Carte auto-organisatrice pour graphes étiquetés

Nathalie Villa-Vialaneix*, Madalina Olteanu**
Christine Cierco-Ayrolles*

*Unité BIA, INRA de Toulouse, Auzeville, France
{nathalie.villa,christine.cierco}@toulouse.inra.fr,
http://www.nathalievilla.org
http://carlit.toulouse.inra.fr/wikiz/index.php/Christine_CIERCO-AYROLLES
**SAMM, Université Paris 1, Paris, France
madalina.olteanu@univ-paris1.fr
http://samm.univ-paris1.fr/OLTEANU-Madalina

Résumé. Dans de nombreux cas d'études concrets, l'analyse de données sur les graphes n'est pas limitée à la seule connaissance du graphe. Il est courant que des informations supplémentaires soient disponibles sur les sommets et que l'utilisateur souhaite combiner ces informations à la structure du graphe lui-même pour comprendre l'intégralité des données en sa possession. C'est ce problème que nous souhaitons aborder dans cet article, en nous focalisant sur une méthode de fouille de données qui combine classification (non supervisée) et visualisation : les cartes auto-organisatrices. Nous expliquons comment l'utilisation de méthodes à noyaux permet de combiner de manière efficace des informations de natures diverses (graphe, variables numériques, facteurs, variables textuelles...) pour décortiquer la structure des données et en offrir une représentation simplifiée. Notre approche est illustrée sur divers exemples : un premier exemple, sur des données simulées, permet de comprendre comment se comporte l'algorithme. Un second exemple illustre la méthode sur un graphe réel de plusieurs centaines de sommets, qui modélise un corpus de documents médiévaux.

1 Introduction

Dans de nombreuses applications dans lesquelles les données sont modélisées par un graphe (ou réseau), il est courant que des informations additionnelles (qui n'ont pas de relation directe avec la structure relationnelle du réseau) soient disponibles. Ces informations peuvent qualifier les sommets ou les arêtes du réseau ; dans le premier cas, on parle de « graphes étiquetés ». Les étiquettes peuvent être multiples et de natures diverses (numériques, catégorielles, textuelles). Les analyses statistiques qui visent à aider l'utilisateur à comprendre ses données doivent alors permettre de tenir compte de l'intégralité de l'information disponible et pas seulement de la structure du graphe. La question peut être abordée sous plusieurs angles : en étudiant la corrélation entre distance dans le réseau et similarité entre étiquettes (comme dans le cas des études sur l'homophilie dans les réseaux sociaux : voir Adamic et al. (2003); Crandall et al.

(2008); Aiello et al. (2010); Cointet et Roth (2010)), en faisant appel à des modèles de diffusion pour modéliser une dynamique dans les étiquettes du réseau (voir Valente et al. (1997); Newman (2002); Christakis et Fowler (2007)) ou bien en utilisant des outils issus de la statistique spatiale pour trouver et quantifier des phénomènes d'auto-corrélation dans le réseau (voir Laurent et Villa-Vialaneix (2011)).

Une méthodologie standard pour la fouille de graphe est la recherche de communautés : celle-ci consiste à partitionner les sommets du graphe en groupes de sommets denses qui partagent (comparativement) peu de liens entre eux. Deux articles de revue Fortunato (2010); Schaeffer (2007) présentent les principales approches développées dans ce domaine. Ici, nous présentons une approche pour détecter des communautés telles que non seulement les sommets d'une même communauté soient fortement inter-connectés mais aussi aient des étiquettes similaires. Cruz et al. (2011) aborde ce problème en utilisant une approche en deux temps, recherchant d'abord des communautés de manière classique par optimisation de la modularité, puis rafinant ces communautés en optimisant une enthropie basée sur la description des sommets. Dans cet article, nous proposons une approche en un temps, basée l'algorithme de carte auto-organisatrice Kohonen (2001) : celui-ci permet, en effet, de combiner classification non supervisée et visualisation en projetant les individus étudiés (ici les sommets du graphe) dans des neurones organisés topologiquement sur une grille de faible dimension (généralement égale à 2). La version initiale de l'algorithme est destinée à analyser des individus décrits par des variables numériques mais diverses variantes ont été proposées pour étendre son utilisation à des données décrites par des variables catégorielles (Cottrell et Letrémy, 2005) ou plus généralement à des données décrites par une dissimilarité (Kohohen et Somervuo, 1998; El Golli et al., 2006; Rossi et al., 2007; Hammer et al., 2011; Olteanu et al., 2012) ou par un noyau (Mac Donald et Fyfe, 2000; Villa et Rossi, 2007; Boulet et al., 2008).

Dans cet article, nous étendons l'algorithme SOM à noyau stochastique (décrit dans Mac Donald et Fyfe (2000); Villa et Rossi (2007)) en introduisant une approche multi-noyaux permettant l'intégration d'informations diverses dans la carte produite. La méthodologie proposée est décrite dans la section 2. Elle est ensuite illustrée sur deux exemples, un exemple simulé, utilisé pour montrer comment la méthode se comporte en présence d'informations contradictoires sur les données et un exemple réel issu d'un graphe décrivant des relations entre individus à partir d'un corpus d'actes notariés médiévaux.

2 Une carte auto-organisatrice pour graphe étiqueté

Dans la suite, nous supposerons donné un graphe $\mathcal{G}$ avec n sommets $\{1,\dots,n\}$, simple et pondéré par des poids $(W_{ij})_{i,j=1,\dots,n}$ ($W_{ij} \geq 0$ et $W_{ij} = W_{ji}$). En outre, chaque sommet i est décrit par D « étiquettes » (variables) $(c_i^d)_{d=1,\dots,D}$.

2.1 Noyaux

Les relations entre sommets et les similarités entre étiquettes sont décrites au moyen d'autant de noyaux. Un noyau K sur l'espace abstrait $\mathcal{X}$ est une application de $\mathcal{X} \times \mathcal{X}$ dans $\mathbb{R}$, symétrique ($K(x,x') = K(x',x)$) et positive ($\forall\, i = 1,\dots,N \sum_{kk'} \alpha_k \alpha_{k'} K(x_k,x_{k'}) \geq 0$). Aronszajn (1950) montre qu'une telle application est un produit scalaire d'une projection ϕ des données de $\mathcal{X}$ dans un espace de Hilbert $(\mathcal{H},\langle .,. \rangle)$: $K(x,x') = \langle \phi(x),\phi(x') \rangle$. L'intérêt

croissant autour de ce type de similarités vient du fait qu'une fois le noyau choisi, ni ϕ, ni $\mathcal{H}$ n'ont besoin d'être explicites pour pouvoir calculer des distances entre individus.

Dans le cas d'un graphe, plusieurs noyaux décrivant les similarités entre sommets peuvent être utilisés. Les plus populaires sont des versions régularisées du Laplacien L du graphe (voir Smola et Kondor (2003)), comme le noyau de la chaleur $e^{-\beta L}$ (*heat kernel*, Kondor et Lafferty (2002)) ou bien le noyau de temps de parcours (*commute time kernel*, Fouss et al. (2007)), qui n'est autre que l'inverse généralisée du Laplacien du graphe et s'interprète comme la mesure du temps moyen nécessaire pour relier deux sommets du graphe par une marche aléatoire sur les arêtes.

Pour décrire les similarités entre étiquettes des sommets, plusieurs choix sont possibles, dépendant de la nature des données :

- si les étiquettes $(c_i^d)_i$ sont numériques ($\in \mathbb{R}^M$), le noyau le plus simple est le noyau linéaire : $K_d(c_i^d,c_{i'}^d) = (c_{i'}^d)^T c_i^d$. D'autres noyaux permettent d'appliquer une transformation non linéaire sur les données, permettant de capter des corrélations plus complexes que la corrélation linéaire comme, par exemple, $K_d(c_i^d,c_{i'}^d) = e^{-\beta\|c_i^d - c_{i'}^d\|^2}$ (noyau Gaussien) ou bien $K_d(c_i^d,c_{i'}^d) = (1 + (c_{i'}^d)^T c_i^d)^P$ (noyau polynomial de degré P) ;
- si les étiquettes sont des variables catégorielles, une approche courante est de recourir au codage disjonctif de ces variables (c'est-à-dire à leur recodage par modalité en 0/1) et d'utiliser un noyau pour variable numérique. Notons que, dans le cas où le noyau linéaire est utilisé, cette opération conduit à utiliser comme noyau entre deux individus, le nombre d'attributs communs aux deux individus ;
- si les étiquettes sont des mots ou plus généralement du texte, plusieurs noyaux ont été proposés, tous basés sur le nombre d'occurrences de sous-parties communes aux deux textes comparés (voir Watkins (2000)). Une implémentation de ces noyaux est proposée dans le package **kernlab** du logiciel libre R (voir R Development Core Team (2012); Karatzoglou et Feinerer (2010)).

Le noyau final retenu pour mesurer la similarité globale entre les sommets i et i' est alors

$$K_T(i,i') = \alpha_0 K_0(i,i') + \sum_d \alpha_d K_d(c_i^d,c_{i'}^d)$$

où K_0 est le noyau choisi pour mesurer la similarité induite par la structure du graphe et les $(\alpha_d)_{d=0,\ldots,D}$ sont des réels positifs tels que $\sum_d \alpha_d = 1$. Il est facile de montrer qu'un tel noyau satisfait aux conditions de l'article Aronszajn (1950).

2.2 Carte auto-organisatrice multi-noyaux

Une fois que les diverses informations sur les sommets du graphe ont été combinées pour calculer une mesure de proximité globale entre ces sommets, nous utilisons le noyau résultat K_T comme produit scalaire dans l'algorithme de carte auto-organisatrice pour plonger les sommets du graphe dans une carte de faible dimension. De manière plus précise, la version stochastique de l'algorithme de carte auto-organisatrice à noyau, comme décrite dans Mac Donald et Fyfe (2000); Villa et Rossi (2007), est utilisée pour positionner sur la carte les sommets du graphe. L'organisation des sommets sur la carte tient alors compte, à la fois, de la structure du graphe mais aussi des proximités entre les diverses étiquettes.

De manière plus précise, les sommets du graphe $\mathcal{G}$, $\{1,\ldots,n\}$ sont projetés sur une carte de faible dimension composée de M neurones, $\{U_1,\ldots,U_M\}$. Une relation de voisinage, h, est

définie entre les neurones et évolue au cours de l'algorithme de manière à converger progressivement vers un voisinage restreint au neurone lui-même. Chaque neurone U_j est représenté par un prototype p_j qui prends ses valeurs dans l'espace image, $\mathcal{H}$, induit implicitement par le noyau K_T. Si ϕ est la projection implicite induite par K_T dans $\mathcal{H}$, les prototypes sont définis comme $p_j = \sum_{i=1}^{n} \gamma_{ji}\phi(i)$, tels que $\gamma_{ji} \geq 0$ et $\sum_{i=1}^{n} \gamma_{ji} = 1$. Les $(\gamma_{ji})_{i,j}$ peuvent être initialisés de manière aléatoire.

L'algorithme alterne alors, de manière itérative,

– sélection aléatoire d'un sommet i et affectation dans le neurone pour lequel le prototype est le plus proche au sens de la distance dans $\mathcal{H}$:

$$f(i) \leftarrow \arg\min_j \|\phi(i) - p_j\|_{\mathcal{H}}, \tag{1}$$

– mise à jour des prototypes :

$$\gamma_{ji'} \leftarrow \gamma_{ji'} + \mu^t h^t(f(i),j)\,(\delta_{i'i} - \gamma_{ji'})$$

où $\delta_{i'i} = 1$ si $i = i'$ et 0 sinon. Le paramètre μ^t est adaptatif, généralement décroissant en $1/t$.

L'équation (1) est résolue en remarquant que la norme dans $\mathcal{H}$ est déduite du produit scalaire K. La minimisation s'effectue donc sur les quantités $\sum_{kk'} \gamma_{jk}\gamma_{jk'}K(k,k') - 2\sum_i \gamma_{jk}K(i,k)$. L'algorithme est stoppé à stabilisation de la classification, qui intervient après que la relation de voisinage ait été réduite au seul neurone : dans cette phase finale, l'algorithme est similaire à un algorithme k-means et sa convergence est donc assurée. Une dernière étape affecte tous les sommets du graphe à un neurone selon le critère de l'équation (1).

3 Applications

3.1 Données simulées

Dans cette partie, nous présentons un exemple simple sur des données simulées pour comprendre comment l'algorithme se comporte en présence de données de natures diverses. Pour cela, nous avons généré aléatoirement un jeu de données de 150 observations réparties en 6 groupes de 25 observations chacun, de la manière suivante :

– des relations entre les 150 observations ont été simulées par un graphe simple non pondéré qui ressemble au modèle aléatoire « planted 3-partition » décrit dans Condon et Karp (2001). Les sommets des groupes 1 et 2, des groupes 3 et 4 et des groupes 5 et 6 sont indicernables du point de vue de ce graphe : les arêtes entre les sommets de ces ensembles sont générées de manière aléatoire avec une probabilité égale à $0{,}3$ (modèle de Erdös Reyni, Erdõs et Rényi (1959)). Les arêtes entre les sommets d'ensembles distincts sont générées selon le même modèle aléatoire mais avec une probabilité moindre : $0{,}01$ (entre les sommets des groupes 1 ou 2 et les sommets des groupes 3 ou 4 et entre les sommets des groupes 3 ou 4 et les sommets des groupes 5 ou 6) ou $0{,}005$ (entre les sommets des groupes 1 ou 2 et les sommets des groupes 5 ou 6).
Les graphes simulés sont représentés dans la figure 1 (à gauche).

– des données numériques entre les 150 observations ont été simulées selon deux gaussiennes à deux dimensions : les groupes 1, 3 et 5 et les groupes 2, 4 et 6 sont indistinguables du point de vue des valeurs numériques prises. De manière plus précise, pour

chacun des deux ensembles, des variables aléatoires de distribution Gaussienne, centrée respectivement en $(0,0)$ et en $(1,1)$ et de matrice de covariance $\sigma^2\mathbb{I}_2$ (ou $\mathbb{I}_2$ est la matrice identité) avec $\sigma = 0{,}3$.
Les données numériques sont représentées dans la figure 1 (à droite).

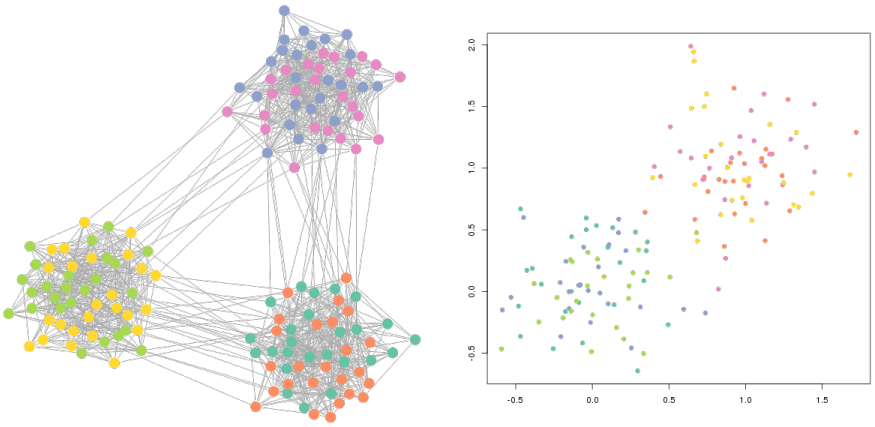

FIG. 1 – *À gauche : graphes aléatoires générés par des modèles de graphes en classes et représentés par un algorithme de force comme décrit dans Fruchterman et Reingold (1991) et implémenté dans le package* R **igraph** *; Csardi et Nepusz (2006)). À droite : distribution des variables numériques. Dans les deux graphiques, les 6 groupes de 25 sommets sont représentés par des couleurs différentes (groupe 1 : vert, groupe 2 : jaune, groupe 3 : bleu/vert, groupe 4 : rouge, groupe 5 : bleu et groupe 6 : rose)*

L'algorithme de cartes auto-organisatrices à noyau a été appliqué pour produire une topologie des données en utilisant

- pour le graphe, le noyau de temps de parcours ;
- pour les données numériques, un noyau Gaussien, dont le paramètre a été calibré de manière automatique selon la méthode décrite dans Caputo et al. (2002).

Trois résultats ont été produits : dans le premier, les deux noyaux ont été combinés (avec $\alpha_1 = \alpha_2 = \frac{1}{2}$) et dans les deux autres, chacun des deux noyaux a été utilisé seul. La carte obtenue est donnée dans la figure 2. Chaque diagramme circulaire représente un neurone de la carte. La taille du diagramme est proportionnelle au nombre d'observations classées dans ce neurone et les couleurs représentent la répartition des six classes initiales dans le neurone. Les arêtes reliant les diagrammes sont de largeur proportionnelle au nombre total d'arêtes reliant les sommets classés dans les deux classes respectivement. Les cartes basées sur une combinaison des noyaux sont les seules à retrouver les 6 classes et à proposer une organisation de celles-ci qui corresponde à l'organisation des données initiales, selon le graphe et les données numériques sous-jacentes.

La même expérience a été répétée 100 fois sur des générations aléatoires de données correspondant à un modèle un peu plus complexe (dans lesquels les groupes de sommets sont moins facilement identifiables) : dans celui-ci, les arêtes d'ensembles distincts sont générées avec une probabilité plus forte, respectivement égale à $0{,}1$ et $0{,}05$ et les écarts types des distri-

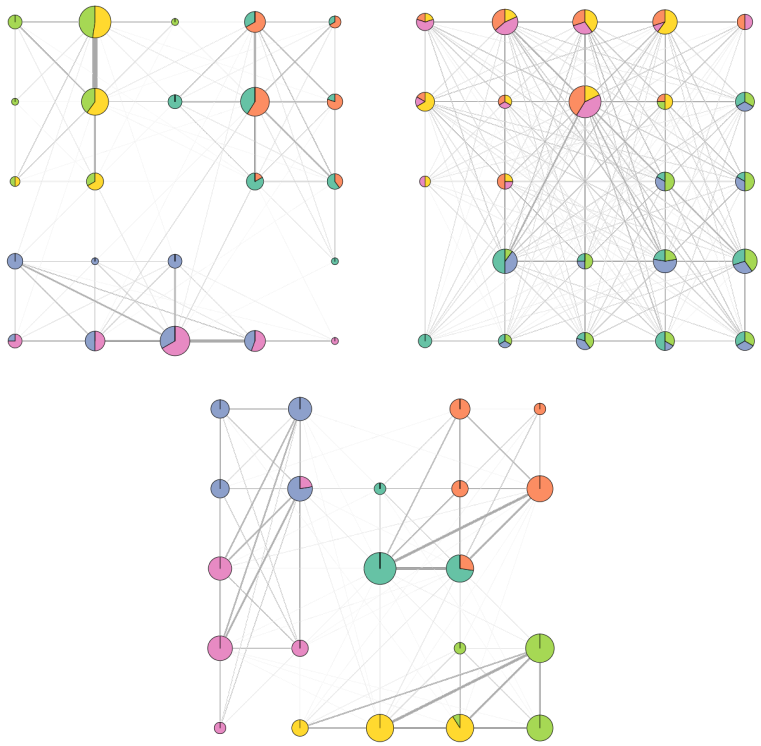

FIG. 2 – *Cartes obtenues à partir du premier jeu de données : en haut à gauche avec le graphe seulement, en haut à droite avec les données numériques seulement et en bas avec la combinaison des deux données. Voir le texte pour une description plus détaillée des graphiques.*

butions Gaussiennes ont été augmentées à $\sigma = 0{,}6$. Enfin, l'information mutuelle normalisée, (voir Danon et al. (2005)) entre les classes sous-jacentes et les classes retenues par l'algorithme a été calculée : cette mesure permettant de quantifier l'adéquation entre deux partitions, est comprise entre 0 et 1 et vaut 1 lorsque les deux partitions sont identiques. Les résultats correspondant à l'utilisation du graphe seul, des données numériques et des deux types de données sont fournies dans la boîte à moustaches de la figure 3. L'information mutuelle normalisée obtenue par le modèle combinant les deux noyaux est largement améliorée par rapport à l'information mutuelle utilisant une seule des deux informations : la combinaison des deux noyaux permet donc bien d'incorporer les informations des deux provenances de manière cohérente.

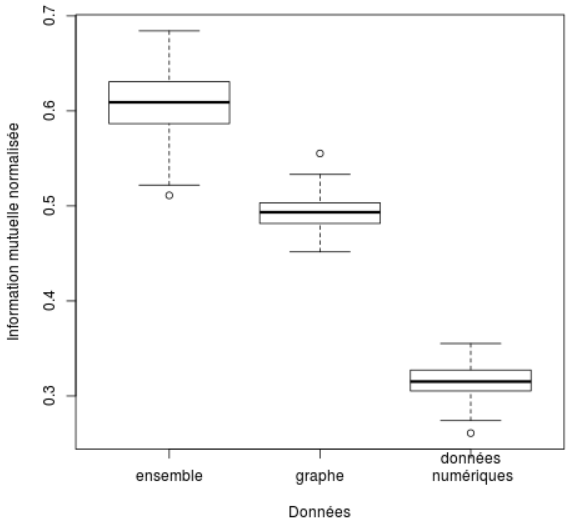

FIG. 3 – *Boîte à moustaches des informations mutuelles normalisées par rapport à la classification de référence sur 100 réalisations aléatoires des données pour diverses cartes auto-organisatrices utilisant l'un ou l'autre des noyaux (sur le graphe ou les données numériques) ou bien une combinaison des deux noyaux (« ensemble »).*

3.2 Données issues d'un corpus d'archives médiévales

Le graphe étudié dans cet exemple est issu d'un corpus d'actes notariés médiévaux décrit dans Boulet et al. (2008); Rossi et Villa-Vialaneix (2011); Hautefeuille et Jouve (2012). Ce corpus est le travail original d'un feudiste qui a été employé pour collecter et retranscrire tous les actes notariés mentionnant des rentes qui auraient été rédigés sur les seigneurie de "Castelnau Montratier" entre 1250 et 1700 (approximativement). Ce travail était destiné au nouveau propriétaire de la seigneurie, pour lui permettre de lever les loyers qui devaient lui revenir. Les documents sont donc tous de nature assez similaire et limité à une aire géographique étroite (d'environ 300 km^2).

Chacun des actes du corpus contient une ou plusieurs trasactions qui ont été digitalisés dans une base de données consultable librement à `http://graphcomp.univ-tlse2.fr`. Les transactions elles-mêmes contiennent des informations variées, avec des degrés de précision divers selon les transactions. En général, sont au moins mentionnés les noms des participants et la date de la transaction (au moins l'année). Un graphe a été tiré de ces informations dans lequel

- les sommets modélisent les individus directement impliqués dans les transactions (les notaires et les confronts parfois mentionnés ne sont donc pas inclus). Le graphe contient 1 446 individus et 3 192 arêtes.
- deux sommets sont reliés par une arête si les deux individus ont été impliqués dans une transaction commune ;
- les sommets sont étiquetés par la date moyenne d'activité de l'individu (variable numérique) et par le nom de famille de l'individu (variable textuelle).

L'utilisation de ces trois informations permet donc de regrouper des individus de la même famille, ayant des relations sociales similaires et une activité à des dates proches. Les noyaux suivants ont été combinés avec un poids identique ($\alpha_d = 1/3$ pour $d = 1,..,3$) :

- noyau de temps de parcours pour le graphe ;
- noyau linéaire pour les dates ;
- noyau spectral pour les distances entre noms de famille (voir Karatzoglou et Feinerer (2010), ici nous avons fixé le paramètre de taille des séquences communes comptées à 4).

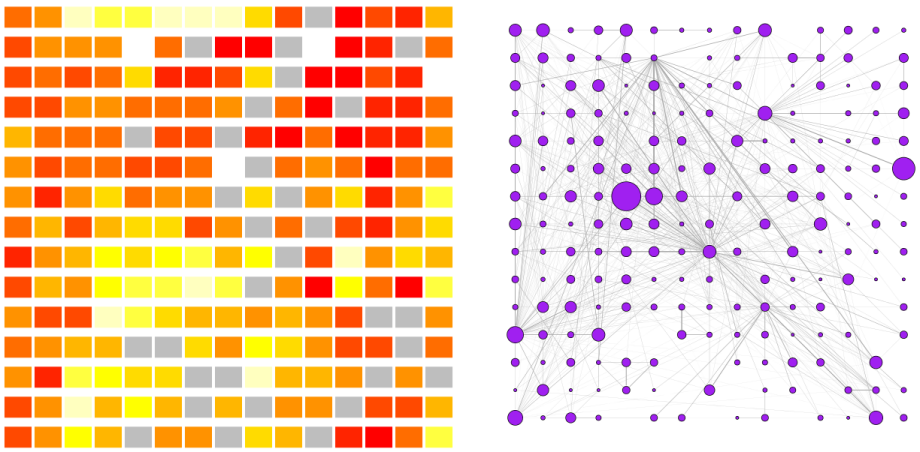

FIG. 4 – *Carte des dates moyennes par neurones (à gauche : les dates les plus récentes sont en jaunes, les plus anciennes en rouge) et carte du réseau (à droite : les aires des sommets sont proportionnelles au nombre d'individus classés dans le neurone considéré et les épaisseurs des arêtes sont proportionnelles au nombre total de liens entre les individus des deux neurones divisé par la racine carré du produit des deux effectifs des neurones pour éviter de sur-représenter des arêtes entre sommets de fort effectif).*

Les résultats sont présentés dans les figures 4 et 5. On y observe une bonne organisation des noms de famille, avec un regroupement spatialisé des noms de famille comme par exemple, la

FIG. 5 – *Carte des principaux noms de famille (au maximum deux par neurone) et effectif d'apparition du nom dans le neurone.*

famille Combelcau (en bas à droite), proche des familles Combe et Lacombe qui sont similaires au niveau des noms (dont l'orthographe dérive assez facilement à cette époque) ou bien la famille Bosseran en haut au centre ou la famille Estairac, aussi en haut au centre. De manière répétée dans la carte, le neurone contient des individus de même nom, avec des occurences fréquemment supérieures à 5. La carte présente également une bonne organisation selon les dates. Certains neurones proches correspondent à des dates assez différentes : en bas à droite, par exemple, les dates des neurones sont différentes mais les noms de famille similaires. La topologie reflète donc une similarité entre individus (même famille) et les différents clusters permettent de séparer les individus selon leur période d'activité. Du point de vue du réseau,

la carte présente aussi une bonne organisation, avec des neurones plus fortement connectés au centre de la carte et des neurones plus isolés sur les bords. Les communautés extraites semblent cohérentes du point de vue des trois données d'entrée, fournissant à l'utilisateur historien, un moyen de se focaliser sur des communautés locales, homogènes du point de vue des dates, des relations et des familles impliquées.

4 Conclusion

Dans cet article, nous avons présenté une approche permettant d'obtenir des communautés dans un graphe en tenant compte d'informations additionnelles connues sur les sommets. L'utilisation d'une combinaison de noyaux semble être une méthodologie pertinente qui arrive à tirer profit de l'ensemble des données disponibles. Dans les applications présentées, la combinaison linéaire a été réalisée au moyen de poids identiques, chaque noyau étant donc considéré avec la même importance. Une extension de ce travail, soumis pour publication (Olteanu et al., 2013), permet l'apprentissage adaptatif des poids de la combinaison des noyaux pour optimiser la classification en donnant plus ou moins d'importance à certains types d'informations.

Références

Adamic, L., O. Buyukkokten, et E. Adar (2003). A social network caught in the web. *First Monday 8*.

Aiello, L., A. Barrat, C. Cattuto, G. Ruffo, et R. Schifanella (2010). Link creation and profile alignment in the aNobii social network. In *Proceedings of the Second IEEE International Conference on Social Computing (SocialCom)*, Minneapolis, USA.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society 68*(3), 337–404.

Boulet, R., B. Jouve, F. Rossi, et N. Villa (2008). Batch kernel SOM and related laplacian methods for social network analysis. *Neurocomputing 71*(7-9), 1257–1273.

Caputo, B., K. Sim, F. Furesjo, et A. Smola (2002). Appearance-based object recognition using svms : which kernel should i use ? In *Proceedings of NIPS workshop on Statistical methods for computational experiments in visual processing and computer vision*, Whistler.

Christakis, N. et J. Fowler (2007). The spread of obesity in a large social network over 32 years. *New England Journal of Medicine 357*, 370–379.

Cointet, J. et C. Roth (2010). Local networks, local topics : Structural and semantic proximity in blogspace. In *Proceedings of the International Conference on Weblogs and Socail Media (ICWSM)*, Washington, DC, USA.

Condon, A. et R. Karp (2001). Algorithms for graph partitioning on the planted partition model. *Random Structures Algorithms 18*(2), 116–140.

Cottrell, M. et P. Letrémy (2005). How to use the Kohonen algorithm to simultaneously analyse individuals in a survey. *Neurocomputing 63*, 193–207.

Crandall, D., D. Cosley, D. Huttenlocher, J. Kleinberg, et S. Suri (2008). Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th SIGKDD*, pp. 160–168.

Cruz, J., C. Bothorel, et F. Poulet (2011). Entropy based community detection in augmented social networks," , 2011 international conference, pp.163-168 doi : 10.1109/cason.2011.6085937. In *Proceedings of Computational Aspects of Social Networks (CASoN)*, pp. 163–168.

Csardi, G. et T. Nepusz (2006). The igraph software package for complex network research. *InterJournal Complex Systems*.

Danon, L., A. Diaz-Guilera, J. Duch, et A. Arenas (2005). Comparing community structure identification. *Journal of Statistical Mechanics*, P09008.

El Golli, A., F. Rossi, B. Conan-Guez, et Y. Lechevallier (2006). Une adaptation des cartes auto-organisatrices pour des données décrites par un tableau de dissimilarités. *Revue de Statistique Appliquée LIV*(3), 33–64.

Erdõs, P. et A. Rényi (1959). On random graphs. i. *Publicationes Mathematicae 6*, 290–297.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports 486*, 75–174.

Fouss, F., A. Pirotte, J. Renders, et M. Saerens (2007). Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. *IEEE Trans Knowl Data En 19*(3), 355–369.

Fruchterman, T. et B. Reingold (1991). Graph drawing by force-directed placement. *Software Pract Exper 21*, 1129–1164.

Hammer, B., A. Gisbrecht, A. Hasenfuss, B. Mokbel, F. Schleif, et X. Zhu (2011). Topographic mapping of dissimilarity data. In *Proceedings of WSOM 2011*, pp. 1–15.

Hautefeuille, F. et B. Jouve (2012). Defining rural elites in the 13th to 15th centuries at the crossroads of historical, archeological and mathematical approaches. Submitted to the Journal of Interdisciplinary History.

Karatzoglou, A. et I. Feinerer (2010). Kernel-based machine learning for fast text mining in R. *Computational Statistics and Data Analysis 54*, 290–297.

Kohohen, T. et P. Somervuo (1998). Self-organizing maps of symbol strings. *Neurocomputing 21*, 19–30.

Kohonen, T. (2001). *Self-Organizing Maps, 3rd Edition*, Volume 30. Berlin, Heidelberg, New York : Springer.

Kondor, R. et J. Lafferty (2002). Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th International Conference on Machine Learning*, pp. 315–322.

Laurent, T. et N. Villa-Vialaneix (2011). Using spatial indexes for labeled network analysis. *Information, Interaction, Intelligence (i3) 11*(1).

Mac Donald, D. et C. Fyfe (2000). The kernel self organising map. In *Proceedings of 4th International Conference on knowledge-based intelligence engineering systems and applied technologies*, pp. 317–320.

Newman, M. (2002). Spread of epidemic disease on networks. *Phys Rev E 66*(016128).

Olteanu, M., N. Villa-Vialaneix, et C. Cierco-Ayrolles (2013). Multiple kernel self-organizing maps. Submitted.

Olteanu, M., N. Villa-Vialaneix, et M. Cottrell (2012). On-line relational som for dissimilarity data. In P. Estevez, J. Principe, P. Zegers, et G. Barreto (Eds.), *Advances in Self-Organizing Maps (Proceedings of WSOM 2012)*, Volume 198 of *AISC (Advances in Intelligent Systems and Computing)*, Santiago, Chile, pp. 13–22. Springer Verlag, Berlin, Heidelberg.

R Development Core Team (2012). *R : A Language and Environment for Statistical Computing*. Vienna, Austria. ISBN 3-900051-07-0.

Rossi, F., A. Hasenfuss, et B. Hammer (2007). Accelerating relational clustering algorithms with sparse prototype representation. In *6th International Workshop on Self-Organizing Maps (WSOM)*, Bielefield, Germany. Neuroinformatics Group, Bielefield University.

Rossi, F. et N. Villa-Vialaneix (2011). Représentation d'un grand réseau à partir d'une classification hiérarchique de ses sommets. *Journal de la Société Française de Statistique 152*(3), 34–65.

Schaeffer, S. (2007). Graph clustering. *Computer Science Review 1*(1), 27–64.

Smola, A. et R. Kondor (2003). Kernels and regularization on graphs. In M. Warmuth et B. Schölkopf (Eds.), *Proceedings of the Conference on Learning Theory (COLT) and Kernel Workshop*, Lecture Notes in Computer Science, pp. 144–158.

Valente, T., S. Watkins, M. Jato, A. van der Straten, et L. Tsitsol (1997). Social network associations with contraceptive use among comeroonian women in voluntary associations. *Social Science & Medecine 45*, 677–687.

Villa, N. et F. Rossi (2007). A comparison between dissimilarity SOM and kernel SOM for clustering the vertices of a graph. In *6th International Workshop on Self-Organizing Maps (WSOM)*, Bielefield, Germany. Neuroinformatics Group, Bielefield University.

Watkins, C. (2000). Dynamic alignment kernels. In A. Smola, P. Bartlett, B. Schölkopf, et D. Schuurmans (Eds.), *Advances in Large Margin Classifiers*, Cambridge, MA, USA, pp. 39–50. MIT P.

Summary

In a number of real-life applications, the user is interested in analyzing the graph together with additional information known on its nodes. The combination of all the sources of information can help him to better understand the dataset in its whole. The present article focus on such an issue, by using self-organizing maps, that combine clustering and visualization. Using a kernel version of the algorithm makes it possible to combine various types of information (graph, numerical values, factors, strings...). Several simplified representations of the data can then be derived from the obtained map. The approach is illustrated on two examples: the first one is simulated data that seeks at proving the usefulness of the method. The second example is a real-life application where the graph, that contains several hundred nodes, has been extracted from a corpus of medieval documents.