Mesure de similarité de graphes par noyau de sacs de chemins

Frédéric Suard,Alain Rakotomamonjy

LITIS EA 4051, INSA/Université de Rouen,
avenue de l'université,
76800 Saint Etienne du Rouvray, FRANCE
frederic.suard@insa-rouen.fr

Résumé − La classification de graphes s'appuie généralement sur une mesure de similarité entre graphes utilisée ensuite par un classifieur. Nous proposons ici l'utilisation des méthodes à noyaux pour une application de reconnaissance de formes représentées à l'aide de graphes. Nous introduisons tout d'abord la notion de noyau de graphe que nous étendons en proposant des noyaux entre des sacs de chemins. Nos premiers résultats montrent l'intérêt de cette approche par rapport aux approches classiques de comparaison de graphes.

Abstract − A common approach for classifying shock graphs is to use a dissimilarity measure on graphs and a distance based classifier. In this paper, we propose the use of kernel functions for data mining problems on shock graphs. The first contribution of the paper is to extend the class of graph kernel by proposing kernels based on bag of paths. Then, we propose a methodology for using these kernels for shock graphs retrieval. Our experimental results show that our approach is very competitive compared to graph matching approaches and is rather robust.

1 Introduction

La reconnaissance de formes à l'aide d'images a fait l'objet de nombreux travaux lors de ces dernières années. Parmi les nombreuses problématiques impliquées dans ce processus, une des principales consiste à extraire des caractéristiques les plus pertinentes possibles afin d'extraire les données les plus représentatives à partir des signaux .

Les caractéristiques sont généralement extraites à partir des données brutes du signal. Cette information reste donc fidèle aux données mais relativement de bas niveau. L'idée consiste alors à structurer les données entre elles afin d'extraire une information de plus haut niveau. Une manière de répondre à cette alternative repose sur l'utilisation de graphes. Un graphe permet en effet de structurer différentes composantes (les nœuds) à l'aide d'arcs en apportant une information supplémentaire à propos de la structures des composantes. Dans le cas de la reconnaissance d'objets, les graphes sont une représentation compacte et structurée des formes des objets.

De nombreuses approches proposent ainsi d'utiliser des graphes pour la reconnaissance d'objets [8, 2, 1]. La problématique consiste alors à comparer les graphes entre eux. Des solutions existent, comme par exemple la mesure *edit-distance* ou la recherche du plus grand sous graphe commun. Nous proposons d'élargir la notion de comparaison de graphes à l'aide d'un noyau de graphe, c'est à dire un produit scalaire entre graphes. La définition d'un noyau de graphe permet également de bénéficier de la théorie sousjacente des méthodes à noyau.

Le calcul d'un noyau de graphe est composé de deux étapes principales. Tout d'abord, les graphes sont décomposés en des ensembles de chemins puis le noyau final est obtenu en comparant ces ensembles de chemins. La comparaison de deux structures est ainsi ramenée à la comparaison d'ensembles des composantes de ces structures.

Après avoir redéfini briévement la construction des ensembles de chemins (section 2), nous montrerons que les formulations de noyau de graphes peuvent s'écrire comme des noyaux sur des ensembles (section 3). Enfin, nous appliquerons finalement cette approche de noyai de graphe comme mesure de similarité pour la reconnaissance d'objets (section 4).

2 Sac de chemins

Nous allons tout d'abord définir la représentation par graphe et la notion de sac de chemins.

2.1 Représentation par graphe

Soit N un ensemble fini de nœuds et A un ensemble fini d'arcs tel que $A \in N \times N$. Un graphe G est ainsi défini par un ensemble de nœuds et d'arcs tel que $G = (N, A)$.

Lors de ces travaux, nous avons suivi la méthode proposée par Diruberto et al. [6] afin d'extraire des graphes à partir d'images d'objets dont nous connaissons leur masque binaire, c'est à dire la région exacte occupée par l'objet dans l'image. A partir de cette image binaire, nous extrayons le squelette morphologique qui est ensuite transformé en graphe (voir figure 1). Les nœuds du graphe sont les intersections ou les terminaisons des branches du squelette. Le graphe est obtenu en reliant les nœuds à l'aide

des branches du squelette.

Comme le graphe n'est qu'une structure, il convient d'apporter de l'information caractéristique de la forme représentée par l'intermédiaire d'étiquettes. Le graphe est donc étiqueté en apportant diverses informations concernant la topologie de l'objet relatives à la position des nœuds, l'orientation et la longueur des arcs, la consistance de l'objet aux emplacement des nœuds ou des arcs. Une fonction ℓ est utilisée pour attribuer des étiquettes aux nœuds et arcs du graphe G.

2.2 Constitution d'un sac de chemin

Un chemin h peut être défini comme une succession de p nœuds adjacents appartenant au graphe G : $h = \{n_1, n_2, ..., n_p\}$. Il s'agit donc d'un sous graphe de G.

Les chemins peuvent être générés en parcourant aléatoirement les graphes comme le propose Kashima et al. [5] ou bien en considérant les chemins les plus courts entre chaque nœud à l'aide d'un algorithme de type Dijkstra afin d'éviter la redondance d'information et les phénomènes d'aller et retour dans le graphe.

Un graphe peut alors être décomposé en un ensemble de chemins $P = \{h_1, h_2, ...\}$. Nous conservons ainsi un ensemble de chemins qui contiennent une partie de l'information structurelle du graphe, tout en étant moins complexe. Nous pouvons ainsi décomposer le graphe, objet complexe, en un ensemble d'objets plus basiques. Néanmoins, cette approche ne permet pas de conserver l'information globale sur la structure du graphe.

Nous apportons ici une contrainte sur la définition des chemins. Il est en effet possible de définir un seuil sur p de façon à limiter la longueur des chemins. L'intérêt de cet apport revient à étudier la pertinence de la constitution de chemins de grande longueur. Les chemins plus longs semblent a priori contenir davantage d'information structurelle, mais sont par conséquent une source de redondance car ils contiennent nécessairement des chemins plus courts. Nous pourrons ainsi évaluer la nécessite de ne prendre en compte la totalité ou non des chemins extraits. Lors de la constitution du sac de chemins, nous ne retenons ainsi que les chemins du graphe dont la longueur est inférieure ou égale au seuil fixé.

Enfin, la comparaison de deux chemins h et h' de même longueur p est effectuée à l'aide de la fonction K_L.

$$K_L(h,h') = K_n(\ell(n_1),\ell(n'_1))\times \prod_{i=2}^{p} K_a(\ell(n_{i-1},n_i),\ell(n'_{i-1},n'_i))K_n(\ell(v_i),\ell(v'_i)) \quad (1)$$

Cette comparaison revient à comparer les nœuds et les arcs présents sur les chemins h et h'. Cela revient ainsi à comparer les valeurs des étiquettes présentes. Cette comparaison est effectuée à l'aide de fonctions noyau K_n et K_a qui dépendent ainsi des types de données présentes dans les étiquettes. Dans notre cas, les étiquettes sont des vecteurs de scalaires et peuvent faire appel à un noyau linéaire ou un noyau gaussien, mais nous pouvons facilement imaginer d'appliquer des étiquettes structurées et par conséquent d'utiliser des fonctions noyaux différentes. Il suffit pour cela de redéfinir les fonctions K_n et K_a.

3 Noyau de graphe

Un graphe peut donc être représenté par un sac de chemins. La définition d'un noyau de graphe revient donc à définir un noyau entre des ensembles. Wallraven et al. [10] propose ainsi de construire un noyau à partir d'une combinaison de noyaux mineurs, c'est à dire de combiner des noyaux entre sacs de chemins tel que

$$K(G_1,G_2) = K(P_1,P_2)$$

Kashima propose d'effectuer la somme des comparaisons [5] en les pondérant par les probabilités p_1 et p_2 de générer les chemins puisqu'il définit les parcours de graphe de façon aléaoire :

$$K(G_1,G_2) = \sum_{h_1,h_2\in P_1,P_2} p_1(h_1)p_2(h_2)K_L(h_1,h_2) \quad (2)$$

Nous proposons maintenant de nouvelles formulations de noyau de graphe. La première appelée *k-mean* calcule la moyenne de toutes les comparaisons de chemins :

$$K(G_1,G_2) = \frac{1}{|P_1|}\frac{1}{|P_2|}\sum_{i:h_i\in P_1}\sum_{j:h_j\in P_2} K_L(h_i,h_j) \quad (3)$$

Comme tous les chemins sont pris en compte, certaines mesures peuvent être masquées par des mesures moins importantes. Pour remédier à cet inconvénient, nous proposons de ne retenir que les meilleures valeurs de comparaisons pour chaque chemin. Nous aboutissons ainsi à une deuxième formulation intitulée *max matching kernel* :

$$K(G_1,G_2) = \frac{1}{2}[\hat{K}(P_1,P_2) + \hat{K}(P_2,P_1)] \quad (4)$$

avec $\hat{K}(P_1,P_2) = \frac{1}{|P_1|}\sum_{i:h_i\in P_1}\max_{j:h_j\in P_2} K_L(h_i,h_j)$ qui permet de ne retenir que la meilleure mise en correspondance pour chaque chemin. Comme cette formulation n'est pas définie positive en théorie, nous proposons d'utiliser une approximation de ce noyau qui soit définie positive.

En effet, nous utilisons l'approximation [4] :

$$\max_{j:h_j\in P_2} K_L(h_i,h_j) \approx \sum_{j:h_j\in P_2} K_{d_L}(h_i,h_j)$$

le noyau $K_{d_L}(h_1,h_2) = \exp\left(-\frac{d_L(h_1,h_2)^2}{2\sigma^2}\right)$, d_L étant la distance induite par le noyau K_L, est défini positif pour $\sigma > 0$. La formulation *matching kernel* est donnée par :

$$\hat{K}(P_1,P_2) = \frac{1}{|P_1|}\frac{1}{|P_2|}\sum_{i:h_i\in P_1}\sum_{j:h_j\in P_2} K_{d_L}(h_i,h_j) \quad (5)$$

Une dernière approche est inspirée des travaux de Desobry et al. [3]. Elle consiste à évaluer si les chemins contenus dans les ensembles occupent la même région dans l'espace des caractéristiques. D'après Desobry, cela revient à comparer les distributions de probabilité de chaque ensemble

de chemins. Un SVM *one-class* est ainsi utilisé pour estimer les distributions en résolvant le problème d'optimisation suivant :

$$\min_{f\in\mathcal{H},b} \frac{1}{2}\|f\|^2_{\mathcal{H}} + \frac{1}{\nu n}\sum_i \max(0, b - f(x_i)) - b$$

avec $\mathcal{H}$ l'espace de Hilbert généré par le noyau K_L, $\nu \in [0,1]$ un paramètre de régularisation [7]. La distribution d'un ensemble de chemins S_n est déterminée en calculant le contour de S_n telle que :

$$f_{P_n}(h) = \sum_i \alpha_i^{P_n} K_L(h_i, h) - b_{P_n}$$

Ainsi la formulation *path level set* définie positive du produit scalaire entre deux ensembles est formulée par :

$$\begin{aligned} &K(G_1, G_2) = K(P_1, P_2) \\ &= \langle b_{P_1}, b_{P_2}\rangle \cdot \sum_{i:h_i\in P_1}\sum_{j:h_j\in P_2} \alpha_i^{P_1}\alpha_j^{P_2} K_L(h_i, h_j) \end{aligned} \qquad (6)$$

4 Résultats

Nous avons appliqué ces formulations de noyau de graphe à de la reconnaissance d'objet sur la base Rudgers Tool qui contient 25 objets répartis en 8 classes (figure 3), réparties selon des classes outils ou des formes biologiques. Cette base a déjà fait l'objet de travaux sur la mise en correspondance de graphes [9, 2, 8].

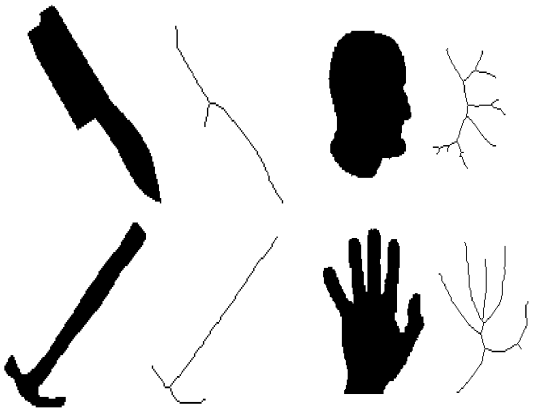

FIG. 1 – Exemple d'images et leur graphe associé

Le test de reconnaissance utilise chaque objet de la base comme une requête dont le résultat est le classement des 24 objets classés par valeur décroissante de leur similarité. La similarité est donnée par la valeur du noyau de graphe. En effet, lorsque le noyau est normalisé, il est possible d'assimiler sa valeur à une norme, utilisable comme mesure de similarité. Le classement est correct si l'objet appartenant à une classe contenant n objets présente un classement dont les $n-1$ objets restant de la classe sont les premiers objets classés.

Les nœuds sont étiquetés par la distance au plus proche point de contour et la distance au centre de l'objet. Les arcs sont étiquetés par leur longueur. Toutes les étiquettes sont normalisées afin d'appartenir à l'intervalle $[0,1]$. Cette normalisation est effectuée indépendamment sur chaque objet. La longueur des arcs est, par exemple, normalisée par rapport à la longueur de la plus petite ellipse contenant l'objet.

Différents paramètres interviennent lors du calcul des noyaux mineurs K_a et K_n. Nous avons effectué différents tests au préalable afin de fixer ces paramètres et avons retenu un noyau gaussien avec une largeur de bande de 0.1 pour tous les noyaux.

Nous présentons maintenant les résultats obtenus pour évaluer l'efficacité de la mesure de similarité par noyau de graphe. Dans un premier temps, nous avons étudié l'évolution des performances lorsque le seuil de la longueur des chemins varie. Ce test nous permet ainsi de vérifier la pertinence de la définition des chemins afin, d'une part, de valider le gain apporté par la présence d'une information de structure, et d'autre part, d'étudier la nécessité de limiter ou non la longueur des chemins extraits des graphes.

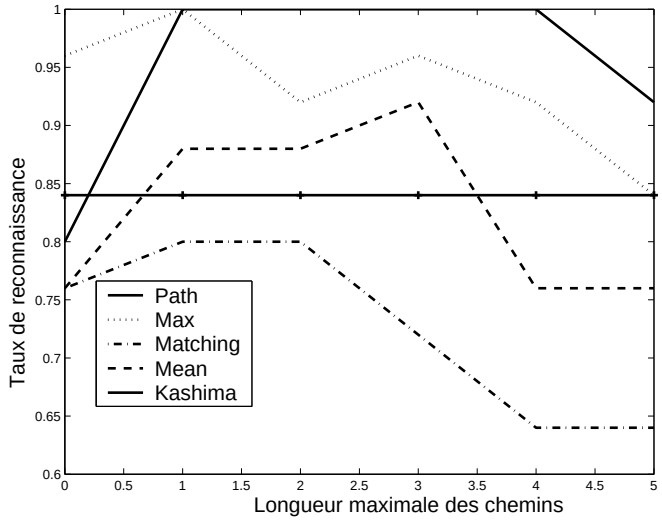

FIG. 2 – Taux de reconnaissance selon la longueur maximale des chemins pour différents noyaux

Nous avons ainsi reporté les résultats obtenus sur la figure 2 lorsque nous faisons varier la longueur maximale possible des chemins générés pour les différentes formulations que nous avons présentées précédemment. Par rapport à l'état de l'art existant sur cette même base d'images, Sebastian [8] et Demirci [2] ont chacun rapporté 1 erreur, soit 96% de bonne classification.

Pour un seuil de 0, les chemins considérés ne contiennent qu'un seul nœud, ce qui peut s'apparenter à une approche de sac de vecteurs où aucune information structurelle n'est présente concernant l'agencement des nœuds. Ici, le noyau *max matching kernel* (eq.4) obtient les meilleures performances avec 96% de bonne reconnaissance, tandis que les autres formulations obtiennent moins de 90%. Nous pouvons observer que le taux de reconnaissance augmente par la suite lorsque le seuil sur la longueur maximale augmente puis décroit. Ce résultat peut s'expliquer par le fait qu'à partir d'une longueur de 1, les sac de chemins contiennent une information sur la structure entre les nœuds, ce qui permet d'améliorer les performances. Ensuite, l'information devient trop redondante, ce qui nuit aux performances.

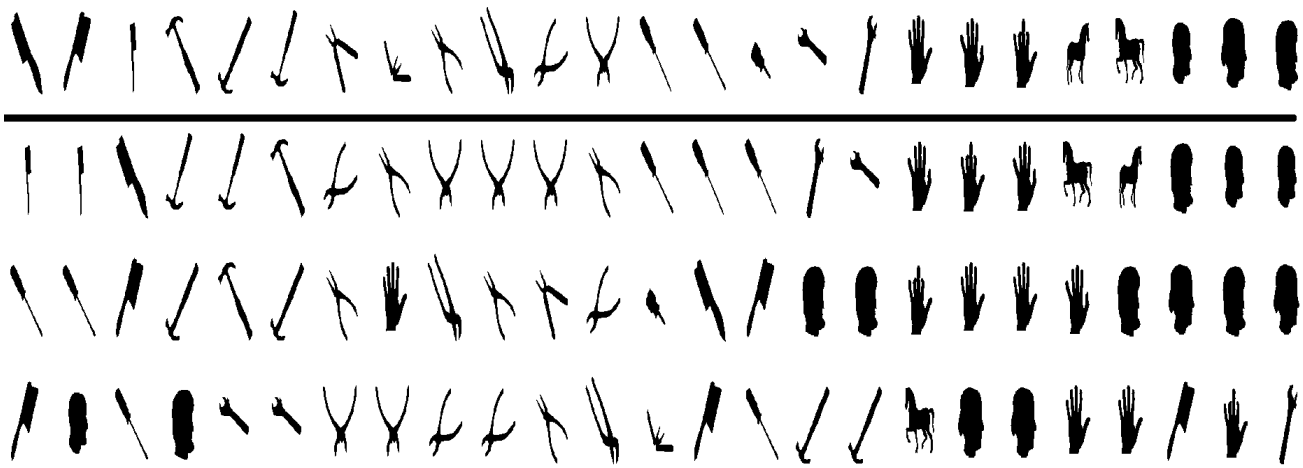

FIG. 3 – Résultat des premiers classements obtenus pour chaque image requête de la première ligne.

Pour cette application, une longueur optimale semble être de 2 ou 3.

Il faut également noter que les étiquettes restent relativement simples mais les formulations *path level-set* (eq. 6) et *max matching kernel* (eq.4) ne commettent aucune erreur sur l'ensemble des requêtes, en particulier lorsque le seuil varie de 1 à 4 pour le noyau *path level-set*.

Enfin, nous remarquons que les formulations *matching kernel* (eq. 5) et *mean kernel* (eq. 3) sont moins performantes que la formulation *max matching kernel* (eq. 4) qu'elles approximent. Malgré la positivité assurée théoriquement, elles se révèlent donc moins performantes qu'une formulation non définie positive. Cela est peut être du également à de mauvais paramètres des noyaux, ou bien une insuffisance d'étiquettes.

La figure 3 montre un exemple de requête pour une longueur maximale de 2 avec le noyau *path level-set*. Les objets mieux classés correspondent à la bonne classe, cependant quelques erreurs apparaissent dans les classement suivant, principalement pour les classes "brosse" et "tournevis" qui peuvent être confondus. En effet, les graphes extraits des formes de ces objets restent relativement proches. Les étiquettes choisies sont peut être insuffisamment discrimantes pour ces classes-ci qui nécessiteraient peut être des données plus locales, telles que la texture ou une information sur les contours.

5 Conclusion

Ces travaux nous ont permis d'introduire le noyau de graphe comme alternative à la mise en correspondance de graphe pour mesurer la similarité entre graphes. Nous avons ainsi montré qu'un graphe peut être représenté par un sac de chemins à partir duquel le noyau est calculé. Cette approche nous a ainsi permis d'étudier de nouvelles formulations de noyau de graphes à l'aide de l'approche *kernel on set*.

Nous avons ensuite comparé ces formulations pour un problème de reconnaissance d'objets et les résultats obtenus sont très prometteurs.

Nous prévoyons par la suite d'apporter de nouvelles informations dans les étiquettes du graphe, par exemple des histogrammes d'orientation de gradient pour les nœuds ou des caractéristiques de texture sur les arcs. Durant nos expériences, nous nous sommes aperçus que la construction du graphe est également dépendante de la qualité du squelette. Un squelette trop bruité conduit en effet à des graphes trop complexes. Une dernière perspective consiste ainsi à étudier de nouvelles méthodes de construction de graphes, par exemple à partir de points d'intérêt ou d'une segmentation région, moins sensible au bruit.

Références

[1] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19 :255–259, 1998.

[2] F. Demirci, A. Shokoufandeh, L. Bretzner, and S. Dickinson. Object recognition as many-to-many feature matching. *Internation Journal of Computer Vision*, 69(2) :203–222, 2006.

[3] F. Desobry, M. Davy, and W.J. Fitzgerald. A class of kernels for sets of vectors. In *Proceedings of the 13th European Symposium on Artificial Neural Networks*, 2005.

[4] B. Haasdonk and C. Bahlmann. Learning with distance substitution kernels. In Springer, editor, *Pattern Recognition - Proc. of the 26th DAGM Symposium*, 2004.

[5] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the Twentieh International Conference on Machine Learning*, 2003.

[6] C. Di Ruberto. Recognition of shapes by attributed skeletal graphs. *Pattern Recognition*, 37(1) :21–31, 2004.

[7] B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, 2001.

[8] T. Sebastian, P. Klein, and B. Kimia. Recognition of shapes by editing shock graphs. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(5) :550–571, 2001.

[9] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35 :13–32, 1999.

[10] C. Wallraven, B.Caputo, and A. Graf. Recognition with local features : the kernel recipe. In *Proceedings of International Conference on Computer Vision*, pages 257–264, 2003.