléculaires en utilisant des méthodes informatiques. Cette discipline se base sur le *principe de similarité* qui stipule que deux molécules structurellement similaires possèdent des activités et/ou des propriétés similaires. En chimioinformatique, une molécule peut être naturellement représentée par un graphe $G = (V, E, \mu, \nu)$, où le graphe non-étiqueté (V, E) code la structure de la molécule tandis que μ assigne à chaque sommet son

élément chimique correspondant et ν représente le type de la liaison entre deux atomes.

Une première famille de méthodes, la plus répandue en chimioinformatique, est basée sur la corrélation entre un ensemble de descripteurs associé à la molécule et une propriété particulière. 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 (Todeschini & Consonni, 2000). Ce vecteur est traité par des méthodes d'apprentissage afin de prédire la ou les propriétés recherchées. Cette approche s'appuie sur le vaste ensemble de méthodes d'apprentissage existant dans la littérature. Cependant, la définition d'un vecteur de descripteurs à partir d'une molécule, ie. un graphe, implique une perte d'information. 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 (Poezevara *et al.*, 2009), issue de l'analyse de données, consiste à trouver des sous-graphes ayant une importante différence de fréquence d'apparition entre les les deux ensembles d'exemples positifs et négatifs. La seconde sous-famille (Brun *et al.*, 2010), 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 appariement structurel entre chaque prototype et la représentation sous forme de graphe de la molécule à traiter. 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. En utilisant un noyau défini semi-positif, la valeur de $k(G, G')$, où G et G' désignent deux graphes, correspond à un produit scalaire entre deux vecteurs $\psi(G)$ et $\psi(G')$ dans un espace de Hilbert. Une grande famille de noyaux sur graphes est basée sur la construction d'un sac de sous-structures pour chaque graphe et déduit la similarité entre les deux graphes de la similarité entre les deux sacs. Dans (Kashima *et al.*, 2004), les noyaux sur graphes sont basés sur la comparaison d'ensembles infinis de marches extraites de chaque graphe. D'autres méthodes (Mahé & Vert, 2008) définissent des noyaux utilisant un ensemble infini d'arbres au lieu de marches. 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-graphes (Shervashidze *et al.*, 2009), appelés *graphlets*. Une

autre approche (Neuhaus & Bunke, 2007) vise à définir des noyaux définis positifs à partir de la distance d'édition entre deux graphes.

Dans cet article, deux nouveaux noyaux sur graphes sont présentés. Un premier noyau, présenté dans la Section 2., combine la distance d'édition et le Laplacien de graphe afin d'obtenir un noyau défini positif. Une méthode pour mettre à jour efficacement ce noyau est aussi proposée. Notre second noyau, présenté en Section 3., utilise une approche différente basée sur l'énumération explicite de tous les sous-arbres présents dans un graphe acyclique et non-étiqueté. L'efficacité et la complémentarité de ces deux noyaux sont finalement démontrés grâce aux expériences présentées dans la Section 4..

2. Noyau Basé sur la Distance d'Édition

Un chemin d'édition entre deux graphes G et G' est défini par une suite d'opérations (ajout, suppression ou ré étiquetage de sommet ou d'arête) permettant de transformer G en G'. Donné une fonction de coût $c(.)$ associée à chaque opération, le coût du chemin d'édition est défini comme la somme de tous les coûts de chaque opération élémentaire. Le coût minimal parmi tous les chemins d'édition transformant G en G' est appelé la *distance d'édition* entre les deux graphes. Une distance élevée indique une faible similarité entre les deux graphes alors qu'une faible distance indique une forte similarité. Selon (Neuhaus & Bunke, 2007), la complexité du calcul de la distance d'édition exacte augmente exponentiellement avec la taille des graphes traités, ce qui limite le calcul de la distance d'édition exacte aux graphes de petite taille. Pour pallier à ce problème, une méthode (Riesen & Bunke, 2009) permet de calculer une estimation sous-optimale de la distance d'édition exacte en $\mathcal{O}(()nv^2)$ où n et v sont respectivement égaux au nombre de sommets et au degré maximal des deux graphes. Malheureusement, la distance d'édition n'étant pas obligatoirement une métrique, les noyaux triviaux basés sur la distance d'édition ne sont pas toujours définis semi-positifs. Des méthodes (Neuhaus & Bunke, 2007) ont été proposées pour résoudre ce problème. Toutefois, les noyaux proposés ne sont pas explicitement basés sur le problème de minimisation traité par les méthodes à noyaux. Ce problème de minimisation peut être décrit comme ceci : étant donné un noyau k et un ensemble de graphes $D = \{G_1, \ldots, G_n\}$, la matrice de Gram K associée à D est une matrice $n \times n$ définie par $K_{i,j} = k(G_i, G_j)$. Dans le cadre des méthodes à noyaux, un problème de classification ou de régression basé sur K peut être

vu comme la minimisation de l'équation suivante :

$$f^* = \arg min_{f \in \mathbb{R}^n} CLoss(f, y, K) + f^t K^{-1} f \quad (1)$$

où $CLoss(.,.,.)$ désigne une fonction de coût représentant la distance entre le vecteur f et le vecteur des valeurs connues y. Comme indiqué dans (Steinke & Schölkopf, 2008), le terme $f^t K^{-1} f$ dans l'équation 1 peut être considéré comme un terme de régularisation qui contrebalance le terme d'attache aux données, représenté par la fonction $CLoss(.,.,.)$. Par conséquent, l'inverse de K peut être considéré comme un opérateur de régularisation. Inversement, l'inverse d'un opérateur de régularisation défini semi-positif peut être considéré comme un noyau. En suivant une méthode de conception de noyau récemment présentée (Brun *et al.*, 2010), nous construisons un opérateur de régularisation défini semi-positif sur l'ensemble des fonctions f assignant une valeur réelle à chaque graphe $\{G_1, \ldots, G_n\}$. L'inverse de cet opérateur définit un noyau sur l'ensemble $\{G_1, \ldots, G_n\}$. Pour construire cet opérateur de régularisation, considérons une matrice $n \times n$ d'adjacence W définie par $W_{ij} = e^{-\frac{d(G_i, G_j)}{\sigma}}$, où $d(.,.)$ représente la distance d'édition et σ est une variable d'ajustement. Le Laplacien de W est défini par $l = \Delta - W$ où Δ est une matrice diagonale définie par : $\Delta_{i,i} = \sum_{j=1}^{n} W_{i,j}$. La littérature sur la théorie spectrale des graphes (Chung, 1997) établit que l est une matrice symétrique définie semi-positive dont les valeurs propres sont positives, 0 ayant une multiplicité d'au moins 1. Cette dernière propriété rend impossible l'inversion de l. Pour résoudre ce problème, nous utilisons le Laplacien régularisé (Smola & Kondor, 2003) $\tilde{l}$ de W défini par $\tilde{l} = I + \lambda l$ où λ est un coefficient de régularisation. La valeur propre minimale de $\tilde{l}$ est égale à 1 et la matrice $\tilde{l}$ est donc définie positive. De plus, étant donné un vecteur f, nous avons :

$$f^t \tilde{l} f = \|f\|^2 + \lambda \sum_{i,j=1}^{n} W_{ij} (f_i - f_j)^2 \quad (2)$$

Intuitivement, minimiser l'équation 2 revient à construire un vecteur f avec une faible norme qui assigne les graphes ayant une faible distance d'édition (et donc un poids fort) à des valeurs proches. Cette contrainte correspond au terme de régularisation requis par l'équation 1 afin d'interpoler les valeurs y sur l'ensemble des graphes $\{G_1, \ldots, G_n\}$. Notre noyau non-normalisé est donc défini par : $K_{un} = \tilde{l}^{-1}$. Nous pouvons noter qu'un noyau Laplacien normalisé et régularisé peut aussi être considéré en définissant la matrice $\tilde{L} =$

$\Delta^{-\frac{1}{2}}\tilde{l}\Delta^{-\frac{1}{2}}$. Nous avons dans ce cas, pour un vecteur f :

$$f^t\tilde{L}f = \sum_{i=1}^{n}\frac{f_i^2}{\Delta_{ii}} + \lambda\sum_{j=1}^{n}\frac{W_{ij}}{\sqrt{\Delta_{ii}\Delta_{jj}}}(f_i - f_j)^2.$$

La matrice $\tilde{L}$ est définie positive et son noyau associé est défini comme $K_{norm} = \tilde{L}^{-1}$. Nous pouvons noter que notre noyau Laplacien normalisé et régularisé n'est pas défini comme l'inverse du Laplacien normalisé et régularisé $I + \lambda\Delta^{-\frac{1}{2}}l\Delta^{-\frac{1}{2}}$. Cette nouvelle expression est toutefois conforme à la contrainte de régularisation qui doit être ajoutée à l'équation 1 et fournit des avantages significatifs lorsqu'une nouvelle donnée doit être comparée (Section 2.1.).

2.1. Mise à jour de la matrice de Gram

Considérons tout d'abord un noyau défini par le Laplacien non normalisé. Étant donné notre base d'apprentissage $D = \{G_1, \ldots, G_n\}$, le test d'un nouveau graphe G nécessite la mise à jour du Laplacien non-normalisé l avec ce nouveau graphe ainsi que le calcul du noyau mis à jour, défini comme l'inverse du Laplacien régularisé et non-normalisé $K = (I + \lambda l)^{-1}$. Cette méthode triviale possède une complexité en $\mathcal{O}((n+1)^3)$, où n est le nombre de graphes considérés, ce qui la rend coûteuse si le nombre de graphes à traiter est élevé. Dans cette section, nous proposons une méthode visant à réduire la complexité de la mise à jour du noyau.

Étant donné le Laplacien non-normalisé et régularisé $\tilde{l}_n = (I_n + \lambda(\Delta_n - W_n))$ défini sur l'ensemble de graphes D, sa version mise à jour $\tilde{l}_{n+1}$ définie sur $D \cup \{G\}$ est définie par :

$$\tilde{l}_{n+1} = \begin{pmatrix} \tilde{l}_n - \delta_n & B \\ B^t & 1 - \sum_i B_i \end{pmatrix}.$$

où $B = (-\lambda \exp(\frac{-d(G,G_i)}{\sigma}))_{i=\{1,\ldots,n\}}$ est calculé à partir des poids entre le nouveau graphe G et chaque graphe $(G_i)_{i=\{1,\ldots,n\}}$ de la base d'apprentissage et δ_n est une matrice diagonale avec $(\delta_n)_{i,i} = B_i$. La matrice $\tilde{l}_{n+1}$ est inversible puisque sa valeur propre minimale est égale à 1 (Section 2.). Son inverse peut être calculé efficacement en utilisant une inversion par blocs :

$$K_{un} = (\tilde{l}_{n+1})^{-1} = \begin{pmatrix} \Gamma & \Theta \\ \Lambda & \Phi \end{pmatrix} \text{ avec } \begin{cases} \Gamma &= E^{-1} + \Phi E^{-1} B B^t E^{-1} \\ \Theta &= -E^{-1} B \Phi \\ \Lambda &= -\Phi B^t E^{-1} \\ \Phi &= (1 - \sum_i B_i - B^t E^{-1} B)^{-1} \end{cases} \tag{3}$$

où $E = \tilde{l}_n - \delta_n$. Le calcul de la mise à jour de notre noyau, en utilisant (3), revient à calculer l'inverse de la matrice $E = \tilde{l}_n + \delta_n$, ce qui peut être efficacement estimé en utilisant un développement à l'ordre K de $(I - \tilde{l}_n^{-1}\delta_n)^{-1}$:

$$(\tilde{l}_n - \delta_n)^{-1} = \tilde{l}_n^{-1}(I - \tilde{l}_n^{-1}\delta_n)^{-1} \approx \sum_{k=0}^{K} l_n^{-k-1}\delta_n^k. \tag{4}$$

Cette somme converge sous la condition $\|l_n^{-1}\delta_n\|_2 < 1$, pour $\lambda < 1$. En effet :

$$\|\tilde{l}_n^{-1}\delta_n\|_2 \leq \|\tilde{l}_n^{-1}\|_2 \|\delta_n\|_2 \leq \|\delta_n\|_2 \leq \lambda \max_{i=1,n} exp(\frac{-d(G, G_i)}{\sigma})$$

Le dernier terme de cette équation est strictement inférieur à 1, quel que soit λ strictement inférieur à 1. Nous pouvons de plus, montrer facilement que l'erreur d'estimation est inférieure à ϵ pour tout K supérieur à :

$$-\sigma \frac{log(2\epsilon)}{\min_{i=1,n} d(G, G_i)}. \tag{5}$$

L'équation 4 permet d'estimer l'inverse de $(\tilde{l}_n - \delta_n)$ par la somme des matrices pré-calculées l_n^{-k-1} multipliées par des matrices diagonales. En utilisant ce pré-calcul, l'inverse de $(\tilde{l}_n - \delta_n)$ et par conséquent le calcul du noyau mis à jour peut être effectué en KN^2.

Considérons à présent le Laplacien normalisé et régularisé (Section 2.) $\tilde{L} = \Delta^{-\frac{1}{2}}\tilde{l}\Delta^{-\frac{1}{2}}$, son inverse est égal à $\tilde{L}^{-1} = \Delta^{\frac{1}{2}}\tilde{l}^{-1}\Delta^{\frac{1}{2}}$ et nous obtenons :

$$K_{norm} = \Delta^{\frac{1}{2}} K_{un} \Delta^{\frac{1}{2}} \tag{6}$$

La mise à jour du Laplacien régularisé et normalisé peut donc être déduite de la mise à jour du Laplacien régularisé et non-normalisé.

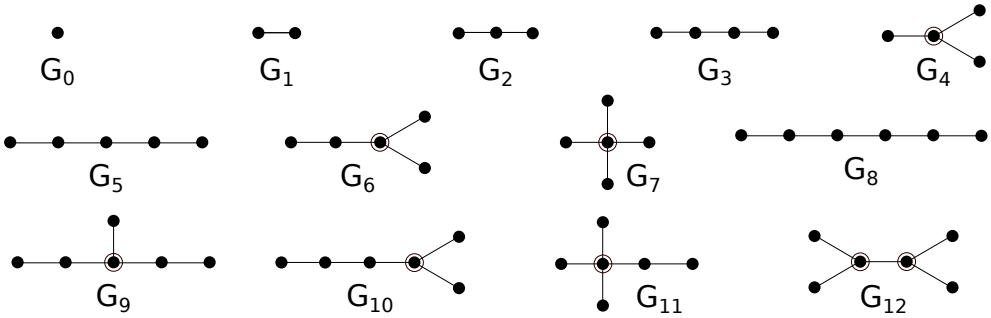


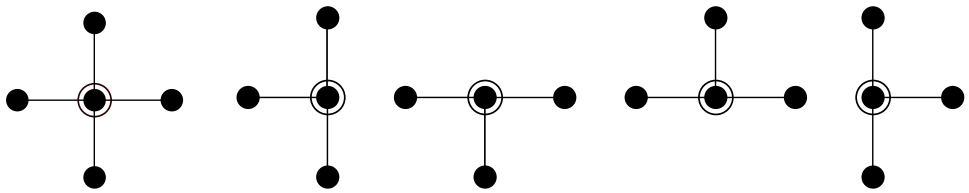
FIGURE 1: Ensemble des graphes acycliques et non-étiquetés avec un maximum de 6 sommets et un degré inférieur à 5. Les centres des *3-étoiles* et *4-étoiles* sont entourés.

3. Noyaux sur treelets

Une approche alternative aux noyaux basés sur la distance d'édition consiste à représenter chaque graphe par un sac de motifs. La similarité entre les deux graphes est ensuite déduite de la similarité entre les deux sacs de motifs. Comme mentionné dans la Section 1., la plupart de ces noyaux sont basés sur des motifs linéaires. À l'inverse, d'autres méthodes (Rogers & Hahn, 2010) ne prennent en compte que des motifs circulaires, sans traiter les structures linéaires imbriquées dans une structure plus complexe. Dans (Shervashidze *et al.*, 2009), une méthode pour énumérer, pour tout graphe non-étiqueté, tous ses sous-graphes connectés ayant jusqu'à 5 sommets est décrite. Nous proposons ici d'adapter cette méthode à l'énumération des sous-arbres composés au maximum de 6 sommets et ayant un degré inférieur ou égal à 4 dans un graphe acyclique et non-étiqueté. Les motifs résultants sont appelés treelets.

3.1. Calcul de la Distribution des treelets

En se basant sur la méthode décrite par (Shervashidze *et al.*, 2009), l'énumération des treelets débute par l'énumération de tous les chemins d'une longueur inférieure ou égale à 6. Un parcours en profondeur récursif avec une profondeur maximale de 6 à partir de chaque sommet permet de calculer la distribution des treelets G_0, G_1, G_2, G_3, G_5 et G_8 (Fig. 1). Nous pouvons remarquer que chaque chemin est retrouvé depuis ses deux extrémités et doit donc être compté $\frac{1}{2}$ fois à chaque découverte. Pour calculer la distribution des treelets restants, notre méthode repose sur la détection des sommets de



(a) G_7 (b) Décomposition de G_7 en G_4

FIGURE 2: G_7 contient 4 G_4.

TABLE 1: Conditions nécessaires pour énumérer les treelets basés sur une *n-étoile*. $N(v)$ et $d(v)$ représentent respectivement le voisinage et le degré du sommet v.

treelet	treelet source	Condition
G_6	*3-étoile*	$\lvert\{v; v \in N(R_{3-étoile}); d(v) \geq 2\}\rvert \geq 1$
G_9	*3-étoile*	$\lvert\{v; v \in N(R_{3-étoile}); d(v) \geq 2\}\rvert \geq 2$
G_{10}	*3-étoile*	$\exists v_0 \in N(R_{3-toile}); d(v_0) \geq 2$ et $\lvert\{v; v \in N(v_0) - \{R_{3-étoile}\}; d(v) \geq 2\}\rvert \geq 1$
G_{11}	*4-étoile*	$\lvert\{v; v \in N(R_{4-étoile}); d(v) \geq 2\}\rvert \geq 1$
G_{12}	*3-étoile*	$\lvert\{v; v \in N(R_{3-étoile}); d(v) \geq 3\}\rvert \geq 1$

degré 3 et 4. Ces sommets sont respectivement désignés $R_{3-étoile}$ et $R_{4-étoile}$ et sont les centres des treelets *3-étoile* et *4-étoile*. Nous pouvons noter que une *4-étoile* (G_7) contient quatre *3-étoiles* (Fig. 2). Cette première analyse des degrés de chaque sommet permet de calculer la distribution des treelets G_4 et G_7. Les treelets G_6, G_9,G_{10} et G_{12} sont énumérés à partir du voisinage des *3-étoiles*. Par exemple, G_6 nécessite une *3-étoile* avec au minimum un sommet périphérique possédant un degré supérieur ou égal à 2. Les propriétés définissant les treelets basés sur une *3-étoile* ou sur une *4-étoile* sont récapitulées dans le Tableau 1. Nous pouvons remarquer que G_{12} est symétrique car il contient deux $R_{3-étoile}$ et doit donc être compté $\frac{1}{2}$ fois car il sera détecté à partir des deux $R_{3-étoile}$. Les conditions présentées dans le Tableau 1 définissent les conditions requises pour l'existence d'un treelet centré autour d'une *n-étoile*. Toutefois, ces conditions ne garantissent pas l'unicité du treelet. Un exemple est illustré dans la Fig. 3 : le sommet le plus à droite possède un degré égal à 4 alors qu'un degré supérieur ou égal à 2 est nécessaire pour construire un treelet G_9. Trois différents G_9 sont donc construits à partir des cinq même sommets. Nous constatons facilement qu'aucun isomorphisme n'existe entre

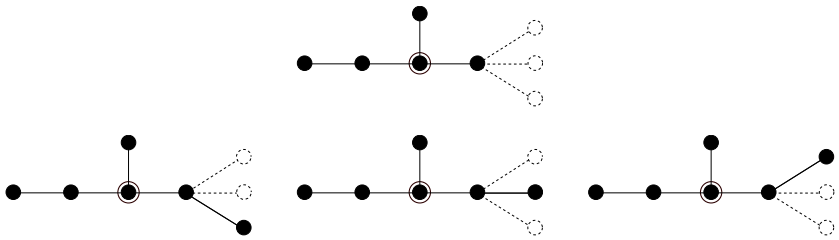


FIGURE 3: Trois permutations de G_9 partageant la même base commune.

les treelets décrits dans la Fig. 1. De plus, nous pouvons montrer facilement (Cayley, 1875) que le nombre de graphes acycliques, non-étiquetés, de taille inférieure ou égale à 6 et ayant un degré maximal égal à 4 est égal à 13. Ces graphes sont représentés dans la Fig. 1.

Lorsque tous les treelets d'un graphe G ont été énumérés, un vecteur, appelé *spectrum* de G, représentant la distribution des treelets est calculé. Chaque élément de ce vecteur est égal à la fréquence d'un treelet dans G : $f_i(G) = |(G_i \subset G)|$.

3.2. Définition du noyau treelet

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 très présents mais portent peu d'information. Nous utilisons donc un noyau Gaussien afin de mieux représenter les différences entre les deux spectres : $k_{Treelet}(G, G') = \sum_{k=0}^{N} e^{-\frac{(f_k(G)-f_k(G'))^2}{\sigma}}$ où σ est une variable d'ajustement utilisée pour pondérer les différences entre les fréquences de deux treelets et N est le nombre de treelets à énumérer. Ce noyau est considéré comme un noyau Gaussien entre deux vecteurs et est donc défini positif.

4. Expérimentations

La première expérience est effectuée sur un jeu de données d'inhibiteurs de la monoamine oxidase (MAO) [1] qui est composé de 68 molécules divisées en

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

TABLE 2: Comparaison des méthodes sur un problème de classification.

Méthode	Précision de la classification
KWMean (Dupé & Brun, 2009)	88% (60/68)
Noyau basé sur la Distance d'Édition (Neuhaus & Bunke, 2007)	90% (61/68)
Noyau Laplacien Standard (Eq. 6)	90% (61/68)
Noyau Laplacien Rapide (Eq. 6)	90% (61/68)
Marches Aléatoires (Vishwanathan *et al.*, 2010)	82% (56/68)

TABLE 3: Estimation de la température d'ébullition d'alcanes en C°.

Méthode	Erreur Moyenne	Écart Type	Corrélation
Réseaux de Neurones	3.11453	3.69993	0.9827
Noyau Laplacien	10.7948	16.4484	0.9412
Noyau treelet	1.40663	1.91695	0.9992

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 précision de la classification est mesurée en utilisant une validation croisée sur un SVM. Le Tableau 2 montre les résultats obtenus par le noyau Laplacien utilisant la distance d'édition sous-optimale (Riesen & Bunke, 2009). Le noyau Laplacien obtient une précision de 90%, ce qui correspond au score le plus élevé. Nous pouvons remarquer que l'autre méthode obtenant 90% de bonne classification est aussi basée sur la distance d'édition. Ce dernier noyau peut toutefois ne pas être défini semi-positif. Nous pouvons aussi noter que l'utilisation de la méthode rapide (Section 2.1.) n'altère pas la précision du résultat de classification (Tableau 2, lignes 4 et 5). Un nombre maximum de 9 itérations est nécessaire pour assurer un ϵ inférieur ou égal à 10^{-4} (Équation 5), ce qui permet de réaliser l'inversion en $\mathcal{O}(9N^2)$ au lieu de $\mathcal{O}(N^3)$ si la méthode triviale était utilisée. Cette réduction de complexité permet de diminuer le temps de calcul par un facteur de $1,8$ (soit $0,273$ ms au lieu $0,498$ ms en moyenne pour une inversion). Le noyau treelet n'a pas été testé sur ce jeu de données car il est conçu pour les graphes non-étiquetés.

La deuxième expérience est effectuée sur un jeu de données composé d'alcanes. Un alcane est une molécule acyclique composée de carbones et d'hy-

drogènes, ce qui permet de la représenter par un graphe acyclique et non-étiqueté, les atomes d'hydrogène étant implicitement représentés. Nous pouvons noter que, dû aux propriétés chimiques du carbone, le degré des graphes est inférieur ou égal à 4. Le jeu de données utilisé dans (Cherqaoui & Villemin, 1994) est composé de 150 alcanes, chacun étant associé à une température d'ébullition. En utilisant le même protocole de tests que celui utilisé dans (Cherqaoui & Villemin, 1994), nous avons estimé le point d'ébullition de chaque alcane en utilisant 90% des molécules comme base d'apprentissage et les 10% restants comme données à prédire. Le Tableau 3 montre les résultats obtenus par différentes méthodes. Les mauvais résultats obtenus par le noyau Laplacien peuvent s'expliquer par la faible information disponible dans des graphes non-étiquetés. En effet, pour des graphes non étiquetés, l'heuristique utilisée pour estimer la distance d'édition (Riesen & Bunke, 2009) assigne l'ensemble des sommets des deux graphes en utilisant pour seule information le degré des sommets. Cette méthode considère donc deux assignements comme équivalents si des sommets avec un même degré existent dans les deux graphes. Dans ce cas, l'utilisation de la distance d'édition sous-optimale implique une mauvaise comparaison des graphes. D'autre part, notre noyau surpasse les précédents résultats obtenus dans (Cherqaoui & Villemin, 1994) et basés sur les réseaux de neurones combinés avec des descripteurs chimiques.

5. Conclusion

Dans cet article, nous avons proposé un noyau Laplacien basé sur une distance d'édition sous-optimale et combiné avec une mise à jour efficace du noyau afin de prédire les propriétés des nouvelles données. Les expériences montrent l'efficacité de ce noyau sur des jeux de données de molécules complexes composées de plusieurs hétéroatomes. Toutefois, ce noyau n'obtient pas de bons résultats sur des graphes non-étiquetés. Nous proposons donc un nouveau noyau basé sur l'énumération exhaustive des treelets dans un graphe acyclique et non-étiqueté. Ce noyau obtient de meilleurs résultats que ceux obtenus par d'anciennes méthodes, mais reste restreint aux graphes non-étiquetés. Nos futurs travaux seront consacrés à supprimer cette limitation en adaptant ce noyau aux graphes étiquetés.

Références

BRUN L., CONTE D., FOGGIA P., VENTO M. & VILLEMIN D. (2010).

Symbolic learning vs. graph kernels : An experimental comparison in a chemical application. *Proc. of the 1st Int. Workshop on Querying Graph Structured Data*.

CAYLEY A. (1875). On the analytic forms called trees, with applications to the theory of chemical combinations. *Reports British Assoc. Adv. Sci.*, **9**, 427–460.

CHERQAOUI D. & VILLEMIN D. (1994). Use of a neural network to determine the boiling point of alkanes. *J. Chem. Soc. Faraday Trans.*, **90**, 97–102.

CHUNG F. (1997). *Spectral graph theory*. AMSP.

DUPÉ F.-X. & BRUN L. (2009). Tree covering within a graph kernel framework for shape classification. In *ICIAP*, p. 278–287.

KASHIMA H., TSUDA K. & INOKUCHI A. (2004). *Kernels for graphs*, In *Kernel Methods in Computational Biology*, chapter 7, p. 155–170. MIT Press.

MAHÉ P. & VERT J.-P. (2008). Graph kernels based on tree patterns for molecules. *Machine Learning*, **75**(1), 3–35.

NEUHAUS M. & BUNKE H. (2007). *Bridging the gap between graph edit distance and kernel machines*. World Scientific Pub Co Inc.

POEZEVARA G., CUISSART B. & CRÉMILLEUX B. (2009). Discovering emerging graph patterns from chemicals. In *Proc. of the 18th ISMIS 2009*, p. 45–55, Prague : LNCS.

RIESEN K. & BUNKE H. (2009). Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing*, **27**(7), 950–959.

ROGERS D. & HAHN M. (2010). Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.*, **50**(5), 742–754.

SHERVASHIDZE N., VISHWANATHAN S. V., PETRI T. H., MEHLHORN K. & BORGWARDT K. M. (2009). Efficient graphlet kernels for large graph comparison. In *Proceedings of AIStats*, p. 488–495.

SMOLA A. & KONDOR R. (2003). Kernels and regularization on graphs. In *Learning theory and Kernel machines : 16th Annual Conference on Learning Theory and 7th Kernel Workshop*, p. 144 : Springer Verlag.

STEINKE F. & SCHÖLKOPF B. (2008). Kernels, regularization and differential equations. *Pattern Recogn.*, **41**, 3271–3286.

TODESCHINI R. & CONSONNI V. (2000). *Handbook of Molecular Descriptors*. Weinheim : WILEY-VCH.

VISHWANATHAN S., BORGWARDT K. M., KONDOR I. R. & SCHRAUDOLPH N. N. (2010). Graph kernels. *Journal of Machine Learning Research*, **11**, 1201–1242.