Graphes prototypes vs. graphe médian généralisé pour la classification de données structurées

Romain Raveaux, Eugen Barbu, Sébastien Adam, Pierre Héroux, Éric Trupin

Graphes Prototypes *vs*. Graphe Médian Généralisé pour la Classification de Données Structurées

Romain Raveaux[1] – Eugen Barbu[2] – Sébastien Adam[3] – Pierre Héroux[3] – Eric Trupin[3]

[1] L3I – Université de La Rochelle
Avenue Michel Crépeau, 17042 La Rochelle cédex 1, FRANCE
romain.raveaux01@univ-lr.fr

[2] ACM Professional Member

[3] Université de Rouen, LITIS EA 4108 BP 12 - 76801 Saint-Etienne du Rouvray, FRANCE
Prenom.Nom@univ-rouen.fr

Résumé : *Dans [BAR 06], nous avons présenté une méthode de classification de données structurelles basée sur l'utilisation de graphes prototypes extraits par un algorithme génétique. Dans ce nouvel article, nous comparons d'un point de vue théorique et d'un point de vue expérimental cette approche à l'utilisation des graphes médians généralisés. Nous montrons que la modification du critère de choix des prototypes pour tenir compte de la distribution inter-classe des données, ainsi que la possibilité d'extraire plusieurs représentants par classe permettent d'améliorer de façon significative les performances de classification.*

Mots-clés : Classification supervisée de graphes, Graphes prototypes, graphes médians généralisés, algorithmes génétiques.

1 Introduction

Cet article aborde la problématique de classification supervisée de graphes relationnels attribués. Dans de nombreuses applications, il est en effet nécessaire d'affecter une classe à un graphe inconnu, après une phase d'apprentissage s'appuyant sur un ensemble de graphes dont la classe est connue.

Dans [BAR 06], nous avons proposé une approche de classification de graphes s'appuyant sur (i) une mesure de dissimilarité appelée signature de graphes [LOP 03] et sur (ii) l'utilisation d'un algorithme génétique pour générer un ensemble de graphes prototypes constituant la base d'apprentissage d'un classifieur de type K plus proches voisins.

Cet article présente la suite de ces travaux. Nous y comparons du point de vue théorique et du point de vue expérimental les graphes prototypes proposés dans [BAR 06] aux travaux actuels de la littérature concernant les Graphes Médians d'Ensemble (GME) et les Graphes Médians Généralisés (GMG) [JIA 01, FER 07]. Notre objectif est de montrer que les graphes prototypes constituent une extension intéressante des graphes médians dans le contexte particulier de la classification supervisée. Cette extension concerne 2 aspects : (i) pour chacune des classes de graphes, plusieurs graphes prototypes peuvent être générés alors qu'un seul représentant est extrait dans le cas des graphes médians (GM), (ii) le critère utilisé pour construire l'ensemble de graphes prototypes n'est pas la somme des distances aux graphes de la base comme dans le cas des GM mais le taux de bonne classification obtenu lors d'une phase de classification par kppv sur un ensemble de test. C'est ainsi le critère final de l'application, prenant en compte non seulement la distribution intra-classe des graphes, mais aussi la distribution inter-classe, qui est utilisé pour générer les prototypes.

Plusieurs tests sur différentes bases de données synthétiques et réelles sont présentés dans la suite de cet article. Les résultats obtenus tendent à montrer que dans un contexte de classification supervisée, les graphes prototypes que nous extrayons permettent l'obtention de meilleures performances que l'utilisation des GME et des GMG.

La suite de l'article est organisée de la façon suivante. Dans la section 2, le concept de graphe médian généralisé est rappelé, puis comparé d'un point de vue formel au concept de graphes prototypes. La section 3 rappelle ensuite la méthode proposée pour générer les graphes prototypes. La mesure de dissimilarité ainsi que l'algorithme d'apprentissage y sont présentés. La section 4 décrit ensuite les expérimentations menées et les résultats obtenus. Enfin, une discussion conclut cet article dans la section 5.

2 Définition et notations

Définition 1 *Un graphe relationnel attribué dirigé est un 4-uplet $G = (V, E, \mu, \xi)$ où :*

- *V est l'ensemble des nœuds,*
- *$E \subset V \times V$ est l'ensemble des arcs,*
- *$\mu : V \rightarrow L_V$ est une fonction affectant une étiquette aux nœuds*
- *$\xi : E \rightarrow L_E$ est une fonction affectant une étiquette aux arcs.*

Le problème de classification supervisée de graphes peut être défini comme suit

Définition 2 *Soit un ensemble d'apprentissage de graphes $T = \{\langle g_i, c_i \rangle\}_{i=1}^{L}$, où $g_i \in \chi$ est un graphe étiqueté et $c_i \in C = \{C_n\}_{k=1}^{N}$ est la classe du graphe parmi les N classes présentes dans les L éléments de T. L'apprentissage d'un classifieur de graphes consiste alors à apprendre à partir de*

T une fonction $f(g) : \chi \rightarrow C$ permettant d'affecter une classe à un graphe inconnu.

La littérature propose principalement deux types d'approches pour résoudre un problème de classification supervisée de graphes : les approches à base de noyaux [KAS 03, KAS 04, SUA 05, MAH 05] et des approches de type K Plus Proches Voisins (KPPV). Cette dernière est la plus fréquemment adoptée pour sa simplicité de mise en œuvre, et ses bonnes performances. Elle souffre toutefois de trois défauts majeurs que sont sa complexité calculatoire, sa complexité mémoire, et sa sensibilité aux exemples bruités. Pour diminuer ces défauts, une solution naturelle consiste à « réduire » la base d'apprentissage de graphes, en sélectionnant ou en générant des représentants à partir de l'ensemble des éléments de la base initiale. Ces techniques sont particulièrement adaptées lorsque le calcul des dissimilarités requiert un nombre de calculs important. Parmi les techniques existantes d'extraction de représentants de classe, la seule approche ayant à notre connaissance été adaptée aux représentations structurelles consiste à utiliser le graphe médian.

Si on dispose d'une distance ou d'une mesure de similarité d permettant de comparer deux graphes, on peut alors définir le Graphe Médian d'Ensemble (GME) d'un ensemble de graphes :

Définition 3 *Soit $S = \{g_1, g_2, \ldots, g_n\}$ un ensemble de graphes respectant la définition 1. Le Graphe Médian de l'Ensemble S (GME) est défini par $\widehat{g}$ tel que :*

$$\widehat{g} = \arg\min_{g \in S} \sum_{i=1}^{n} d(g, g_i) \quad (1)$$

$\widehat{g}$ est alors un élément de l'ensemble S.

Cette définition a été étendue à la notion de Graphe Médian Généralisé (GMG) n'appartenant pas nécessairement à S.

Définition 4 *Soit U l'ensemble de tous les graphes pouvant être construits en utilisant les ensembles L_V et L_E. Soit $S \subset U$. On peut alors définir le GMG de l'ensemble S par $\overline{g}$ tel que :*

$$\overline{g} = \arg\min_{g \in U} \sum_{i=1}^{n} d(g, g_i) \quad (2)$$

De tels graphes médians, qu'ils soient ou non généralisés peuvent être utilisés comme représentants de classe dans le cadre d'un processus de classification [FER 06]. Toutefois, le processus est alors modélisant puisqu'il ne tient pas compte des interactions entre classes. Dans l'approche que nous proposons, nous préférons une approche discriminante, dans laquelle les représentants de classes sont extraits en tenant compte de toutes les classes. Par ailleurs, dans l'approche que nous proposons, ce sont les performances finales que permettent d'obtenir les graphes prototypes qui sont utilisées comme critère.

Dans le cas de l'utilisation d'un kppv comme algorithme de décision, on cherche alors à définir un ensemble contenant 1 représentant (issu de U) de chacune des classes tel que :

$$\begin{aligned} \widetilde{G_1} &= \{\widetilde{g_1}, \widetilde{g_2}, ..., \widetilde{g_N}\} \\ &= \arg\min_{\{g_i\}_{i=1}^{N} \in U} errKppv(T, \{g_i\}_{i=1}^{N}) \end{aligned}$$

où $errKppv\left(T, \{g_i\}_{i=1}^{N}\right)$ désigne ici l'erreur de classification commise par un classifieur de type Kppv sur T en utilisant $\{g_i\}_{i=1}^{N}$ en tant que base d'apprentissage.

Cette notion peut être très simplement étendue à l'utilisation de M prototypes par classe en calculant :

$$\begin{aligned} \widetilde{G_M} &= \{\widetilde{g_{11}}, \ldots, \widetilde{g_{1M}}, \widetilde{g_{21}}, \ldots, \widetilde{g_{2M}}, \ldots, \widetilde{g_{N1}}, \ldots, \widetilde{g_{NM}}\} \\ &= \arg\min_{\{g_{im}\}_{i=1,m=1}^{N,M} \in U} errKppv\left(T, \{g_{im}\}_{i=1,m=1}^{N,M}\right) \end{aligned}$$

Dans la section suivante, nous montrons comment nous avons utilisé les algorithmes génétiques pour résoudre les différents problèmes de sélection de représentants décrits sous forme de problème d'optimisation dans cette section.

3 Génération des médians et des prototypes

Dans la section précédente, nous avons montré que l'extraction de représentants de graphes, qu'ils soient des graphes médians ou des graphes prototypes pouvait s'exprimer comme un problème d'optimisation. Dans cette section, nous proposons d'utiliser un algorithme génétique pour résoudre ces problèmes. Les Algorithmes Génétiques (AG) sont des algorithmes d'optimisation heuristiques adaptatifs basés sur les hypothèses évolutionnaires de la sélection naturelle et de la génétique. Les AGs réalisent une exploration aléatoire guidée par les objectifs de l'espace des solutions possibles d'un problème donné. Pour ce faire, ils procèdent de façon itérative à la génération de populations d'individus par (i) l'application d'opérateurs génétiques et (ii) l'évaluation de ces populations grâce à une fonction d'adaptation que le processus itératif vise à optimiser.

Dans les applications proposées ici, qu'il s'agisse de l'extraction de graphes médians ou de génération graphes prototypes, un individu code un ensemble de représentants par classe (un représentant par classe dans le cas des graphes médians, et M dans le cas des graphes prototypes). L'objectif quant à lui diffère en fonction de l'approche. Dans le cas des graphes médians généralisés, il s'agit de minimiser, indépendamment dans chacune des classes, la distance du médian à tous les éléments. Dans le cas des graphes prototypes, il s'agit cette fois de minimiser l'erreur de classification produite par un classifieur de type « 1 plus proche voisin » sur une base de test lorsque l'ensemble des prototypes codés par l'individu est utilisé comme base d'apprentissage.

Dans les deux cas, la complexité de la mesure de dissimilarité utilisée est un paramètre critique. Or, les méthodes permettant le calcul de distances entre deux graphes (distance d'édition, distances basées sur le plus grand sous-graphe commun) sont réputées pour avoir des complexités non polynomiales dans le pire des cas les rendant inexploitables dans un processus génétique. Afin de contourner ce problème,

nous utilisons une mesure de dissimilarité entre graphes basée sur une signature vectorielle. Cette mesure introduite par Lopresti et Wilfong [LOP 03] est calculée en temps linéaire.

3.1 Signature de graphe

Étant donné un graphe non-orienté et non étiqueté $G^{no} = (V, E)$ où V représente l'ensemble des sommets et E représente l'ensemble des arêtes, la signature $S_{1a}(G^{no})$ est définie comme un vecteur dont la composante i dénombre les sommets de degré i.

$$S_{1a}(G^{no}) \equiv (n_0, n_1, n_2, \ldots) \tag{3}$$

avec $n_i = |\{v \in V | deg(v) = i\}|$

Étant donnés deux graphes non orientés et non étiquetés G_1^{no} et G_2^{no}, il est alors possible de définir $d^{no}(G_1^{no}, G_2^{no})$ comme étant la norme L_1 entre les signatures de G_1^{no} et G_2^{no}.

$$d^{no}(G_1^{no}, G_2^{no}) \equiv S_{1a}(G_1^{no}) - S_{1a}(G_2^{no}) \tag{4}$$

Dans le cas des graphes orientés non-étiquetés, la signature $S_{1b}(G^o)$ proposée par Lopresti et Wilfong dénombre les sommets de G^o disposant de i arcs entrants et j arcs sortants.

$$S_{1b}(G^o) \equiv (n_{0,0}, n_{0,1}, n_{1,0}, ldots) \tag{5}$$

avec $n_{i,j} = |\{v \in V | indeg(v) = i, outdeg(v) = j\}|$

La mesure de dissimilarité d^o entre deux graphes orientés non étiquetés G_1^o et G_2^o est alors définie comme la norme L_1 entre les signatures $S_{1b}(G_1^o)$ et $S_{1b}(G_2^o)$.

$$d^o(G_1^o, G_2^o) \equiv S_{1b}(G_1^o) - S_{1b}(G_2^o) \tag{6}$$

En généralisant le modèle de graphe, il est alors possible de considérer que les nœuds et les arcs du graphe peuvent être étiquetés par une valeur nominale. La structure d'arcs d'un nœud doit alors être adaptée pour prendre en compte l'étiquetage des arcs entrants et sortants. Ainsi, si les arcs du graphes prennent leurs étiquettes dans un ensemble $\{l_1, \ldots, l_\alpha\}$, la structure d'arcs d'un nœud v est un 2α-tuple $(x_1, \ldots, x_\alpha, y_1, \ldots, y_\alpha)$, où x_i représente le nombre d'arcs entrant dans v dont l'étiquette est l_i et où y_j représente le nombre d'arcs sortant de v dont l'étiquette est l_j. La signature $S_{1c}(G^e)$ dénombre les nœuds de G^e disposant d'une structure d'arcs donnée, et ce pour toute les structures d'arcs possibles. Une signature $S_2(G^e)$ s'attache à décrire la distribution des étiquettes $\{l_1, \ldots, l_\beta\}$ des nœuds. Ainsi, $S_2(G^e)$ peut être définie comme un vecteur dont la i^{e} composante correspond au nombre de nœuds de G^e étiquetés l_i.

$$S_2(G^e) \equiv (m_1, m_2, \ldots, m_\beta) \tag{7}$$

avec $m_i = |\{v \in V | lab(v) = l_i\}|$

Au final, dans le cas le plus général des graphes orientés et étiquetés, il est possible de définir une mesure de dissimilarité d^e entre deux graphes G_1^e et G_2^e comme étant la somme des normes L_1 entre les signatures S_{1c} et S_2.

$$d^e(G_1^e, G_2^e) \equiv S_{1c}(G_1^e) - S_{1c}(G_2^e) v + S_2(G_1^e) - S_2(G_2^e) \tag{8}$$

Lopresti et Wilfong énoncent plusieurs propriétés concernant la distance entre signatures de graphes. Les différentes mesures de dissimilarité présentées précédemment sont des distances entre les signatures vectorielles décrivant les graphes. En revanche, ces mesures sont des pseudo-métriques entre les graphes. En effet, ces mesures respectent les propriétés de symétrie, de non-négativité et l'inégalité triangulaire, mais ne respectent pas l'unicité. En effet, deux graphes différents peuvent avoir des signatures identiques. En revanche, ces différentes pseudo-métriques peuvent être mises en relation avec une véritable distance d'édition ed par la relation suivante.

$$d^{no}(G_1^e, G_2^e) \leq d^o(G_1^e, G_2^e) \leq d^e(G_1^e, G_2^e) \leq 4.ed(G_1^e, G_2^e)$$

3.2 Algorithme génétique

3.2.1 Principe de l'algorithme

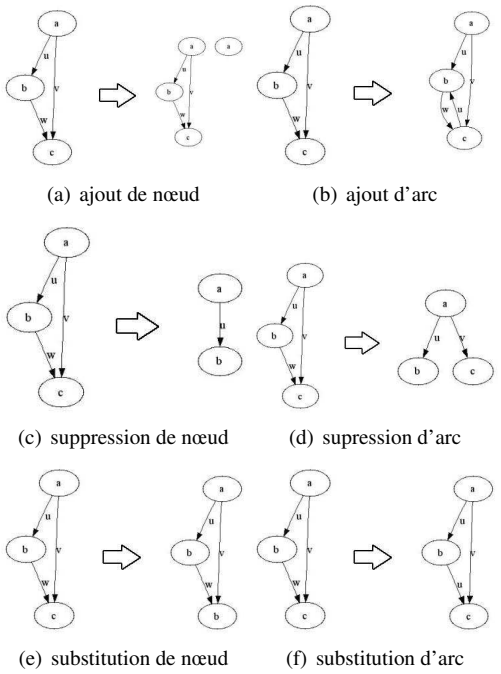
Comme évoqué précédemment, nous utilisons un même algorithme génétique pour résoudre les deux problèmes d'optimisation que sont la génération de graphes médians généralisés ou de graphes prototypes. Dans les deux cas, le même codage des individus est utilisé. En revanche, si l'opérateur génétique de mutation est commun aux deux problèmes, la fonction d'adaptation et l'opérateur de croisement sont adaptés au cas d'usage.

Algorithme 1 Algorithme génétique

ENTRÉE : T : base d'apprentissage du problème initial de classification
ENTRÉE : M : nombre de représentants par classe à générer
ENTRÉE : $taillePopulation$
ENTRÉE : $nombreGeneration$
ENTRÉE : $tauxMutation$
ENTRÉE : μ : nombre de meilleurs[1] individus à conserver
SORTIE : $listePrototypes$: liste de M prototypes par classe
  tantque la population initiale est incomplète **faire**
    $ind \leftarrow$ tirage aléatoire de M éléments par classe dans T
    calculer l'adaptation de ind
    insérer ind dans la population initiale
  fin tantque
  pour $i = 1$ à $nombreGeneration$ **faire**
    les μ meilleurs individus[1] sont conservés dans la nouvelle population
    tantque la nouvelle population est incomplète **faire**
      $op \leftarrow$ tirage aléatoire entre mutation ou croisement[2]
      si $op =$ mutation **alors**
        $ind \leftarrow$ tirage aléatoire d'un individu dans la population précédente[3]
        $nouvelInd \leftarrow$ mutation(ind)
        calculer l'adaptation de $nouvelInd$
        insérer $nouvelInd$ dans la nouvelle population
      sinon
        $(ind_1, ind_2) \leftarrow$ tirage aléatoire de deux individus dans la population précédente[3]
        $(nouvelInd_1, nouvelInd_2) \leftarrow$ croisement(ind_1, ind_2)
        calculer l'adaptation de $nouvelInd_1$ et de $nouvelInd_2$
        insérer $nouvelInd_1$ et $nouvelInd_2$ dans la nouvelle population
      finsi
    fin tantque
  fin pour
  renvoyer le meilleur individu[1] de la dernière génération

L'algorithme que nous utilisons est donné par l'Alg. 1. Les μ meilleurs individus d'une population sont reportés sur la génération suivante afin d'assurer de ne pas dégrader le meilleur individu d'une génération à l'autre.

[1] au sens de la fonction d'adaptation
[2] en fontion de $tauxMutation$
[3] en fonction de leur adaptation



(a) ajout de nœud (b) ajout d'arc

(c) suppression de nœud (d) supression d'arc

(e) substitution de nœud (f) substitution d'arc

FIG. 1 – Mutation

3.2.2 Codage des individus et opérateurs génétiques

Le contexte dans lequel nous travaillons nous impose d'adapter le codage des individus et les opérateurs génétiques au cas des graphes. Nous rappelons qu'un individu code une solution potentielle du problème soit 1 ou M graphes représentants par classe pour chacune des N classes.

L'exécution de l'opérateur de mutation correspond à l'exécution aléatoire d'une des opérations d'édition applicables à un graphe représentée sur la Fig. 1, à savoir l'ajout, la suppression et la substitution d'un nœud ou d'un arc. Pour ces opérations, le nœud, l'arc ou les étiquettes concernés sont tirés aléatoirement avec une probabilité uniforme.

L'opération de croisement vise à permettre l'échange de matériel génétique entre deux solutions potentielles du problème à optimiser. En l'espèce, il s'agit donc d'échanger un certain nombre de graphes représentants entre deux individus, soit d'échanger une partie de ces graphes.

Dans l'application de génération du graphe médian généralisé, chaque classe n'est représentée que par un seul représentant. L'opération de croisement consiste alors à « croiser » les graphes représentants de chacun des individus, et ce, pour chaque classe. Pour ce faire, les deux graphes à « croiser » sont chacun séparés aléatoirement en deux sous-graphes. Les arcs liant des nœuds appartenant à deux sous-graphes différents sont appelés arcs externes. Les sous-graphes des deux représentants sont alors permutés en recombinant les arcs externes (cf. Fig. 2)

Dans l'application de génération de graphes prototypes, l'opération de croisement est adaptée pour obtenir une convergence plus rapide. Chaque individu encode M représentants par classe. Lors de l'évaluation de l'individu, chaque représentant est affecté d'un score fonction du nombre d'éléments qu'il a bien classés. Parmi les M représentants d'une classe les $M/2$ meilleurs sont appelés bons représentants et

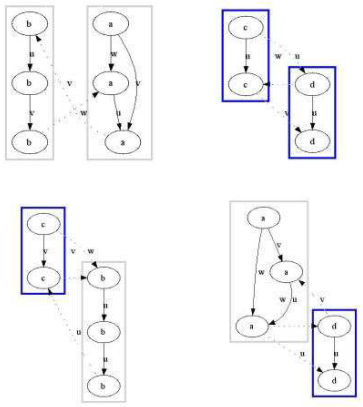


FIG. 2 – Croisement de graphes

les autres mauvais représentants. Lors du croisement de deux individus de I_1 et I_2, il y a échange des bons représentants de I_2 avec les mauvais représentants de I_1. I_1 et I_2 partagent donc la moitié de leurs gènes en concentrant les meilleurs représentants sur un individu.

3.2.3 Fonction d'adaptation

Si les opérateurs génétiques peuvent être adaptés pour orienter efficacement l'exploration de l'espace de solutions, la fonction d'adaptation est essentielle et définit le problème d'optimisation.

Dans le cas de la génération du GMG, un individu sera d'autant mieux évalué que la somme des distances entre les représentants et les graphes de leur classe sera faible.

Dans le cas de la génération de graphes prototypes, un individu correspond à la base d'apprentissage d'un algorithme de plus proche voisin. Son évaluation est directement fonction du taux de bonne classification obtenu sur une base de test.

4 Expérimentation et résultats

4.1 Données utilisées

Pour évaluer des algorithmes de classification de graphes, nous sommes confrontés à la difficulté de trouver des bases étiquetées mises à disposition et reconnues. Pour surmonter ce problème, nous avons choisi de mener nos tests sur trois bases de données que nous présentons ici. L'une est composée de données synthétiques permettant d'évaluer l'algorithme de classification de graphe dans un cadre général, indépendamment de leur utilisation. Les deux autres bases, quant à elles, s'attachent au cadre applicatif de la reconnaissance de symboles. Les caractéristiques de ces trois bases sont synthétisées dans le tableau 1.

4.1.1 Graphes synthétiques : la base A

Cette base est constituée de 28 000 graphes uniformément distribués en 100 classes. Il s'agit de graphes orientés étiquetés. Les étiquettes de noeuds et d'arcs proviennent de deux ensembles distincts. Le cadre général permettant la production de graphes aléatoires proposé par Erdös et Rényi [ERD 59] n'a pas pour objectif de décrire des classes de

	Base A	Base B	Base C
Nombre de classes	100	10	32
|Apprentissage|	14128	114	9600
|Validation|	14101	56	3200
Nombre moyen de nœuds par graphe	12.03	5.56	8.84
Nombre moyen d'arcs par graphe	9.86	11.71	10.15
Degré moyen des nœuds	1.63	4.21	1.15

TAB. 1 – Caractéristiques des bases de données

graphes similaires. Pour ce faire, nous proposons un processus en deux étapes, pour la création de classes de graphes. Dans une première étape, N graphes aléatoires sont construits en utilisant le modèle Erdös-Rényi [ERD 59], N étant le nombre de classes à générer. Les entrées du modèle Erdös-Rényi sont le nombre de noeuds du graphe et la probabilité d'établissement d'un arc entre deux noeuds. Dans une seconde étape, les graphes sont modifiés par suppression ou ré-étiquetage de noeuds ou d'arcs. Les graphes ainsi obtenus subissent une seconde modification, en remplaçant des noeuds choisis aléatoirement par un sous-graphe aléatoire. Ces modifications génèrent des graphes présentant une similarité plus importante en intra-classe qu'en inter-classes.

4.1.2 Reconnaissance de symboles : la base B

Les données de cette base sont obtenues par extraction d'une représentation sous forme de graphe d'un corpus de 180 images de symbole bruitées à partir de 10 images modèles idéales provenant du concours de reconnaissance de symboles GREC [VAL 04]. Dans cette représentation, les étiquettes nominales des nœuds du graphe dérivent la forme des composantes connexes du symbole et les arcs représentent la topologie de ces composantes connexes en terme de relation de voisinage. Nous renvoyons le lecteur à [BAR 06] pour davantage de détails sur la construction de la représentation structurelle des symboles.

4.1.3 La base de Ferrer

Dans [FER 06], Ferrer propose d'extraire une représentation structurelle sur une collection de symboles graphiques, 12 800 images réparties en 32 classes. Ces images de symboles sont à une échelle unique et n'ont pas subi de rotation, ils proviennent de la base GREC [VAL 04]. Ces symboles sont composés essentiellement de segments se terminant soit par un point de jonction (PJ) soit par un point terminal (PT).

Dans le but de prouver la robustesse des prototypes contre le bruit, 4 niveaux de distorsion ont été introduits. Les distorsions sont engendrées en déplaçant chaque PJ ou PT aléatoirement à l'intérieur d'un cercle de rayon r, donné comme paramètre pour chaque niveau de bruit. Si un PJ est modifié alors tous les segments qui lui sont connectés sont également déplacés pour assurer la connexité. Pour chaque classe et pour chaque niveau de bruit 100 images sont générées. Ainsi pour chaque classe nous avons 400 éléments. Finalement, pour les 32 classes de notre problème, 12 800 images sont créées (32×400).

Dans son application, Ferrer est contraint par la théorie spectrale des graphes d'utiliser des graphes étiquetés par des

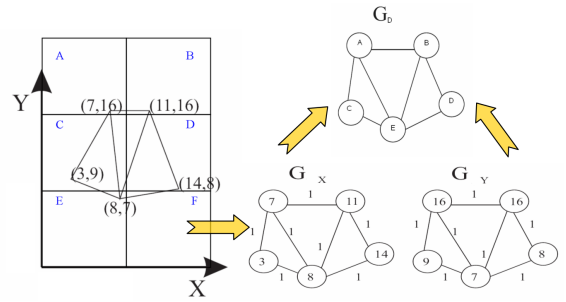


FIG. 3 – Du symbole au graphe

réels positifs ou nuls. De ce fait, il construit pour chaque symbole un graphe pour chacune des coordonnées. Ainsi les nœuds de G_x (respectivement G_y) représentant les PTs ou les PJs sont étiquetés avec la coordonnée en x (respectivement y). Les arcs correspondent aux segments entre les points.

De notre côté, nous sommes contraints par l'usage de la signature de graphe à n'utiliser que des étiquettes nominales. Pour ce faire, nous procédons à une discrétisation des coordonnées des PTs et PJs par le biais d'un maillage (cf. Fig. 3). Dans les expérimentations décrites par la suite, nous avons utilisé un maillage 4×4.

4.2 Protocole expérimental et résultats

Une première expérience vise à comparer les performances en classification de graphes obtenues en choisissant un représentant par classe par rapport à un algorithme $kppv$ classique. Dans cette expérience, nous comparons sur les bases A, B et C les performances en classification par plus proche voisin lorsque le représentant de la classe est :

- le Graphe Médian d'Ensemble (GME) ;
- un Graphe Médian Généralisé (GMG) obtenu par algorithme génétique ;
- un Graphe Prototype (GP) obtenu par algorithme génétique ;

L'algorithme génétique est paramétré avec un taux de mutation égal à 0.1 (choisi suite à une expérimentation), une population de 200 individus et 100 générations.

Les taux de classification donnés dans le tableau 2 amènent à plusieurs remarques. On remarque bien évidemment que l'usage de représentants dégrade les taux obtenus par ppv, mais les complexités temporelle et mémoire s'en trouvent réduits de façon drastique puisqu'il n'y plus que N calculs de distance[1]. Sur l'ensemble des bases les résultats obtenus à l'aide du graphe médian généralisé sont supérieurs à ceux issus de l'utilisation du graphe médian d'ensemble. Cela valide le fait que le GMG généré par algorithme génétique dispose d'un meilleur pouvoir modélisant de la classe que le graphe médian d'ensemble. Par ailleurs, les résultats obtenus par utilisation des graphes prototypes surclassent ceux correspondant aux GMG. Cette observation valide l'hypothèse selon laquelle un processus discriminant offre des performances supérieures à celles découlant d'un processus modélisant.

[1] N est le nombre de classes

	1ppv	GME	GMG	GP
Base A	100.0%	94.3%	96.3%	97.8%
Base B	96.4%	62.5%	64.3%	78.6%
Base C	99.7%	76.4%	83.8%	88.4%

TAB. 2 – Comparaison des taux de classification en fonction de la nature du représentant de la classe

	Base		
M	A	B	C
1	97.8	78.6	88.4
2	99.1	85.7	91.0
3	99.7	91.1	92.0
4	99.7	91.1	94.5
5	99.7	92.8	95.0

TAB. 3 – Taux de classification en fonction du nombre de prototypes par classe

Dans un second temps, nous avons souhaité étudier l'influence du nombre M de prototypes par classe sur le taux de classification. Les résultats donnés dans le tableau 3 montrent clairement que le taux de classification croît avec le nombre de représentants. Cela indique que l'augmentation du nombre de prototypes tend d'une part à mieux décrire la difficulté du problème de classification. En effet, on constate que c'est sur le problème *a priori* le plus difficile (base B) que le paramètre M a le plus fort impact.

5 Conclusion et perspectives

Dans cet article, nous avons présenté et comparé plusieurs approches de classification de graphes par recherche de plus proche voisin dans le cas où les classes sont représentées par des graphes médians ou des graphes prototypes.

Les expérimentations ont été menées sur plusieurs bases synthétiques et réelles dont le volume montre que ces approches peuvent être appliquées à ces masses de données structurelles.

Les résultats indiquent que le graphe médian généralisé que nous approchons grâce à un algorithme d'optimisation génétique dispose d'un meilleur pouvoir modélisant que le graphe médian d'ensemble [JIA 01].

Ces mêmes résultats illustrent également que le processus de classification discriminant par graphes prototypes prenant en considération l'ensemble des classes offre de meilleures performances que le processus modélisant offert par les graphes médians.

Enfin, nous avons proposé la possibilité de synthétiser M représentants par classe et montré que les performances augmentent avec ce paramètre. Le nombre de représentants tend à une meilleure description du problème de classification. Dans ce sens, un parallèle peut être établi avec l'augmentation du nombre vecteurs supports d'un classifieur à vaste marge.

Nous envisageons à court terme de poursuivre l'étude concernant l'influence du paramètre M afin d'examiner la convergence des performances de l'approche par représentant vers celles du ppv.

Nous envisageons également d'étudier le compromis entre performances en reconnaissance et la réduction de la base d'apprentissage. Une optimisation multi-objectifs de ces critères concurrents autoriserait le choix du paramètre M *a posteriori* de manière optimale et suivant le cas d'usage. Il peut éventuellement être envisagé une valeur de M différente selon les classes.

Enfin, à la différence des approches modélisantes, les approches discriminantes n'offrent pas naturellement de processus de rejet pourtant souvent nécessaire dans les applications réelles. Dans ce cadre, il sera nécessaire d'utiliser un nouvel algorithme d'optimisation capable d'appréhender plusieurs objectifs.

Références

[BAR 06] BARBU E., RAVEAUX R., LOCTEAU H., ADAM S., HÉROUX P., , TRUPIN E., Classification de graphes par algorithmes génétiques et signatures de graphes. Application à la reconnaissance de symboles, *Colloque International Francophone sur l'Ecrit et le Document*, 2006, pp. 91-96.

[ERD 59] ERDÖS P., RÉNYI A., On random graphs, *Publicationes Mathematicae Debrecen*, vol. 6, 1959, pp. 290-297.

[FER 06] FERRER M., VALVENY E., , SERRATOSA F., Spectral Median Graphs Applied to Graphical Symbol Recognition, *CIARP*, 2006, pp. 774-783.

[FER 07] FERRER M., SERRATOSA F., , VALVENY E., On the Relation Between the Median and the Maximum Common Subgraph of a Set of Graphs, *GbRPR*, 2007, pp. 351-360.

[JIA 01] JIANG X., MÜNGER A., , BUNKE H., On Median Graphs : Properties, Algorithms, and Applications., *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, n° 10, 2001, pp. 1144-1151.

[KAS 03] KASHIMA H., TSUDA K., , INOKUCHI A., Marginalized kernels between labeled graphs, *Proceedings of the Twentieth International Conference on Machine Learning*, 2003, pp. 321-328.

[KAS 04] KASHIMA H., TSUDA K., , INOKUCHI A., Kernel for graph, *Kernel Methods in Computational Biology*, 2004, pp. 155-170.

[LOP 03] LOPRESTI D. P., WILFONG G. T., A fast technique for comparing graph representations with applications to performance evaluation, *IJDAR*, vol. 6, n° 4, 2003, pp. 219-229.

[MAH 05] MAHÉ P., UEDA N., AKUTSU T., PERRET J.-L., , VERT J.-P., Graph kernels for molecular structure-activity relationship analysis with support vector machines, *Journal of Chemical Information and Modeling*, vol. 45, n° 4, 2005, pp. 939-951.

[SUA 05] SUARD F., GUIGUE V., RAKOTOMAMONJY A., , BENSRHAIR A., Pedestrian detection using stereovision and graph kernels, *Proceedings of the IEEE Intelligent Vehicle Sysposium*, 2005, pp. 267-272.

[VAL 04] VALVENY E., DOSCH P., Symbol recognition contest : a synthesis, LLADÓS J., KWON Y. B., Eds., *Selected Papers of the 5th IAPR International Workshop on Graphics Recognition*, vol. 3088 de *Lecture Notes in Computer Science*, pp. 368-385, Springer Verlag, 2004.