ATELIER FOUILLE VISUELLE DE DONNEES : METHODOLOGIE ET EVALUATION

Dans le cadre de la conférence Extraction et Gestion de Connaissances (EGC'2012)

Bordeaux, 31 janvier 2012

Hanene Azzag, Bénédicte Le Grand, Monique Noirhomme, Fabien Picarougne, François Poulet

Atelier Fouille Visuelle de Données : méthodologie et évaluation

Hanene Azzag*, Bénédicte Le Grand**, Monique Noirhomme***, Fabien Picarougne****, François Poulet♯

*Université Paris 13, LIPN-UMR 7030 - CNRS 99, avenue Jean-Baptiste Clément
93430 Villetaneuse, France
hanene.azzag@lipn.univ-paris13.fr
**LIP6 - Université Pierre et Marie Curie - 4, place Jussieu
75005 Paris, France
benedicte.le-grand@lip6.fr
***Institut d'Informatique, Facultés Universitaires Notre-Dame de la Paix (FUNDP)
Rue Grandgagnage, 21
B-5000 Namur, Belgique
monique.noirhomme@info.fundp.ac.be
****LINA - Polytech'Nantes, rue Christian Pauc
BP50609 F-44306 Nantes Cedex 3, France
fabien.picarougne@univ-nantes.fr
♯Universite de Rennes I - IRISA, Campus de Beaulieu
35042 RENNES Cedex, France
francois.poulet@irisa.fr

L'édition 2012 de l'atelier « *Fouille Visuelle de Données* » met en avant les aspects de méthodologie et d'évaluation. Le but de cet atelier est de fournir un lieu d'échange et de présentation de méthodes nouvelles, d'axes de recherche, de développements dans le domaine de la visualisation en extraction de connaissances et de la fouille visuelle de données classiques et/ou structurées en graphes.

Cet atelier prend pour support les groupes de travail *Fouille de Grands Graphes* (EGC-FGG) et *Visualisation d'informations, interaction et fouille de données* (GT-VIF). Notre ambition est de permettre aux participants d'aborder tous les thèmes de la Fouille Visuelle de Données et concerne aussi bien les chercheurs du monde académique que ceux du secteur industriel, et autant les notions conceptuelles que les applications. Les thèmes abordés en Fouille Visuelle de Données sont très nombreux et nous nous intéresserons principalement aux questions relatives au choix des méthodologies de conception en fouille visuelle de données et en visualisation d'information, à la réalisation efficace et à la confrontation de ces principes aux applications réelles, notamment dans le domaine des grands graphes. Nous débattrons également sur les succès et échecs récemment rencontrés.

Comité d'organisation

- Hanene Azzag (LIPN, Université de Paris 13)
- Bénédicte Le Grand (LIP6-Université Pierre et Marie Curie - Paris 6)
- Monique Noirhomme (Institut d'Informatique, FUNDP, Namur, Belgique)
- Fabien Picarougne (LINA, Université de Nantes)
- François Poulet (IRISA, Université de Rennes I)

Comité de programme

- Nadir Belkhiter, (Université de Laval, Québec)
- Sadok Ben Yahia (Université de Tunis El Manar, Tunisie)
- Fatma Bouali (LI Tours et Université de Lille2)
- Lydia Boudjeloud (LITA, Université de Metz)
- Fanny Chevalier, (Ontario College of Art and Design University, Canada)
- Etienne Cuvelier, (Ecole Centrale, Paris)
- Thanh-Nghi Do, (Université de Can Tho, Vietnam)
- Jean-Daniel Fekete (INRIA)(Président AFIHM)
- Fabrice Guillet (LINA, Université de Nantes)
- Jean-Loup Guillaume, (LIP6, Université Pierre et Marie Curie - Paris 6, Paris)
- Rushed Kanawati (LIPN, Université Paris 13)
- Pascale Kuntz (LINA, Université de Nantes)
- Mustapha Lebbah (LIPN, Université de Paris 13)
- Philippe Lenca, (Télécom Bretagne, Brest)
- Benoit Otjacques (Centre Gabriel Lippmann, Luxembourg)
- Paul Richard (ISTIA, Université d'Angers)
- Gilles Venturini (LI, Université de Tours)
- Fabrice Rossi (SAMM, Université Paris 1 Panthéon-Sorbonne)
- Michel Soto, (LIP6, Université René Descartes, Paris 5)

Programme

14h - 14h10
Accueil et présentation de l'atelier

14h10 - 14h30
Présentation du Groupe de Travail conjoint entre l'AFIHM et EGC *Visualisation d'informations, interaction et fouille de données - (EGC-VIF)*
Gilles Venturini

14h30 - 15h
Détection Visuelle de Communautés : Une Evaluation des Approches 2D, 3D Perspective et 3D Stéréo
Nicolas Greffard, Fabien Picarougne, Pascale Kuntz p. 4

15h - 15h30
Évaluation de ProxiViz pour la fouille visuelle de données multidimensionnelles
Nicolas Heulot, Michaël Aupetit, Jean-Daniel Fekete p. 15

15h30 - 16h
Pause café

16h - 16h30
Clustering multi-niveaux de graphes : hiérarchique et topologique
Nhat-Quang Doan, Hanane Azzag, Mustapha Lebbah p. 27

16h30-17h
Exploration graphique de données séquentielles
Reto Bürgin, Gilbert Ritschard, Emmanuel Rousseaux p. 39

17h-17h30
FOLKOVIZ : Visualisation socio-sémantique de folksonomies
Amira Mouakher, Mariam Daoud, Sadok Ben Yahia p. 51

Détection Visuelle de Communautés: Une Evaluation des Approches 2D, 3D Perspective et 3D Stéréo

Nicolas Greffard*, Fabien Picarougne*
Pascale Kuntz*

*Equipe COD - LINA - Polytech'Nantes, rue Christian Pauc
BP50609 F-44306 Nantes Cedex 3
{prénom.nom}@univ-nantes.fr,
http://http://www.polytech.univ-nantes.fr/COD/

Résumé. Depuis les années 90, les problèmes de tracés de graphes en 3D ont été principalement restreints à la 3D perspective. Cependant, les avancées technologiques récentes permettent d'implémenter des solutions 3D stéréoscopiques d'une grande qualité et à faible coût via l'intégration de la disparité binoculaire, l'un des facteurs principaux de la perception de la profondeur. Ce papier explore l'intérêt de la stéréoscopie dans le cadre de la détection visuelle de communautés, qui est une tache de grande importance dans l'analyse de réseaux sociaux. Une évaluation menée sur 35 utilisateurs avec des graphes de complexité croissante indique que dans la majorité des cas, la stéréoscopie offre de meilleurs performances que la 3D perspective. En comparant la stéréoscopie avec des tracés 2D, bien que le temps de réponse soit en faveur de ces derniers, la qualité des résultats dépend principalement de la complexité des graphes observés. Pour un grand nombre de clusters et une probabilité de chevauchements importante, la stéréoscopie connait de meilleurs de résultats que l'approche 2D qui, quant à elle, semble plus appropriée pour les structures les plus simples.

1 Introduction

Longtemps après les travaux pionniers de Kolmogorov et Barzdin (1967), les tracés de graphes en 3D ont connu une phase de grand intérêt dans les années 90 dans la communauté de Graph Drawing. Mis à part la beauté théorique de la question, l'intérêt était principalement motivé par le développement de nouvelles technologies 3D et par l'exploration de nouvelles applications émergentes comme le VLSI design (ex. Rosenberg (1983) Battista et al. (1998), Eades et al. (2000), Wood (2003)).

Cependant, l'intérêt envers la 3D a rapidement décliné, à tel point, qu'il est parfois considéré comme un épiphénomène préjudiciable dans la communauté de Graph Drawing (cf. la conférence invitée de Eades à GD'10 [1]). La critique principale de la 3D concerne principalement le manque de visibilité entrainé par l'occlusion, inhérente à l'utilisation de la troisième

1. http ://www.graphdrawing.org/gd2010/invited.html

dimension.

Nous pensons cependant que ce declain est du à une mauvaise déf nition de la 3D qui a été le plus souvent employée comme de la 2D, sans se soucier de la perception de la dimension supplémentaire. Or, les avancés technologiques récentes en matière de 3D stéréoscopique permettent d'implémenter des restitutions d'une grande qualité tout en introduisant la disparité binoculaire, l'un des facteurs principaux de la perception de la profondeur. Bien que la question générale des bénéf ces de la stéréoscopie par rapport à la 2D reste une question largement ouverte, les investigations concernant l'utilité de la 3D attirent une communauté grandissante de chercheurs (se référer à Teyseyre et Campo (2009) pour une vue d'ensemble).

Dans la visualisation de graphes, des travaux récents ont montré l'intérêt de représentations en 3D stéréoscopique de tracés de graphes pour des tâches d'analyse local. En particulier, Ware et Mitchell (2008) ont conf rmé de manière empirique la valeur ajoutée de la 3D stéréoscopique dans une tâche de tracé de plus court chemin entre deux nœuds dans des graphes de taille limitée (Belcher et al. (2003)).

Dans cet article, nous explorons l'intérêt de la stéréoscopie dans une tâche de plus haut niveau : l'identif cation de communautés (i.e. des sous ensembles de nœuds avec une forte densité de connexions entre eux). Cette tâche est d'une extrême importance dans l'analyse de réseaux sociaux où la visualisation connait un engouement croissant. La plupart du temps, ces communautés sont d'abord identif ées par une approche de classif cation automatique(voir Fortunato (2010) pour un état de l'art), et une représentation visuelle de ces clusters est ensuite utilisée. Cependant, la détection de communauté souffre d'un problème majeur : dans beaucoup de jeux de données réels, les communautés ne forment pas de partitions non-ambigües du graphe, et de nombreux chevauchements existent. De façon à surmonter cette diff culté, différentes stratégies alternatives ont été proposées : ex. donner un placement particulier à certains nœuds pré-déf nis (acteurs centraux) qui sont membres de communautés différentes (Auber et al. (2003)), ou encore, dupliquer les nœuds qui appartiennent à différentes communautés (ex. Henry et al. (2008)). D'autres techniques utilisant une autre représentation que les diagrammes nœuds-liens existent, mais nous nous concentrons sur cette représentation dans cet article, puisqu'elle est de loin, la plus populaire, et la seule à avoir été étudiée en 3D.

Dans ce contexte, nous analysons l'utilisation de la stéréoscopie pour la détection de communautés dans une représentation "brute" de graphes -obtenue via l'algorithme Fruchterman-Reingold. Bien que cet algorithme ne soit pas le plus adapté pour cette tâche, notre choix s'est porté sur lui pour deux raisons. Tout d'abord, il permet de mettre en évidence ces communautés sans nécessiter de pré ou post traitement. Mais il est également applicable aussi bien à la 2D qu'à la 3D, ce qui limite considérablement les biais liés à la comparaison de tracés différents. Autrement dit, nous essayons de répondre à la question suivante : "Quelle est la différence entre la représentation stéréoscopique d'un graphe et ses correspondances 2D et 3D perspective dans le cadre d'une tâche d'identif cation de communautés dans des graphes de tailles moyennes et de différentes complexités ?". Nous avons donc conduit une expérience utilisateur avec des graphes pseudo-aléatoires : nous avons demandé aux participants de déterminer le nombre de communautés qu'ils pouvaient détecter tout en mesurant leurs temps de réponse.

La suite de cet article est organisée comme suit. La section 2 introduit quelques notions psycho-visuelles qui ont guidé notre recherche. La section 3 détaille la procédure expérimentale que nous avons menée. Et enf n, les résultats sont analysés dans la Section 4.

2 Perception de la Profondeur

Nous vivons dans un espace tridimensionnel et une longue période d'évolution nous a doté d'organes nous permettant de percevoir cet espace via l'information visuelle. Un grand nombre de recherches dans le domaine des sciences cognitives ont été dédiées à l'étude des méchanismes impliqués dans la perception de cet environnement 3D (e.g. Landy et al. (1991)).

En sus de ceux associés à la détection des formes et des objets, les méchanismes biologiques qui gouvernent la perception de la distance le long de l'axe optique -la profondeur- jouent un rôle crucial. La perception de la profondeur est certainement le fruit de la combinaison de plusieurs méchanismes perceptifs et de nombreuses études ont porté sur l'évaluation de leurs performances relatives (e.g. Hubona et al. (1999), van Schooten et al. (2010), Ware et Mitchell (2005)). Succinctement, il semblerait que l'occlusion (un objet cachant partiellement un autre objet) soit le facteur principal, quelle que soit la distance entre la personne et l'objet. Pour des distances limitées (moins de 40m), la disparité binoculaire associée à la vision stéréoscopique et la perspective de mouvement ont également un impact important. Concernant les distances les plus grandes -qui ne sont pas considérées dans cet article- d'autres facteurs telle que la perspective aérienne rentrent en jeu (Cutting (1997), Saracini et al. (2009)).

La comparaison des effets relatifs de la stéréoscopie et de la perspective de mouvement reste sujette à débat. Le consensus semble pourtant indiquer que ces deux facteurs sont équivalents ou complémentaires dans de nombreuses tâches (e.g. Ware et Franck (1996), Domini et al. (2006)). Plus précisement, le mouvement associé à une projection 3D perspective peut restituer une impression de profondeur authentique : la rotation d'un objet agit coinjointement avec la mémoire spatiale af n de former une représentation mentale de l'objet en 3D (van Schooten et al. (2010)). Cependant, le mouvement est également utile tant en perspective qu'en stéréoscopie af n de révéler des objets cachés dans un axe visuel particulier. Ainsi, mesurer l'interdépendance entre ces deux facteurs est une tâche très diff cile. Ce papier se restreint donc à la comparaison de la stéréoscopie et de la 3D perspective combinées à des mouvements basiques à un niveau macroscopique.

3 Protocole Expérimental

Trois méthodes de visualisation ont été employées durant cette expérience.

- 2D : Un tracé 2D était pré-calculé en utilisant l'algorithme de Fruchterman-Reingold et aff ché sur une surface 2D. Les participants avaient la possibilité de (dé)zoomer et d'effectuer une rotation sur l'axe optique en bougeant la souris.
- 3D Perspective (3D persp) : Un tracé 3D était pré-calculé en utilisant la version tridimensionnelle de l'algorithme de Fruchterman-Reingold et aff ché sur une surface 2D via une projection perspective. En plus du (dé)zoom et de la rotation sur l'axe optique,

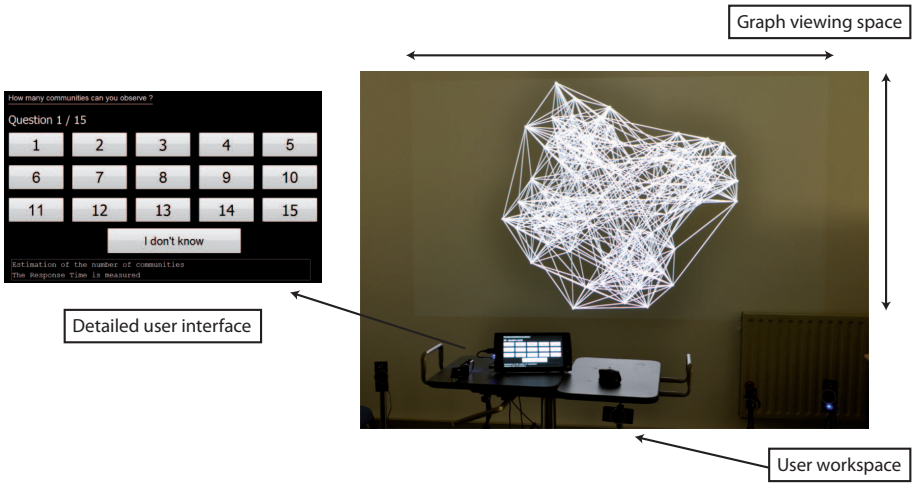


FIG. 1 – *Photographie de l'installation ainsi qu'une capture d'écran de l'interface utilisée par les participants.*

les participants avaient la possibilité de faire tourner leur point de vue autour du graphe (rotation sur x et y).

– 3D Stereoscopique (3D stereo) : Le même tracé et la même projection perspective que pour le cas précédent étaient utilisés mais avec l'introduction d'un deuxième point de vue -calculé en temps réel- permettant d'introduire la disparité binoculaire : un point de vue pour chaque oeil avec un léger décalage le long de l'axe horizontale afin d'imiter la séparation naturelle entre les yeux humains. Les mêmes interactions que pour la perspective étaient disponibles.

3.1 Matériel

Le système de visualisation utilisé tournait sur un ordinateur doté d'un processeur Intel Core 2 Duo (3.00 Ghz) E8400 avec 4 GB de RAM et une carte graphique Nvidia Quadro FX 3800. Tous les graphes étaient affichés en nuances de gris sur un fond noir, avec un algorithme d'anticrénelage afin d'améliorer la qualité de la restitution. La visualisation était projetée sur un mur blanc via un projecteur 3D Acer H5360 (écran de $2,30 \times 1,30\,\mathrm{m}^2$) ayant une résolution de 1280×720 pixels (angle de vue de 0.05 degrés pour un pixel au centre de l'écran). Notre système utilisait une technique de stéréoscopie dite active avec des lunettes à obturateur Nvidia 3D Vision. Utiliser ces lunettes entraine une baisse significative de la luminisoté perçue. Ainsi, afin d'éviter tout biais lié au port de ces lunettes, les participants devaient les garder durant toute l'expérience, quelle que soit la méthode de visualisation utilisée. Les participants pouvaient interagir avec le système via une souris sans fil. Les réponses étaient entrées en utilisant une tablette tactile : différents numéros (allant de 1 à 15, plus une option "Je ne sais pas") étaient affichés et l'utilisateur avait comme instruction de toucher (cliquer) sur le numéro correspondant à sa réponse. Une photographie de l'installation est présentée sur la Fig. 1.

3.2 Base de Données de Graphes

Af n d'analyser l'impact de la stéréoscopie pour différentes topologies, nous avons généré des graphes en utilisant un modèle pseudo aléatoire (ex. Garbers et al, 1990). Le modèle générique $G\left(k, nv, p_{int}, p_{ext}\right)$ employé, dépend de 4 paramètres : le nombre k de communautés *a priori*, le nombre de nœuds par communauté nv et la probabilité p_{int} (resp. p_{ext}) d'avoir un lien entre deux nœuds appartenant à une même communauté (resp. à des communautés différentes). Nous avons ainsi généré 545 graphes avec les paramètres suivants : $k \in \{4, 5, ..., 11\}$, $nv = 10, 20, 30, 40$, $p_{int} = 0.5, 0.6, 0.7, 0.8$ and $p_{ext} = 0.02, 0.03, 0.04, 0.05, 0.065, 0.07, 0.08, 0.1$. Les paramètres p_{int} et p_{ext} ont été déterminés de manière empirique lors d'une étude précédente menée par deux utilisateurs conf rmés. Notons que les paramètres p_{int} et p_{ext} ont été considérés par paires formant ainsi les ratios de complexité :

$$\frac{p_{int}}{p_{ext}} = \left\{ \frac{0.02}{0.8}, \frac{0.02}{0.7}, \frac{0.03}{0.8}, \frac{0.03}{0.7}, \frac{0.03}{0.6}, \frac{0.04}{0.7}, \frac{0.03}{0.5}, \frac{0.05}{0.8}, \frac{0.05}{0.7}, \frac{0.05}{0.6}, \frac{0.065}{0.6}, \frac{0.07}{0.6}, \frac{0.1}{0.8}, \frac{0.08}{0.6}, \right\}$$

Trois exemples de graphes 2D générés par ce modèle sont présentés sur la Fig. 2.

3.3 Participants

35 participants (25 hommes et 10 femmes) ont pris part à cette expérience. Agés de 20 à 50 ans, 30 d'entre eux venaient d'un cursus informatique (chercheurs et étudiants). Trois des participants étaient gauchers avec une utilisation de la souris de la main droite. Seulement deux des participants n'avaient jamais visualisé la moindre image stéréoscopique et 11 parmi les 35 n'étaient pas familiers avec les logiciels 3D comme les jeux vidéos.

3.4 Procédure Expérimentale

Af n de limiter la durée de l'expérience, 15 tracés successifs par méthode de visualisation étaient présentés à chaque participant. Ces tracés étaient tirés de manière aléatoire (sans duplication dans une même méthode) depuis la base de données. Af n d'éviter tout biais potentiel lié à l'apprentissage, les 3 méthodes de visualisation apparaissaient dans un ordre aléatoire (par exemple : 15 tracés en stéréoscopie suivis de 15 en perspective suivis de 15 en 2D). La durée moyenne d'une session était de 15 minutes (28 au maximum).

Avant le début de chaque session, quelques questions étaient posées aux participants af n de recueillir leurs degrés de familiarité avec la théorie des graphes, la visualisation de graphes, les aff chages stéréoscopiques et l'utilisation de logiciels 3D. Une f che de présentation expliquant la procédure à suivre était distribuée et une démonstration rapide du système était effectée. Cette démonstration présentait le système de visualisation (les interactions disponibles, la tâche à effectuer) ainsi que les trois méthodes de visualisation sur un graphe facilement appréhendable (3 communautés, 20 nœuds par communauté une important probabilité p_{int} et une faible probabilité p_{ext}). A la suite de la démonstration, les participants devaient compléter une session d'entrainement af n de se familiariser avec le système. Cet entrainement comprenait 3 tracés de complexité croissante par méthode de visualisation.

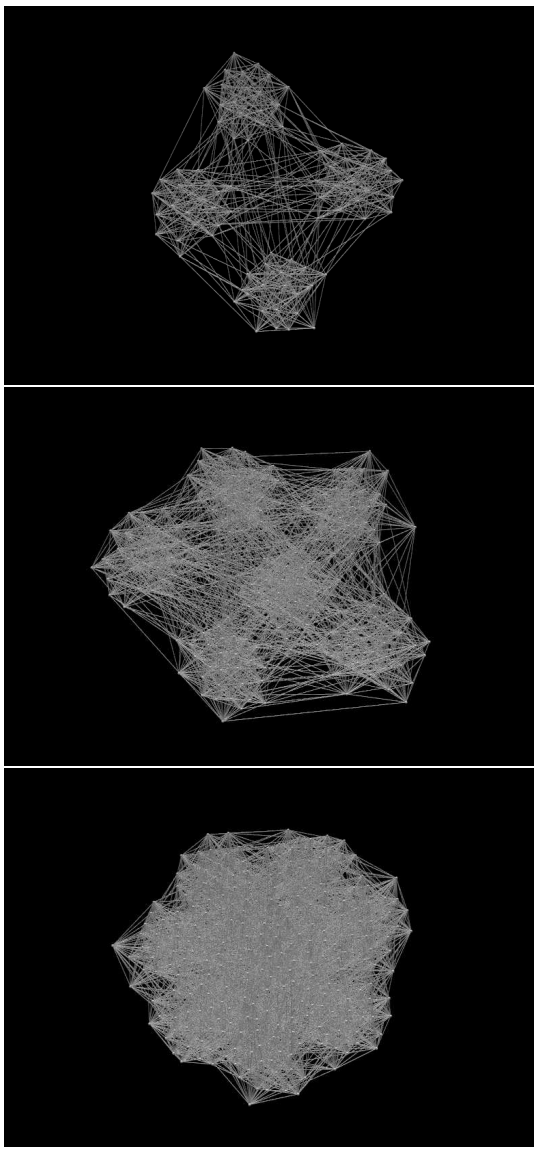


FIG. 2 – *Captures de tracés de graphes de complexité croissante.* $G_1 = G(k = 4, nv = 20, p_{ext} = 0.05, p_{int} = 0.8)$ *;* $G_2 = G(k = 6, nv = 30, p_{ext} = 0.05, p_{int} = 0.7)$ *;* $G_3 = G(k = 8, nv = 30, p_{ext} = 0.07, p_{int} = 0.8)$

Il était ainsi demandé aux participants d'estimer le nombre de communautés présentes dans les graphes aff chés le plus rapidement possible sans l'aide de l'expérimentateur. Si un participant n'arrivait pas à discerner les différentes communautés, la consigne était de passer au tracé suivant via la sélection du bouton "Je ne sais pas". Une fois tous les tracés d'une méthode de visualisation analysés, le participant passait à la série suivante. Cette transition était accompagnée par l'expérimentateur qui introduisait la série suivante ("Vous avez terminé la série XX, vous vous apprêtez à commencer la série YY). A la f n de la session, quelques questions étaient de nouveau posées aux participants af n d'estimer leurs préférences (méthode la plus facile et la plus diff cile) ainsi que leur impression de réussite (méthodes avec les meilleurs et moins bonnes performances).

4 Résultats

4.1 Qualité de la Détection de Communautés

Soit I l'ensemble des instances i présentés aux participants avec le même ratio p_{ext}/p_{int}, $k_{i,ans}$ le nombre de communautés proposé par un participant pour une instance i et k_i le nombre *a priori* de communautés dans le modèle. Pour chaque méthode de visualisation vm, l'erreur est mesurée par la moyenne des différences entre $k_{i,ans}$ et k_i :

$$Error_{vm}(I) = \frac{1}{card(I)} \sum_{i \in I} (|k_{i,ans} - k_i|)$$

Notons que la réponse "Je ne sais pas" était comptée comme une erreur (càd la détection d'une seule communauté), et que pour les valeurs les plus grandes de p_{ext} et les plus faibles de p_{int}, le terme "erreur" n'est pas parfait car la détection peut être très ambigüe. Mais dans ces cas, l'évaluation de la qualité de la détection de communautés reste une question délicate (cf. Fortunato et al, 2010). Notons que des travaux récents (Delest et al. (2006)) ont permis d'améliorer l'évaluation avec des mesures classiques telles que la modularité mais cette question dépasse largement le cadre de cet article.

Les résultats dépendent de la complexité de la structure du graphe mesurée ici par le ratio p_{ext}/p_{int} : pour de petites valeurs, les communautés sont facilement identif ables tandis que pour de plus grandes valeurs, d'importants chevauchements et des densités de communauté faibles rendent la détection délicate. Le tableau 1 montre que la 2D est signif cativement meilleure pour les complexités les plus faibles (Anova2 : $p = 0.01$) : les communautés sont bien séparées sur le tracé et facilement détectables sur un plan. Pour les complexités les plus importantes, la stéréoscopie est légèrement plus performante (Anova2 $p = 0.1$). Mais cette différence n'est signif cative que pour $k > 7$ et pour des complexités supérieures à 0.06 (Anova2 : $p = 0.02$), tel que nous le montre le tableau 2. La dimension perçue additionnelle combinée au mouvement semble aider à faire la distinction entre les différents aggrégats ; et ce, même en présence de "bruit" (chevauchements). La situation est cependant différente pour la 3D perspective pour laquelle l'occlusion explique certainement les différences avec la stéréoscopie. Quoi qu'il en soit, notons que, quelle que soit la méthode employée, la variation de l'erreur devient vite très importante dès la complexité croît (tableau 3). Nous avons observé que cette variation est indépendante de k et qu'elle ne peut pas, dans notre echantillon de participants,

TAB. 1 – *Erreur dans la détection de communautés. Pour chaque méthode de visualisation* vm *et pour chaque intervalle de complexité* (p_{ext}/p_{int}) I*, l'erreur moyenne* $Error_{vm}(I)$

Complexité (p_{ext}/p_{int})	2D	3D persp	3D stereo
$[0.02; 0.04]$	**0.10**	0.37	0.27
$]0.04; 0.06]$	1.62	1.64	**1.40**
$]0.06; 0.11]$	3.27	3.22	**2.78**
$]0.11; 0.15]$	3.47	3.71	**2.99**

être expliquée par l'absence de familiarité avec les logiciels 3D. Cependant des expériences complémentaires sont nécessaires avant de totalement rejeter cette hypothèse.

4.2 Temps de Réponse

Pour chaque méthode de visualisation, le temps de réponse est la moyenne des temps de réponse $Time_{vm}(I)$ des participants. Le tableau 4 nous montre que le temps de réponse de la 2D est signif cativement plus faible que pour la 3D (Anova2 : $p <= 0.001$), quelle que soit la complexité des graphes. De plus, les temps de réponse de la 3D perspective et de la 3D stéréoscopique sont très similaires. Cette différence entre 2D et 3D est probablement due à la nécessité d'intéragir avec les graphes en 3D af n de les analyser dans leur ensemble.

4.3 Perception des Participants

Le tableau 5 indique que les participants semblent préférer la stéréoscopie et semblent l'avoir trouvé plus facile à appréhender que les autres méthodes. Cependant, nous sommes conscients qu'un biais peut exister. En effet, l'expérience en elle-même illustre notre intérêt pour la 3D stéréoscopique, et les participants pouvaient être enclain (même inconsciement) à partager notre enthousiasme. Quoi qu'il en soit, une part de cette subjectivité est corroborée par les résultats de l'expérience. En effet, parmi les participants ayant indiqué avoir mieux réussi en 3D stéréo, 54% ont vu leur intuition conf rmée par les résultats (tandis que seulement 15.5% ont obtenu leurs meilleurs résultats avec la 3D perspective).

5 Conclusion

A notre connaissance, cet article présente une recherche pionnière sur l'utilisation de la stéréoscopie dans un problème de visualisation qui a connu un intérêt croissant durant la dernière décennie : la détection de communautés dans de grands graphes. Notre expérience révèle une différence importante entre la 3D stéréoscopique et la 3D perspective qui a été vivement critiquée par les communautés de Graph Drawing et de visualisation de réseaux sociaux. De plus, bien que le débat reste ouvert, nos résultats expérimentaux indiquent que la stéréoscopie, peut prévaloir sur la 2D pour les graphes aux structures complexes comportant de nombreuses

TAB. 2 – *Erreur dans la détection de communautés. Pour chaque méthode de visualisation vm et pour chaque intervale de complexité (p_{ext}/p_{int}) I, l'erreur moyenne $Error_{vm}(I)$ en fonction de k*

Complexité (p_{ext}/p_{int})	Méthode	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$	$k=11$
$[0.02; 0.04]$	2D	0.0	0.0	0.1	**0.05**	**0.06**	0.16	**0.13**	**0.67**
	3D persp	0.0	0.0	0.13	0.08	0.25	0.35	0.37	1.56
	3D stereo	0.0	0.0	**0.0**	0.07	0.16	**0.11**	0.42	1.67
$]0.04; 0.06]$	2D	0.0	0.14	**0.0**	**0.19**	1.27	2.65	**3.35**	6.94
	3D persp	0.0	0.0	0.08	0.58	1.45	1.92	4.13	5.93
	3D stereo	0.0	0.0	0.13	0.9	**0.63**	**0.69**	3.81	**5.5**
$]0.06; 0.11]$	2D	0.0	0.0	0.5	1.35	2.24	3.95	6.15	7.9
	3D persp	0.0	0.13	0.32	**1.07**	**1.63**	4.44	6.12	7.61
	3D stereo	0.0	0.0	**0.21**	1.6	1.9	**3.42**	**5**	**6.88**
$]0.11; 0.15]$	2D	**0.28**	**0.88**	**2**	**4.69**	6.14	7.13	9	10
	3D persp	0.73	1.23	2.3	5.81	5.44	**6**	8.75	8.75
	3D stereo	0.93	1.35	2.06	4.75	**5.25**	8	**6.6**	**6**

TAB. 3 – *Ecart type de l'erreur dans la détection de communautés en fonction de la complexité (p_{ext}/p_{int})*

Complexity (p_{ext}/p_{int})	2D	3D persp	3D stereo
< 0.06	1.28	1.16	1.67
≥ 0.06	3.63	3.69	3.38

TAB. 4 – *Temps de réponse moyen $Time_{vm}(I)$ en secondes, pour chaque méthode de visualisation vm et pour chaque intervale de complexité (p_{ext}/p_{int})*

Complexité (p_{ext}/p_{int})	2D	3D persp	3D stereo
$[0.02; 0.04]$	**7.3**	14.2	12.3
$]0.04; 0.06]$	**11.1**	17.9	17.7
$]0.06; 0.11]$	**12.4**	22.5	24.7
$]0.11; 0.15]$	**13.1**	21.9	21.1

TAB. 5 – *Perception subjective des participants. Pour chaque méthode de visualisation vm, pourcentage des participants ayant répondu que vm était le plus facile (resp. le plus difficile) et celle avec laquelle ils ont estimé avoir obtenu leurs meilleurs (resp. moins bonnes) performances.*

Réponse	2D	3D persp	3D stereo	NSP
Plus facile	14.2	0	**68.6**	17.2
Plus diff cile	37.1	43	**5.7**	14.2
Meilleurs performances	11.3	0	**74.3**	14.4
Moins bonnes performances	43	34.3	**5.7**	17

communautés de densités variables avec de nombreux chevauchements. Des études complémentaires sont toutefois nécessaires af n de conf rmer ces résultats sur des populations plus larges et pour pouvoir comprendre les différences observées.

Cet article reprend des résultats présentés à Graph Drawing 2011.

Références

Auber, D. ., Y. Chiricota, F. Jourdan, et G. Melancon (2003). Multiscale visualization of small world networks. In *INFOVIS '03 : Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'03)*, pp. 75–81.

Battista, G. D., M. Patrignani, et F. Vargiu (1998). A split&push approach to 3d orthogonal drawing. In *Proceedings of the 6th International Symposium on Graph Drawing*, GD '98, London, UK, pp. 87–101. Springer-Verlag.

Belcher, D., M. Billinghurst, S. Hayes, et R. Stiles (2003). Using augmented reality for visualizing complex graphs in three dimensions. In *ISMAR*, pp. 84–92.

Cutting, J. (1997). How the eye measures reality and virtual reality. *Behavior Research Methods, Instrumentation, and Computers 29*, 29–36.

Delest, M., J.-M. Fedou, et G. Melancon (2006). A quality measure for multi-level community structure. In *Proceedings of the Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, Washington, DC, USA, pp. 63–68. IEEE Computer Society.

Domini, F., C. Caudek, et H. Tassinari (2006). Stereo and motion information are not independently processed by the visual system. In *Vision Res.*, Volume 46, pp. 1707–23. Elsevier.

Eades, P., A. Symvonis, et S. Whitesides (2000). Three-dimensional orthogonal graph drawing algorithms. *Discrete Applied Mathematics 103*(1-3), 55–87.

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

Henry, N., A. Bezerianos, et J.-D. Fekete (2008). Improving the readability of clustered social networks using node duplications.

Hubona, G. S., P. N. Wheeler, G. W. Shirah, et M. Brandt (1999). The relative contributions of stereo, lighting, and background scenes in promoting 3d depth visualization. *ACM Trans. Comput.-Hum. Interact. 6*, 214–242.

Kolmogorov, A. et Y. Barzdin (1967). Abour realization of sets in 3-dimensional space, problems cybernet.

Landy, M. S., L. T. Maloney, et M. J. Young (1991). Psychophysical estimation of the human depth combination rule. Volume 1383, pp. 247–254. SPIE.

Rosenberg, A. L. (1983). Three-dimensional vlsi : a case study. *J. ACM 30*, 397–416.

Saracini, C., R. Franke, E. Blümel, et M. Belardinelli (2009). Comparing distance perception in different virtual environments. *Cognitive Processing 10*, 294–296. 10.1007/s10339-009-0314-7.

Teyseyre, A. et M. Campo (2009). An overview of 3d software visualization. *IEEE Trans. on Visualization and Computer Graphics 15*(1), 114–135.

van Schooten, B. W., E. M. A. G. van Dijk, E. Zudilova-Seinstra, A. Suinesiaputra, et J. H. C. Reiber (2010). The effect of stereoscopy and motion cues on 3d interpretation task performance. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10, New York, NY, USA, pp. 167–170. ACM.

Ware, C. et G. Franck (1996). Evaluating stereo and motion cues for visualizing information nets in three dimensions. In *ACM Transactions on Graphics*, Volume 15, pp. 121–139.

Ware, C. et P. Mitchell (2005). Reevaluating stereo and motion cues for visualizing graphs in three dimensions. In *Proceedings of the 2nd Symposium on Applied Perception in Graphics and Visualization, APGV 2005*, Volume 95. ACM.

Ware, C. et P. Mitchell (2008). Visualizing graphs in three dimensions. In *ACM Transactions on Applied Perception*, Volume 5, pp. 2–15.

Wood, D. R. (2003). Optimal three-dimensional orthogonal graph drawing in the general position model. *Theor. Comput. Sci. 299*, 151–178.

Summary

3D drawing problems of the 90Šs were essentially restricted on representations in 3D perspective. However, recent technologies offer 3D stereoscopic representations of high quality which allow the introduction of binocular disparities, which is one of the main depth perception cues, not provided by the 3D perspective. This paper explores the relevance of stereoscopy for the visual identif cation of communities, which is a task of great importance in the analysis of social networks. A user study conducted on 35 participants with graphs of various complexity shows that stereoscopy outperforms 3D perspective in the vast majority of the cases. When comparing stereoscopy with 2D layouts, the response time is signif cantly lower for 2D but the quality of the results closely depends on the graph complexity : for a large number of clusters and a high probability of cluster overlapping stereoscopy outperforms 2D whereas for simple structures 2D layouts are more eff cient.

Évaluation de ProxiViz pour la fouille visuelle de données multidimensionnelles

Nicolas Heulot*, Michaël Aupetit*
Jean-Daniel Fekete**

*CEA, LIST, Laboratoire Information Modèle et Apprentissage,
F-91191, Gif-sur-Yvette Cedex, France.
nicolas.heulot@cea.fr
**INRIA,
Bât 650, Université Paris-Sud, F-91405, Orsay Cedex, France.
jean-daniel.fekete@inira.fr

Résumé. Nous présentons les résultats d'une évaluation contrôlée de la technique de visualisation ProxiViz pour la fouille visuelle de données multidimensionnelles basée sur des projections non linéaires. ProxiViz affiche le nuage de points classique et y superpose l'information de distance d'origine entre un point sélectionné interactivement par l'utilisateur et les autres points. Nous comparons cette technique avec la visualisation de la projection seule et la visualisation des distorsions locales sur la projection, dans le contexte d'une tâche de classification visuelle des données. Nous évaluons différentes méthodes de représentation visuelle de ces informations de proximité et de distorsion : coloration des points projetés, coloration de leur cellule de Voronoï, et coloration interpolée en arrière-plan. Nous discutons finalement de l'influence des compressions et étirements de l'espace projeté sur l'utilisabilité de ces visualisations.

1 Introduction

Il existe un grand nombre d'outils et de techniques de visualisation conçus pour révéler les informations sous-jacentes à des données multidimensionnelles. La projection des données dans un espace à deux ou trois dimensions sous forme d'un nuage de points est une technique populaire dans le domaine de l'analyse exploratoire de données. On l'utilise principalement pour observer les groupes de données similaires (classes), déterminer leurs structures, leurs frontières, et détecter les données atypiques.

Nous présentons dans cet article les résultats d'une évaluation des performances de deux techniques de visualisation permettant la fouille visuelle *in situ* de projections de données, c'est à dire par l'ajout d'informations directement sur la projection : ProxiViz qui affiche les informations de proximité originelle, et la visualisation des informations de distorsions. On se place dans le cadre d'une tâche de classification visuelle pour évaluer si ces techniques améliorent la qualité d'analyse des projections dans une approche non supervisée. Nous discutons également le choix d'une méthode de projection adaptée à ces techniques en considérant l'influence du type de distorsions sur l'utilisabilité de ces visualisations.

Pour effectuer une projection de données de type individus/variables, on doit dans un premier temps définir une mesure de distance entre individus (comme la distance euclidienne ou la distance géodésique). Il existe ensuite une multitude de méthodes de projections linéaires ou non linéaires. Les méthodes non linéaires en particulier optimisent différentes fonctions objectifs de telle sorte que les distances entre individus dans l'espace d'origine soient au mieux préservées dans l'espace d'arrivée : deux points proches (ou éloignés) sur la projection doivent correspondre à deux individus proches (ou éloignés) dans l'espace de départ.

Mais ces méthodes de projection sont très dépendantes de la topologie de la structure sous-jacente aux données et introduisent des distorsions. Ces distorsions peuvent se caractériser par des erreurs de distances ou des erreurs de voisinages. On en distingue quatre types, d'après la terminologie de Aupetit (2007) : compression/étirement (la distance entre deux points sur la projection est plus faible/importante que la distance entre leurs deux individus d'origine) et recollement/déchirure (deux points voisins/non voisins sur la projection correspondent à deux individus non voisins/voisins dans l'espace de départ). Si on peut réduire le nombre de distorsions ou n'en favoriser qu'un type, on ne peut cependant pas en faire abstraction car elles nuisent à la lecture et à l'interprétation d'une projection. Pour inférer la structure originelle des données *in situ*, c'est à dire directement à partir de la visualisation de la projection, on peut soit visualiser les informations de distorsions, soit visualiser les informations de proximités originelles, comme cela est proposé dans Aupetit (2010).

Dans notre évaluation, nous comparons trois visualisations : la projection avec les informations de distorsions locales, la projection avec l'affichage interactif des informations de distances d'origine relativement à un individu sélectionné par l'utilisateur (ProxiViz), et la projection sans autres informations. Nous considérons trois représentations de ces informations directement sur la visualisation de la projection par le biais d'une échelle de couleur : coloration des points, coloration des cellules de Voronoï, coloration interpolée en arrière plan. Nous comparons également deux types de projections : l'une favorisant les déchirures et l'autre introduisant principalement des recollements.

Nous traiterons dans un premier temps du choix de ces techniques ainsi que des détails de l'évaluation. Puis dans un second temps, nous présenterons et discuterons les résultats de l'évaluation des performances de deux techniques en termes de temps et de nombre d'erreurs, dans le cadre d'une tâche consistant à retrouver le nombre de classes présentes dans les données d'origine à partir de la projection. Nous discuterons enfin l'intérêt, en termes d'utilisabilité, des méthodes de projection non linéaire favorisant les distorsions de type déchirures plutôt que les recollements lors de l'utilisation de ces visualisations.

2 Fouille visuelle de données basée sur des projections

La projection de données, linéaire ou non linéaire, implique une perte d'information. Pour pouvoir tirer des conclusions d'une projection de données en deux dimensions, l'expert doit pouvoir vérifier si les informations représentées sur la projection sont fidèles aux informations originelles. Il existe différentes techniques de visualisation et métriques pour évaluer la qualité d'une projection, la plus standard étant la comparaison des distances euclidiennes $d^*_{i,j}$ dans l'espace de grandes dimensions avec les distances $d_{i,j}$ dans l'espace de faibles dimensions. Nous présentons dans cette partie les techniques de visualisation statiques et interactives utilisées pour l'inférence de la structure sous-jacente aux données à partir d'une projection.

2.1 Visualisation statique

Dans Kruskal (1964), il est proposé d'effectuer cette comparaison par le calcul d'une mesure de stress globale correspondant aux résidus de la somme des carrés des distances : $S = \sum_{i,j} (d_{i,j} - d^*_{i,j})^2$. Kruskal propose également de comparer visuellement les distances en utilisant le diagramme de Shepard. Les distances dans l'espace d'arrivée sont visualisées par rapport aux distances dans l'espace multidimensionnel de départ sous forme d'un nuage de points. La proximité des points par rapport à la droite $y = x$ mesure la qualité de la projection en termes de conservation des distances. Cette technique nécessite cependant de faire visuellement le lien entre deux vues : la représentation de la projection et la représentation du diagramme.

Nous nous intéressons ici uniquement à des techniques de visualisation *in situ*, c'est à dire affichant directement les informations sur la projection. Une technique fréquemment utilisée sur les projections de données linéaires consiste à afficher les mesures de stress en chaque point directement sur la projection par le biais de la couleur. L'utilisabilité de la visualisation du stress dépend directement du choix de la mesure de stress. Il existe une multitude de critères et de mesures pour évaluer le stress d'une projection de manière globale ou locale (voir Venna (2007)).

Dans Seifert et al. (2010), une mesure locale du stress permet de prendre en compte les distorsions de types recollements et déchirures. Pour visualiser ce stress, les auteurs proposent d'utiliser une Stress Map qui se base sur la métaphore de carte géographique utilisée pour la visualisation de corpus de textes (Information Landscape). Les montagnes (ou îles) représentent des groupes de documents similaires de forte densité et sont séparés par des plaines (ou mers). La hauteur des montagnes représente la densité locale des points alors que leur largeur représente la cohésion de ceux ci. Le stress est ensuite représenté avec une échelle de couleur par coloration interpolée de l'arrière plan (Heat Map).

Dans Brodbeck et al. (1997), le stress est représenté autour de chaque point par un disque plein lui même entouré d'un cercle. La largeur du disque représente la valeur du stress et la largeur du cercle concentrique représente le maximum de variation de la position du point par rapport à sa position initiale lors de l'exécution de l'algorithme de projection. Cette technique est très sujette aux problèmes d'occlusions entre points.

Dans Lespinats et Aupetit (2011), les auteurs proposent de visualiser conjointement deux mesures de stress, chacune favorisant la prise en compte d'un type de distorsions (recollements S_1 ou déchirures S_2) :

$$\begin{array}{l} S_1(i) = \sum_j |d_{i,j} - d^*_{i,j}|^2 \times F(d_{i,j}, \sigma) \\ S_2(i) = \sum_j |d_{i,j} - d^*_{i,j}|^2 \times F(d^*_{i,j}, \sigma) \end{array} \text{, avec } F(x,\sigma) = \left\{ \begin{array}{ll} 1, & \text{si } x \leq \sigma \\ 0, & \text{si } x > \sigma \end{array} \right.$$

Une règle arbitraire est proposée pour le choix du rayon de voisinage σ consistant à prendre la moyenne des distances entre tous les points et leur 5ème plus proche voisin dans l'espace d'origine. Ces deux mesures de stress sont représentées par le biais d'une échelle de couleur perceptuellement uniforme à deux dimensions. Les auteurs proposent deux règles d'inférence permettant d'interpréter les contrastes de couleur obtenus et ainsi de pouvoir faire des inférences sur la structure originelle des données.

2.2 Visualisation interactive

Dans Schreck et al. (2010), une mesure locale du stress appelée mesure de précision est proposée. Cette mesure correspond pour un point donné à la norme de la différence entre le vecteur normalisé des distances à ce point dans l'espace d'origine et celui des distances sur la projection. Cette mesure prend en compte un paramètre de localité qui peut être fixé interactivement par l'utilisateur pour définir la taille du voisinage à utiliser dans l'espace d'origine. Les auteurs proposent de visualiser le stress par une *Precision Map*, où la précision est représentée par coloration de l'arrière plan en utilisant une fonction d'interpolation avec un paramètre de lissage.

Une autre technique utilisée avec des algorithmes de projection basés sur les distances géodésiques consiste à afficher les liens entre voisins directs dans l'espace d'origine. Mais elle a pour principal inconvénient d'être sujette aux problèmes de croisement et d'occlusions que l'on retrouve dans les graphes denses. Cette technique est également utilisée de manière interactive dans l'outil d'exploration ProjEx de Paulovich et al. (2007), pour visualiser les voisins dans l'espace d'origine ou dans l'espace d'arrivée.

Dans Aupetit (2007), les distances d'origine relatives à un point de référence sélectionné par l'utilisateur sont visualisées interactivement par la coloration des cellules de Voronoï de chaque point de la projection. Ceci permet non seulement de visualiser les distorsions de projection, mais surtout de détecter visuellement les zones délimitant les classes dans l'espace d'origine.

3 Choix de conception de l'évaluation

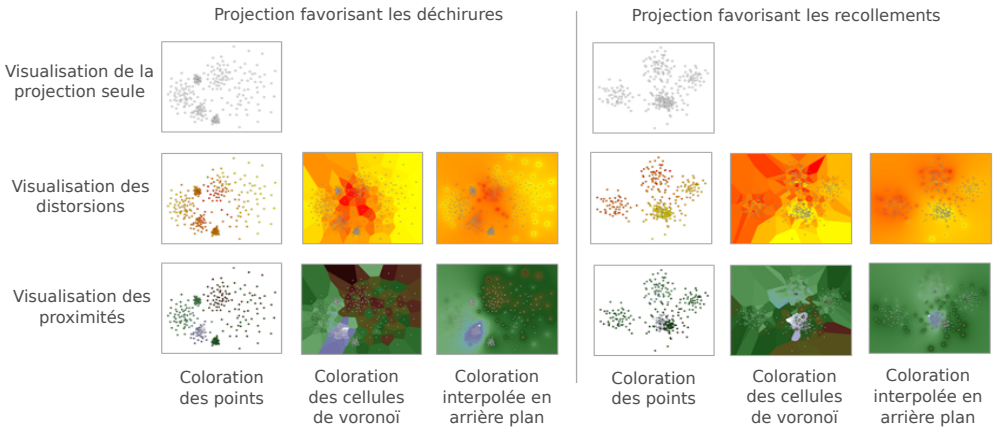


FIG. 1 – *Tableau des différents facteurs : information visualisée, coloration, type de projection (les images correspondent aux différentes visualisations d'un même jeu de données jouet utilisé dans l'évaluation).*

3.1 Jeux de données

Pour assurer la diversité des jeux de données, nous utilisons des jeux de données réels et des jeux de données jouets. Chaque jeu de données est composé d'un nombre fixe de classes de premier niveau, suffisamment séparées et homogènes pour minimiser toute ambiguïté dans le cadre de la tâche de classification visuelle. Le nombre de classes p varie entre 2 et 7 et la dimension d entre 5 et 960 pour des populations aléatoires n allant de 141 à 365 individus.

Les jeux de données jouets sont générés aléatoirement. On tire aléatoirement p centroïdes de dimension d selon une loi normale centrée réduite. Chaque centroïde est ensuite utilisé comme moyenne d'une loi normale dont l'écart type est calculé en fonction de la moitié de la distance minimum aux autres centroïdes, de manière à réduire les chevauchements entre classes. Enfin on tire aléatoirement n individus selon chaque loi. Nous avons ensuite sélectionné trois jeux de données jouets selon le niveau de séparation des points sur la projection, de manière à varier la difficulté de la classification visuelle.

Nous utilisons comme jeux de données réels des échantillons issus des jeux de données : Letter Recognition de dimensions 16 (les lettres A, E, I, O, U avec 50 individus choisis aléatoirement) et CMU Face Images de dimensions 960 (les personnes an2i/at33/saavik/steffi avec les images 32x30 selon les critères left/right/straight/up, angry/happy/neutral/sad et open/sunglasses) provenant de la base de données de Frank et Asuncion (2010). Nous avons essayé de choisir des jeux de données de difficultés différentes avec des classes relativement bien séparées. Pour ajouter des aspects topologiques à la classification visuelle, nous utilisons le jeu de données Teapot proposé dans Zhu et Lafferty (2005) composé de deux variétés.

3.2 Algorithme de projection

Nous cherchons à déterminer si le type de distorsions présentes sur la projection a une influence sur les performances des différentes visualisations permettant la fouille visuelle de projections. Nous utilisons l'algorithme Local Multidimensional Scaling Chen et Buja (2009) qui permet grâce à un paramètre d'orienter la fonction d'optimisation soit vers la fonction objectif de la projection non linéaire de Sammon (1969), soit vers celle de l'Analyse en Composantes Curvilignes de Demartines et Hérault (1997). La première fonction pénalise les déchirures et favorise les recollements. La seconde a l'effet inverse en pénalisant principalement les recollements et en favorisant les déchirures. L'algorithme LMDS permet donc de trouver un compromis entre déchirures et recollements, comme le montre sa fonction objectif :

$$E = \frac{1}{2} \sum_i \sum_{j \neq i} (d_{i,j} - d^*_{i,j})^2 [\lambda F(d_{i,j}, \sigma) + (1 - \lambda) F(d^*_{i,j}, \sigma)]$$

Nous supposons que les déchirures se prêtent mieux à l'utilisation de ProxiViz et de la visualisation des distorsions que les recollements. Nous avons donc choisi les deux configurations extrêmes de l'algorithme pour construire des projections composées principalement soit de déchirures ($\lambda = 0$), soit de recollements ($\lambda = 1$). Le paramètre de voisinage σ tend vers zéro au cours de l'optimisation qui consiste en une descente de gradient stochastique. Pour définir la mesure de stress représentée par la visualisation des distorsions, nous utilisons le résultat de la fonction objectif en fixant arbitrairement le paramètre σ comme la moyenne des distances minimums dans l'espace d'origine.

3.3 Échelles de couleurs

Pour la visualisation des distorsions, nous avons choisi une échelle de couleurs allant du jaune au rouge, comme proposé dans Bentley et Ward (1996). La couleur rouge indique le maximum de stress, c'est à dire le maximum de risque de mauvaise interprétation de la position des points, et la couleur jaune indique le minimum de stress, c'est à dire le minimum de risque de mauvais positionnement des points les uns par rapport aux autres. Pour ProxiViz, nous utilisons l'échelle de couleurs introduite dans Tominski et al. (2008), adaptée à une tâche de comparaison visuelle, pour mieux différencier les zones frontières entre classes. Cette échelle de couleurs va du blanc, indiquant la distance nulle, vers le noir indiquant le maximum de distance par rapport au point de référence, en passant d'abord par des nuances de violet puis de vert.

Dans le cadre de ProxiViz, lorsque l'utilisateur déplace la souris au dessus de la visualisation de la projection, l'affichage des couleurs est instantané sur les points, les cellules de Voronoï, ou en arrière plan. Nous pré-calculons le découpage en cellules de Voronoï de chaque projection. Pour afficher les couleurs en arrière plan, nous utilisons une interpolation de Shepard : la couleur $u(x)$ d'un pixel x est obtenue par pondération de la couleur u_i de chaque point de la projection :

$$u(x) = \sum_{i=0}^{N} \frac{w_i(x) u_i}{\sum_{j=0}^{N} w_j(x)} \text{ , avec } w_i(x) = \frac{1}{d(x, x_i)^p}$$

On fixe arbitrairement le facteur de voisinage $p = 2$ pour préserver au mieux les informations locales. Chaque visualisation est affichée en plein écran sur un écran LCD HP Compaq LA2205wg de 22 pouces avec une résolution de 1680x1050 pixels.

4 Expérience

4.1 Hypothèses

Basé sur notre expérience, nous supposons que ProxiViz est plus efficace en termes d'erreurs que la visualisation des distorsions ou la visualisation de la projection seule, pour retrouver les informations de classification dans les données. Nous avons également l'intuition que les projections favorisant les déchirures plutôt que les recollements sont plus faciles à analyser avec ces outils.

- Hypothèse 1 : En termes de temps, ProxiViz est plus lente que la visualisation des distorsions et la visualisation de la projection seule du fait de son interactivité. Mais en termes d'erreurs, elle est plus performante que les deux autres car elle est directement interprétable pour inférer des informations sur la structure des données.
- Hypothèse 2 : Pour chaque visualisation, il n'y a pas de différence en termes de temps et d'erreurs entre la coloration des points, la coloration des cellules de Voronoï ou la coloration interpolée en arrière plan.
- Hypothèse 3 : Pour chaque visualisation, les projections composées principalement de déchirures sont plus performantes en termes de temps et d'erreurs que les projections composées de recollements.

4.2 Conception

Chaque participant a été confronté successivement à 7 séquences de 12 jeux de test (sans répétition des mesures). L'ordre des séquences et des jeux de test est aléatoire. Pour empêcher l'apprentissage sur les jeux de données, nous avons reprojeté les jeux données selon les deux configurations de l'algorithme de projection pour chaque séquence. Entre chaque séquence, les participants choisissaient de continuer quand ils étaient prêts, et ils pouvaient stopper à chaque instant au cours d'un jeu de test. Nous avons imposé une durée de 30 secondes à 1 minutes maximum pour chaque jeu de test, pour une durée d'évaluation totale de 40 minutes environ, précédée de 20 minutes maximum de présentation de l'évaluation et d'entraînement sur 7 jeux de données jouets (un par combinaison de visualisation et de coloration) pour apprendre à interpréter les visualisations.

4.3 Participants

21 participants (14 hommes et 7 femmes), âgés de 21 à 35 ans (moyenne de 28 ans), provenant de deux laboratoires spécialisés en analyse de données ont été recrutés. Tous les participants, dont 6 étudiants, sont scientifiques. Ils ont tous déclaré avoir une vision normale sans problèmes de vision des couleurs. Seul 5 participants ont dit ne jamais avoir utilisé de projections de données, et les autres y ont recours plusieurs fois par mois.

4.4 Tâche

Les participants devaient répondre à la question : « combien de classes comptez-vous ? », et valider à l'aide de la souris une des réponses possibles (de 1 et 8 classes). Ils devaient considérer des classes contenant au moins 10 points, et la réponse 1 classe signifiait qu'il leur était impossible de distinguer des classes différentes sur la visulisation. Cette tâche a été choisie pour mettre les participants dans le contexte d'une tâche de classification visuelle. Nous avons choisi le décompte des classes plutôt que la construction des classes par sélection à la souris pour mieux contrôler la durée de l'expérience. Le temps pour valider une réponse est enregistré (en secondes) pour chaque jeu de test, ainsi que la réponse sélectionnée. On calcule l'erreur comme la différence entre le nombre de classes trouvés p et le nombre de classes réellement présentes dans les données p^*. On en déduit un pourcentage de validité :

$$v = \begin{cases} \frac{|p-p^*|}{8-p^*} \times 100, \text{ si } p > p^* \\ \frac{|p-p^*|}{1-p^*} \times 100, \text{ si } p \leq p^* \end{cases} \quad (1)$$

5 Résultats

Après une analyse préliminaire des résultats, nous pouvons remarquer que les distributions des erreurs et du temps ne suivent pas des lois normales. On utilise la transformation log sur les mesures du temps pour se rapprocher de la loi normale. Pour analyser la variance, on utilise la méthode non paramétrique Kruskal-Wallis. On remarque, pour les mesures de temps et de validité (1), des effets importants ($p < 2,2.10^{-16}$) pour chaque facteur : information affichée, coloration, type de projection. On utilise le test de Wilcoxon pour comparer les distributions deux-à-deux.

5.1 Effet des informations affichées et de la coloration

Les participants ont réalisé le moins d'erreurs en utilisant ProxiViz (avec de 84,3% à 88,1% de validité des réponses), mais ils ont également été les plus lents avec cette même visualisation. On remarque également que la visualisation des distorsions introduit le plus d'erreurs et que la visualisation seule reste la plus rapide. En ce qui concerne les différentes colorations, on remarque que la visualisation utilisant la coloration des cellules de Voronoï est la plus lente, suivi de la coloration interpolée en arrière plan. Les participants ont réalisé les meilleurs résultats avec ProxiViz utilisant la coloration interpolée en arrière plan (2).

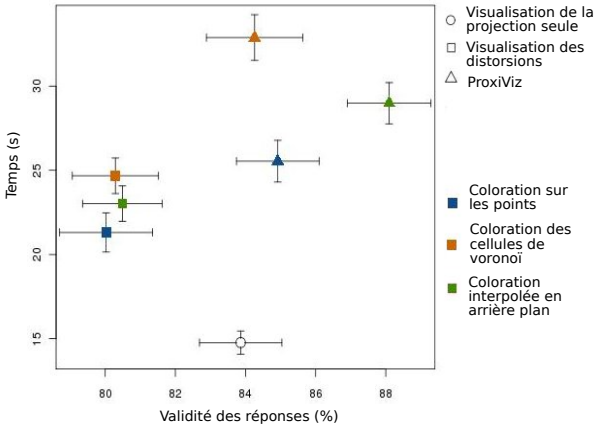


FIG. 2 – *Moyennes des temps de manipulation par rapport aux moyennes des pourcentages de validité (1) pour chaque visualisation (les barres d'erreur représentent l'erreur type).*

En termes de pourcentage de validité, les tests *post hoc* ont révélé que ProxiViz est significativement meilleure que la visualisation des distorsions ($p < 4,8.10^{-12}$) et que la projection seule ($p < 0,025$). La projection seule est significativement meilleure que la visualisation des distorsions ($p < 0,005$). ProxiViz est significativement plus lente que la visualisation des distorsions ($p < 9,7.10^{-11}$) et de la projection seule ($p < 6,9.10^{-33}$). La projection seule est significativement plus rapide que la visualisation des distorsions ($p < 2,69.10^{-15}$).

Concernant la coloration, les tests ont montré que ProxiViz avec la coloration interpolée en arrière plan était significativement meilleure en moyennes des pourcentages de validité que ProxiViz avec la coloration des cellules de Voronoï ($p < 0,05$), et avec la coloration des points ($p < 0,009$). En revanche il n'y a pas de différences significatives entre ProxiViz sur les points et ProxiViz appliqué aux cellules de Voronoï ($p > 0,5$). De même, il n'y a pas de différences significatives entre les différentes colorations pour la visualisation des distorsions. En termes de temps de réponse, ProxiViz avec la coloration des points est meilleure que la coloration des cellules de Voronoï ($p < 3,1.10^{-05}$) et la coloration interpolée ($p < 0,018$), laquelle est meilleure que la coloration des cellules de Voronoï ($p < 0,046$). De même pour la visualisation des distorsions, la coloration des points est meilleure que la coloration des cellules de Voronoï ($p < 0,003$), et la coloration interpolée ($p < 0,05$). En revanche il n'y a pas de différence significative entre la coloration des cellules de Voronoï et la coloration interpolée ($p > 0,2$).

5.2 Effet du type de distorsions

Pour chaque visualisation, les participants ont réalisé moins d'erreurs sur les projections favorisant les déchirures aux recollements. De même, ils ont été plus rapides en moyenne sur les projections dont les distorsions étaient principalement des déchirures (3).

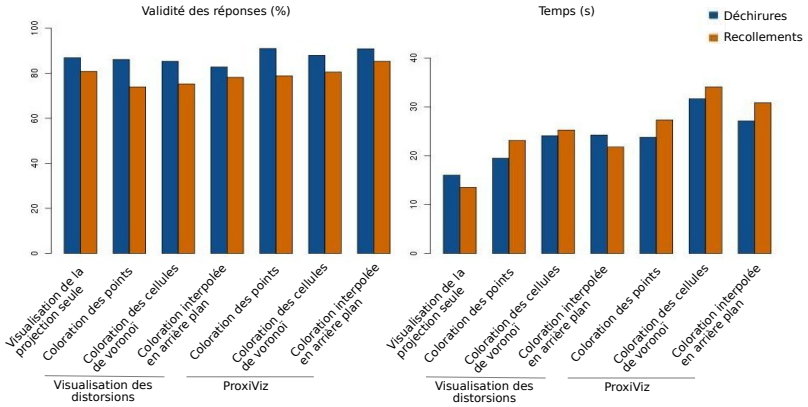


FIG. 3 – *Moyennes des temps de manipulation par rapport aux moyennes des pourcentages de validité (1) pour chaque visualisation*

Les projections favorisant les déchirures sont significativement meilleures en termes de moyennes de pourcentages de validité que les projections avec principalement des recollements ($p < 1,17.10^{-25}$). Plus précisement, en comparant pour chaque visualisation, les projections favorisant les déchirures sont significativement meilleures que celles introduisant principalement des recollements ($2,3.10^{-9} < p < 0,04$). Cependant pour les temps de réponse, les projections favorisant les déchirures sont meilleures que les projections introduisant des recollements uniquement pour la visualisation ProxiViz ($p < 0,014$).

5.3 Informations qualitatives

Chaque participant a rempli un questionnaire après l'évaluation pour donner ses préférences sur les différentes visualisations en les classant sur une échelle Likert de 1 (fortement en désaccord) à 7 (fortement d'accord) en fonction de différents critères.

5.3.1 Préférences générales

Les participants ont majoritairement préféré ProxiViz avec la couleur interpolée en arrière plan (11 participants). Ils ont principalement avancé les raisons esthétiques, la simplicité d'interprétation, et l'interactivité pour motiver leur choix. L'information de proximité a été jugée plus facile à interpréter et plus rassurante que l'information de distorsions. ProxiViz avec la couleur sur les cellules de Voronoï est la deuxième technique préférée (7 participants). Elle a principalement été choisie parce que les cellules de Voronoï permettaient de mieux discriminer les frontières entre zones de couleurs différentes. Enfin deux participants ont préféré ProxiViz avec la couleur sur les points et un participant a choisi la visualisation des distorsions avec la couleur interpolée en arrière plan.

5.3.2 Préférences en rapidité, précision, facilité d'utilisation et esthétique

Les participants se sont sentis rapides et précis avec ProxiViz utilisant la coloration interpolée et la coloration des cellules de Voronoï, et avec la projection seule (médiane ≥ 6). La visualisation ProxiViz avec la coloration interpolée a été jugée la plus esthétique et facile à utiliser (médiane $= 7$), devant ProxiViz avec la coloration des cellules de Voronoï (médiane $= 6$). Les participants ont classé chaque visualisation selon que les points sur la projection sont correctement séparés, modérément séparés, ou pas du tout séparés (nuage de bruit). Dans chaque cas, ProxiViz avec la coloration interpolée et la coloration des cellules de Voronoï a obtenu les meilleurs résultats (médianes du rang ≥ 2).

6 Discussion

De manière générale, nous avons validé nos hypothèses. Pour l'hypothèse 1, ProxiViz est plus performante que la visualisation des distorsions ou la visualisation de la projection seule. La tâche de comptage du nombre de clusters est une tâche de haut niveau nécessitant de l'inférence à partir des informations affichées sur la projection. Ceci peut expliquer le fait que la visualisation de la projection seule soit plus rapide que ProxiViz et que la visualisation des distorsions, car il y a plus d'informations à analyser sur ces deux dernières visualisations. Mais cela peut également expliquer le peu d'écart en termes de temps entre ProxiViz qui est interactive et la visualisation des distorsions qui est statique, car le travail d'inférence avec les informations de distorsions est beaucoup plus difficile, voir parfois impossible.

Contrairement à notre hypothèse 2, nous avons observé des différences significatives en termes de temps et d'erreurs entre les différentes colorations pour ProxiViz et pour la visualisation des distorsions. ProxiViz avec la coloration interpolée donne de meilleurs résultats en temps et en erreurs que les autres colorations. Ces résultats sont également confirmés par les informations qualitatives. On peut donc préconiser l'utilisation de ProxiViz avec la coloration interpolée en arrière plan pour la classification visuelle de données basée sur des projections. On nuancera cependant ce choix par le fait qu'il faut fixer un paramètre de voisinage pour l'interpolation contrairement à la coloration des cellules de Voronoï.

Nous avons validé l'existence d'une différence significative en termes d'erreurs entre les deux types de projections pour chaque visualisation, qui est également valable pour ProxiViz en termes de temps. Nous constatons que les projections favorisant des déchirures sont plus efficaces que celles favorisant des recollements. Ceci peut s'expliquer par le fait que lors de l'analyse de la projection, on interprète d'abord les informations locales (proximité entre points sur la projection par exemple), avant de traiter les informations à l'échelle globale (éloignement entre points par exemple). De ce fait, on a d'abord intérêt à ce que la projection soit localement correcte (i.e. absence de recollements), avant qu'elle soit fausse entre différentes régions (i.e. présence de déchirures). De plus, lorsque deux points A et B sont proches d'un point C, on peut inférer que A et B sont proches l'un de l'autre. En revanche, on ne peut rien inférer si A et B sont éloignés de C. Donc on peut inférer plus d'informations sur les proximités originelles lorsque l'on assure au mieux que des points proches sur la projection soient effectivement proches à l'origine, c'est à dire en évitant les recollements. On peut donc recommander l'utilisation de ProxiViz avec des algorithmes de projection favorisant des déchirures plutôt que des recollements.

7 Conclusion

Nous avons présenté les résultats d'une évaluation comparant ProxiViz, la visualisation des distorsions et la visualisation de la projection seule, dans le cadre de la fouille visuelle de données multidimensionnelles. ProxiViz ressort comme la méthode la plus performante en termes d'erreurs de classification visuelle. Parmi les différentes colorations possibles de ces visualisations pour représenter respectivement les informations de proximités ou de distorsions, la coloration interpolée en arrière plan a obtenu les meilleurs résultats en comparaison avec la coloration des cellules de Voronoï et la coloration des points. En termes d'erreurs et de temps de réponse, les projections favorisant les déchirures aux recollements sont les plus performantes. Nous recommandons donc l'utilisation de ProxiViz avec la coloration interpolée en arrière plan sur des projections favorisant plutôt les déchirures que les recollements pour la fouille visuelle de données multidimensionnelles. Pour améliorer les performances de ProxiViz, nous envisageons maintenant de tester d'autres méthodes d'interpolation de la couleur, comme l'interpolation des voisins naturels de Sibson, et d'ajouter des informations d'historique d'exploration pour tracer et faciliter son utilisation. L'information de proximité étant générique, nous envisageons d'appliquer le principe de ProxiViz à d'autres techniques de visualisation comme les *TreeMap*.

Remerciements

Nous remercions les relecteurs anonymes pour leurs remarques pertinentes. Cette recherche a été réalisée dans le cadre d'une thèse financée par une collaboration CEA/DGA.

Références

Aupetit, M. (2007). Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing 70*(7-9), 1304–1330.

Aupetit, M. (2010). Winsitu, un nouveau paradigme pour l'analyse exploratoire de donnees basee sur des projections. *Revue des Nouvelles Technologies de l'Information (RNTI A3)*, 79–98.

Bentley, C. et M. Ward (1996). Animating multidimensional scaling to visualize n-dimensional data sets. In *Proceedings of the 1996 IEEE Symposium on Information Visualization*, pp. 72 –73.

Brodbeck, D., M. Chalmers, A. Lunzer, et P. Cotture (1997). Domesticating bead : adapting an information visualization system to a financial institution. In *Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)*, pp. 73–80.

Chen, L. et A. Buja (2009). Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association 104*(485), 209–219.

Demartines, P. et J. Hérault (1997). Curvilinear component analysis : A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks 8*(1), 148–154.

Frank, A. et A. Asuncion (2010). UCI Machine Learning Repository. Irvine, CA : University of California, School of Information and Computer Sciences. http ://archive.ics.uci.edu/ml/datasets/.

Kruskal, J. (1964). Multidimensional sclaing by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1–27.

Lespinats, S. et M. Aupetit (2011). CheckViz : Sanity Check and Topological Clues for Linear and Non-Linear Mappings. In *Computer Graphics Forum*, Volume 30, pp. 113–125.

Paulovich, F., M. Oliveira, et R. Minghim (2007). The projection explorer : A flexible tool for projection-based multidimensional visualization. In *Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing*, pp. 27–36.

Sammon, J. (1969). A nonlinear mapping for data structure analysis. *IEEE Trans. Comput. C-18*(5), 401–409.

Schreck, T., T. von Landesberger, et S. Bremm (2010). Techniques for precision-based visual analysis of projected data. *Information Visualization 9*(3), 181–193.

Seifert, C., V. Saboland, et W. Kienreich (2010). Stress Maps : Analysing Local Phenomena in Dimensionality Reduction Based Visualisations. In *Proceedings of the 1st European Symposium on Visual Analytics Science and Technology (EuroVAST'10), Bordeaux, France*.

Tominski, C., G. Fuchs, et H. Schumann (2008). Task-driven color coding. In *Proceedings of the 12th International Conference on Information Visualisation (IV08)*, pp. 373–380.

Venna, J. (2007). Dimensionality reduction for visual exploration of similarity structures. *Helsinki University of Technology, Dissertations in Computer and Information Science Report D20*.

Zhu, X. et J. Lafferty (2005). Harmonic mixtures : combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pp. 1052–1059.

Summary

Projection algorithms like multidimensional scaling are often used to visualize high dimensional data. However, as soon as we attempt to make inference about the original data from the resulting projection visualizations, we face a problem of information loss and visualization reliability with the apparition of distortions like tears and false neighborhoods. We may use different visualization techniques to handle these issues, like color encoding of the distortion information or using a proximity-based visualization like ProxiViz to get true insights about the original data. In this work, we evaluate the cluster identification performances of these techniques. Multidimensional data are visualized using a projection visualization either without additional information, or with overload of local stress or the original distances using color on the dots, on their Voronoi cells or using the Shepard interpolation to color the background. We also compare two projection algorithms, one which is prone to tears and the other one to false neighborhoods. Statistically significant results lead us to provide guidelines about the use of projection visualizations in the context of visual analysis. For instance, the use of ProxiViz with the Shepard interpolation and a projection algorithm prone to tears appears to be a suitable combination for cluster identification.

Clustering multi-niveaux de graphes : hiérarchique et topologique

Nhat-Quang Doan*, Hanane Azzag*, Mustapha Lebbah *

* Université Paris 13, LIPN UMR 7030
99, avenue Jean-Baptiste Clément
93430 Villetaneuse, France
Nhat-Quang.Doan, hanane.azzag, lebbah@lipn.univ-paris13.fr

Résumé. Nous présentons dans cet article une approche nommée Gr-SOtree pour la classification de données structurées en graphes. Cette méthode a l'avantage de proposer une décomposition du graphe dans un nouvel espace de représentation en fournissant une organisation multi-niveaux : topologique et hiérarchique afin de faciliter l'interprétation des relations intrinsèques présentent dans le graphe. Nous évaluerons les capacités et les performances de notre approche sur des graphes de difficultés variables. Des résultats numériques et visuels seront présentes et discutés.

1 Introduction

Actuellement nous rencontrons de plus en plus de données structurées sous forme de graphes. La fouille de graphes est devenue une problématique de recherche intéressante et un défi réel en matière de fouille de données. Les méthodes de visualisation et de classification permettent d'aider à la compréhension des données structurées et particulièrement celles présentées sous forme de graphe. Le cas des grands graphes représente une problématique à part entière dans le domaine de l'apprentissage.

Le but de ce travail est de présenter un algorithme de classification qui permet de visualiser et de décomposer le graphe en plusieurs sous-arbres organisés sur une carte 2D. De manière plus générale, un graphe simple (ou non orienté) est formé par un ensemble d'arêtes qui représentent des connections entre des paires de sommets (ou nœuds). La fouille de graphes consiste à extraire l'information utile se trouvant dans la structure formée par les arêtes et les sommets. Actuellement les algorithmes de classification standards effectuent cette tâche de manière simpliste sans prendre en compte la topologie qui existe entre les sommets.

Dans ce travail, nous souhaitons fournir une décomposition multi-niveaux du graphe d'origine : hiérarchique et topologique. Les modèles topologiques sont souvent utilisés pour la visualisation et la classification non supervisée. Des extensions et des reformulations du modèle SOM ont été décrites dans Bishop et al. (1998) Fabrice Rossi (2010) Barbara Hammer (2009). Ces approches sont différentes les unes des autres, mais partagent la même idée de représenter de grands ensembles de données par une relation géométrique projetée sur une carte topologique 2D. Habituellement, nous disposons d'algorithme offrant une classification topologique

ou encore uniquement hiérarchique Diday et Simon (1976). Dans la littérature il existe peu d'algorithmes offrant la possibilité d'avoir en une seule passe une classification multi-niveaux. Les seuls travaux similaire sont ceux de SOM hiérarchique de Dittenbach et al. (2001) où les auteurs présentent une organisation multi-niveaux sur des référents de la carte en proposant ainsi plusieurs niveaux de cartes.

Certaines méthodes hiérarchiques basées sur la carte ont également été proposées par exemples TS-SOM Koikkalainen et Horppu (2007), GH-SOM Dittenbach et al. (2000), Tree-SOM Samsonova et al. (2006) et SOM-AT Peura (1999). Notre approche Gr-SOTree génère non seulement une carte topologique mais aussi simultanément plusieurs arbres hiérarchiques pendant l'étape d'apprentissage. Un nœud dans la structure d'arbre représente une donnée du graphe.

Nous avons précédemment proposé une version de ce travail mais qui traite des données traditionnelles (individu/variable) Anonymes (2010). Nous avons adapté notre modèle aux données graphes en proposant ainsi un nouvel espace de représentation du graphe : topologique et hiérarchique. Nous avons introduit de nouvelles notions liées à la structure en graphes des données (référent 'leader', fonction de coût, fonction d'affectation en groupe).

2 Clustering de graphes : le modèle GC-SOTree

Les graphes sont des objets combinatoires décrits par : le degré, la connectivité, le chemin et le poids. Ainsi, les méthodes de clustering ne peuvent pas s'appliquer directement sur les graphes. Nous avons ainsi cherché à étudier comment les méthodes vectorielles peuvent être appliquées aux données de type graphes. L'idée commune est d'utiliser une transformation de graphe dans un nouvel espace où la similarité peut être calculée. Des approches utilisant les modèles des cartes auto-organisées comme : dissimilarité SOM (D-SOM) Kohonen et Somervuo (2002) et Kernel SOM Macdonald et Fyfe (2000) ont été proposés pour s'adapter à ce type de données en utilisant des fonctions noyau. Une autre alternative serait d'utiliser la matrice Laplacienne.La combinaison d'un vecteur ou de plusieurs vecteurs propres L est suffisante pour calculer la similarité ou la distance géométrique entre les nœuds Chung (1997), Luxburg (2007).

L'idée principale de notre modèle est de reconstruire le graphe d'origine $G(V, E)$ en le décomposant de manière hiérarchique en plusieurs sous-arbres auto-organisés, formant ainsi une forêt d'arbres projetée sur une carte 2D. En suivant la théorie spectrale des graphes, nous utilisons, en partie, le laplacien et ses vecteurs propres. Ainsi, nous considérons les premiers λ vecteurs propres $e_1, .., e_\lambda$ ($\forall i = 1..\lambda, e_i \in \Re^n$) pour former une nouvelle matrice des données $X \in \Re^{n \times \lambda}$. Une ligne de $\mathbf{x}_i \in X$ illustre un nœud $v_i \in V$ du graphe. Ainsi, nous pouvons décrire un graphe G dans l'espace continu X où la similitude entre deux sommets $sim(v_i, v_j)$ est équivalente à la distance entre leurs vecteurs respectifs $\mathbf{x}_i$ et $\mathbf{x}_j$. L'utilisation de λ vecteurs propres ($1 \leq \lambda \leq n$) pour représenter tous les nœuds du graphe dans $\Re^{n \times \lambda}$, permet de réduire la dimension.

En utilisant les modèles topologiques, un graphe G représenté par X peut être regroupé et visualisé dans une grille régulière en 2D ou 1D (de taille K) en utilisant le processus d'auto-organisation. La grille $\mathcal{C}$ représente une forêt d'arbres organisés sur la grille en K sous-graphes représentés en arbre. Chaque cellule c de la grille est appelée par la suite «support (root)» d'un arbre noté $tree_c$ et chaque nœud de l'arbre représente un nœud $v_i \in V$. Les arbres sont

construits en utilisant les principes d'un algorithme de classification non supervisé hiérarchique Azzag et al. (2006). Chaque nœud v_i sera connecté au plus proche voisin v_j de la même manière que $\mathbf{x}_i$ est le vecteur le plus proche de $\mathbf{x}_j$. Pour chaque paire d'arbres $tree_c$ et $tree_r$ sur la carte, la distance $\delta(c, r)$ est définie comme la longueur de la plus courte chaîne reliant les deux arbres. L'influence mutuelle entre deux sous-arbres $tree_c$ et $tree_r$ de racine c et r sera donc définie, de la même manière que les cartes topologiques, par la fonction $\mathcal{K}^T(\delta(c, r))$ où T représente la taille du voisinage (la température).

Principalement, les algorithmes de clustering ont en commun de considérer le barycentre de chaque cluster comme le prototype qui doit être mis à jour pour chaque itération. Toutefois, le centroïde n'est pas l'élément le plus important dans un graphe. Dans notre approche, au lieu d'utiliser le centroïde, nous avons opté pour l'utilisation du meneur "leader" qui est considéré comme le sommet représentant de tous les sommets d'un graphe ou d'un sous-graphe, Stanoev et al. (2011). Dans la première version de ce travail, nous avons considéré un "leader" comme le nœud qui a le plus grand degré. Ainsi, nous associons à chaque $tree_c$ un prototype noté $leader(c)$ dont l'expression est définie comme suit :

$$leader(c) = \max_{\forall i \in tree_c} (deg(v_i)) \tag{1}$$

Le choix d'un prototype représentatif permet facilement d'adapter notre algorithme à d'autres types de données. Ainsi, la fonction objective de l'auto-organisation des arbres s'écrit comme suit :

$$\mathcal{R}(\phi, \mathcal{L}) = \sum_{c=1}^{k} \sum_{r=1}^{k} \sum_{i \in tree_c} \mathcal{K}(\delta(\phi(\mathbf{x}_i), r)) ||\mathbf{x}_i - \mathbf{x}_{leader(r)}||^2 \tag{2}$$

où $\mathcal{L} = \cup_{r=1}^{k} leader(r)$ and ϕ est la fonction d'affectation d'un groupe de nœuds à la fois.

$$\phi(\textit{childNode}(v_i)) = \arg\min_{r} \sum_{c=1..k} \mathcal{K}^T(\delta(r, c)) \left\|\mathbf{x}_i - \mathbf{x}_{leader(c)}\right\|^2 \tag{3}$$

où $childNode(v_i)$ est l'ensemble des nœuds contenant v_i et tous les nœuds récursivement connectés à v_i. La figure 1 montre un exemple d'affectation simultanée d'un ensemble de nœuds $\{e, f, g\} \subset V$ formant un arbre.

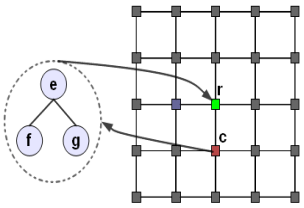


FIG. 1 – *Exemple d'une affectation d'un arbre de nœuds de la cellule c à la cellule r. $\{e, f, g\} \subset V$, $childNode(e) = \{e, f, g\}$*

Require: $v_i \in V$ ($\mathbf{x}_i \in X$), Carte $\mathcal{C}(c \in \mathcal{C})$
Ensure: Forêt d'arbre
1: **repeat**
2: $c \leftarrow \phi(childNode(v_i))$.
3: **if** v_i est connecté **then**
4: $tree_c \leftarrow$ constructTree(v_i, $tree_c$)
5: **else**
6: **if** $c! = c_{old}$ **then**
7: **if** v_i n'est pas déconnecté **then**
8: déconnecter v_i et ses fils de l'arbre $tree_{c_{old}}$
9: **end if**
10: $childNode(v_i) \leftarrow$ reçoit tous les nœuds de v_i
11: $tree_c \leftarrow$ constructTree($childNode(v_i)$,$tree_c$)
12: Mise à jour des meneurs "Leaders" ($r \in \mathcal{C}$)
13: **end if**
14: **end if**
15: **until** Itération Finale

ALG 1: Gr-SOTree

La minimisation de la fonction de coût $\mathcal{R}$ est un problème d'optimisation combinatoire. En pratique, on se contente d'une solution sous-optimale. Nous proposons ici de minimiser la fonction de coût de la même manière que la version des nuées dynamiques, mais en utilisant les caractéristiques statistiques fournies par les arbres (associés à chaque cellule) afin d'accélérer la convergence de l'algorithme.

Algorithme Gr-SOTree : self-Organizing Tree

Notre algorithme principal est donné en pseudo-code par l'algorithme 1. La fonction *construct-Tree* est présentée par l'algorithme 2. Notons par v_{pos} le nœud ou le support où le nœud v_i est connecté dans l'arbre qui est associé à une cellule, et v^+ le nœud connecté à v_{pos} le plus similaire à v_i. Si un nœud v_i a été déconnecté de l'arbre, ce nœud et ses nœuds-fils vont être reconnectés selon la ligne 11 de l'algorithme 1. La fonction *constructTree* est appelée pour chaque nœud de l'ensemble $childNode(v_i)$.

3 Validation numérique et visuelle

Pour évaluer la qualité de l'approche proposée. Nous avons utilisés 3 bases de données supervisées disponibles sur `http://www-personal.umich.edu/~mejn/netdata/`. Ces bases représentent des graphes non orientés et non pondérés.

3.1 Mesure de qualité

Soit $G(V, E)$ un graphe partitionné en k clusters $tree_1, .., tree_c, .., tree_k$. Le nombre de nœuds dans $tree_c$ est noté par N_c ; M_c est le nombre d'arcs dans $tree_c$, $M_c = \{\{u, v\}; u, v \in$

Require: $v_i \in V, tree_j$
Ensure: $Arbres$
1: **if** pas de nœuds ou uniquement un seul nœud connecté à v_{pos} **then**
2: $tree_j \leftarrow$ connecter v_i à v_{pos}
3: **end if**
4: **if** Deux nœud connectés à v_{pos} et c'est la première fois **then**
5: $tree_j \leftarrow$ déconnecter le plus similaire à v_i de v_{pos} (et récursivement tous les nœuds-fils v_i)
6: $tree_j \leftarrow$ connecter v_i à v_{pos}
7: **else**
8: $T_{Dissim}(v_{pos}) \leftarrow min(sim(v_k, v_j))$ où v_j, v_k sont les fils de v_{pos}
9: **if** $sim(v_i, v^+) < T_{Dissim}(v_{pos})$ **then**
10: $tree_j \leftarrow$ connecter v_i à v_{pos}
11: **else**
12: déplacer v_i à v^+
13: **end if**
14: **end if**

ALG 2: constructTree

TAB. 1 – *Bases de données*

Bases	Nœuds	Arcs	Classes
Adjective and Noun	112	425	2
Football Teams	115	616	10
Political blogs	1490	19090	2

$tree_c\}$; et B_c est le nombre d'arcs à la frontière de $tree_c$, $B_c = \{\{u, v\}; u \in tree_c, v \notin tree_c\}$. Pour mesurer la qualité d'un clustering de graphe, nous utilisons 3 critères d'evaluation Leskovec et al. (2010) :

- Conductance : $C = \frac{B_c}{2M_c+B_c}$ mesure la proportion des arêtes qui pointent en dehors du cluster $tree_c$.
- Densité : $V = \frac{M_c}{N_c(N_c-1)/2}$ est la densité des arêtes internes au sein du cluster $tree_c$.
- Modularité Newman (2006b), Clauset et al. (2004), Newman (2006a) :
  $Q = \frac{1}{2m} \sum_{c=1}^{k} \sum_{i \in tree_c} \sum_{j \in tree_c} \left(w_{ij} - \frac{deg(i)deg(j)}{2m} \right)$
  où $m = \frac{1}{2} \sum_{i=1}^{n} deg(i)$.
  L'idée est de prendre le nombre d'arêtes relevant des groupes moins le nombre prévu dans un réseau équivalent avec des arêtes placées au hasard.

On juge qu'un cluster est de bonne qualité s'il a plus de liens internes que de liens externes avec les autres clusters. Ainsi une bonne classification doit retourner une bonne valeur de la modularité et de la densité, à contrario une faible valeur de la conductance.

3.2 Résultats numériques

La table 3.2 présente les mesures moyennes de la qualité et leurs écarts-types obtenus à partir de 10 expériences. Les bases étant supervisées, nous avons pu également calculer la proportion de données bien classées appelée pureté. Tout d'abord, on remarque que la qualité de la classification dépend fortement du choix de λ, mais dans ce travail nous ne cherchons pas pour l'instant à optimiser la valeur de λ et son influence sur les mesures de qualité. Dans cette section expérimentale, nous nous limitons à effectuer certaines expériences afin d'évaluer et de comparer notre méthode avec d'autres approches : K-means, SOM et MST (Minimum Spanning Tree). La méthode MST Grygorash et al. (2006) construit des graphes de voisinages à partir d'un ensemble de données (nœuds) en connectant toutes les pairs de nœuds qui satisfont une certaine fonction de coût basée sur la distance

Dans le tableau 3.2, les résultats de la base "Adjective and noun" montre que les mesures de qualité pour K-means, SOM et Gr-SOTree sont pour la plupart égaux sauf pour la modularité. Le meilleur résultat est obtenu par Gr-SOTree lorsque $\lambda = 11$ avec une densité de 0.30. Cependant, les mauvais résultats pour MST sont assez décevants. Concernant la base "Football", notre méthode est meilleur avec une densité de 0.792 pour $\lambda = 11$. Par contre, la différence dans les valeurs de conductance n'est pas significative, et les valeurs de notre modularité donnent des résultats comparables à ceux obtenues par K-means et SOM. Contrairement au premier exemple, l'algorithme MST semble meilleur, mais encore loin derrière. La meilleure situation pour Gr-SOTree concerne la base «Political blogs» où la méthode proposée domine toutes les mesures pour $\lambda = 39$, sauf pour la pureté.

Method	λ	Conductance $\searrow$	Density $\nearrow$	Purity $\nearrow$	Modularity $\nearrow$
Adjective and noun					
K-means	5	0.734 ± 0.012	0.242 ± 0.044	0.580 ± 0.023	0.139 ± 0.012
	11	0.684 ± 0.015	0.286 ± 0.048	0.577 ± 0.019	0.167 ± 0.027
SOM	5	0.743 ± 0.024	0.175 ± 0.015	0.574 ± 0.028	0.160 ± 0.014
	11	0.691 ± 0.022	0.214 ± 0.025	0.576 ± 0.022	0.199 ± 0.017
MST	5	0.894	0.014	0.580	0.028
	11	0.897	0.008	0.571	0.013
Gr-SOTree	5	0.784 ± 0.014	0.208 ± 0.038	0.565 ± 0.004	0.125 ± 0.015
	11	0.736 ± 0.029	0.300 ± 0.086	0.560 ± 0.016	0.126 ± 0.042
Football Teams					
K-means	5	0.573 ± 0.024	0.717 ± 0.038	0.849 ± 0.045	0.455 ± 0.033
	11	0.541 ± 0.036	0.770 ± 0.057	0.863 ± 0.056	0.480 ± 0.045
SOM	5	0.568 ± 0.085	0.529 ± 0.076	0.692 ± 0.037	0.505 ± 0.018
	11	0.406 ± 0.081	0.645 ± 0.078	0.735 ± 0.055	0.544 ± 0.020
MST	5	0.610	0.440	0.800	0.563
	11	0.478	0.706	0.965	0.589
Gr-SOTree	5	0.564 ± 0.012	0.715 ± 0.060	0.880 ± 0.016	0.464 ± 0.025
	11	0.532 ± 0.039	0.792 ± 0.057	0.878 ± 0.058	0.479 ± 0.037
Political Blogs					
K-means	5	0.801 ± 0.039	0.054 ± 0.013	0.877 ± 0.008	0.114 ± 0.015
	39	0.813 ± 0.028	0.083 ± 0.030	0.840 ± 0.011	0.180 ± 0.024
SOM	5	0.854 ± 0.016	0.081 ± 0.021	0.861 ± 0.003	0.116 ± 0.022
	39	0.845 ± 0.024	0.062 ± 0.017	0.827 ± 0.016	0.160 ± 0.017
MST	5	0.961	0.007	0.53	0
	11	0.928	0.003	0.512	0
Gr-SOTree	5	0.884 ± 0.027	0.094 ± 0.009	0.854 ± 0.013	0.117 ± 0.025
	39	0.785 ± 0.015	0.227 ± 0.071	0.767 ± 0.028	0.197 ± 0.035

3.3 Visualisation des graphes

L'objectif de cette partie expérimentale est de montrer que notre méthode fournit des informations supplémentaires. Dans notre approche nous proposons une décomposition du graphe d'origine en fournissant une organisation multi-niveaux : hiérarchique et topologique. Ces structures permettent de simplifier l'exploitation du graphe en offrant une visualisation conviviale. Nous avons utilisé Tulip ,Auber (2003), comme plateforme de visualisation.

– **Visualisation du graphe original** :
Le graphe original s'affiche avec une couleur unique pour l'ensemble des nœuds dont les étiquettes sont disponibles. La taille du nœud varie selon le degré.
– **Visualisation multi-niveaux : topologique et hiérarchique** :
Cette visualisation représente un nouvelle espace de reconstruction qui produit un graphe hyperbolique et interactif où le premier niveau d'organisation est la vue de la carte topologique qui affiche à chaque cellule un arbre. Nous pouvons également voir la hiérarchie des k arbres créés par Gr-SOTree. Un arbre est clairement identifié par sa structure et

par une couleur différente. Les meneurs "leaders" apparaissent également dans chaque arbre. Par conséquent, Il devient plus facile d'extraire des informations de cette nouvelle arborescence extraite du graphe d'origine. D'autre part, en raison des relations hiérarchiques, cette visualisation a permis d'éliminer les nœuds isolés ou les groupes isolés. Un exemple est présenté dans les figures 2(b) et 3(b).

– **Visualisation des groupes** :
Chaque cluster est représenté par un "leader" et une couleur unique. Pour les trois ensembles de données, l'étiquette d'un cluster (ou une cellule) est déterminée par la règle du vote majoritaire. Par conséquent, nous pouvons dessiner le graphe et permettre aux utilisateurs de comparer avec le graphe d'origine. Par exemple, nous remarquons que les "leaders" sont regroupés en deux grands groupes, figure 3. Chaque cluster peut être représenté par plusieurs "leaders" que nous pouvons considérer comme des points d'intérêts. Il convient de noter que les nœuds qui ont le plus haut degré ne sont pas toujours choisis comme "leader" parce que ces nœuds ont de nombreux liens avec d'autres groupes et que ce type de nœud ne satisfait pas l'équation 1.
– **Visualisation topologique** :
Pour la visualisation topologique, nous présentons ici deux notions : les liens forts (en rose) et les liens faibles (en gris). Comme nous l'avons mentionné plus haut, l'algorithme SOM utilise une fonction de voisinage pour déterminer les voisins d'une cellule et par conséquent d'un arbre. Un lien faible est créé entre deux cellules voisines. Un lien fort est crée s'il existe une arête dans graphe original qui relie un couple de "leaders" situés dans deux cellules voisines. Les figures 2(e) et 3(e) représentent un exemple de visualisation des liens forts et faibles. La taille d'une cellule est proportionnelle aux nombre de nœuds affectés à cette cellule.

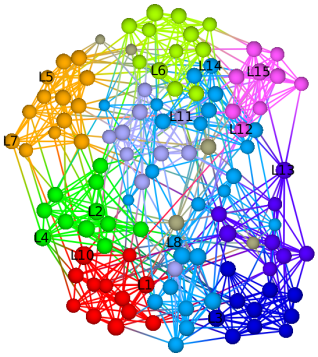
4 Conclusions et perspectives

Dans ces travaux nous avons présenté une nouvelle approche pour le clustering de graphes. Cette nouvelle méthode basée sur l'auto-organisation fournit un nouvel espace de représentation : la carte 2D et les arbres permettant une meilleure représentation d'un graphe. Nous avons introduit deux nouvelles notions importante : la notion de leader en tant que prototype et l'utilisation de la structure d'arbre pour définir une fonction d'affectation d'un ensemble de nœuds. Notre modèle offre un nouvel espace de visualisation proposant une representation du graphe plus riche en information.

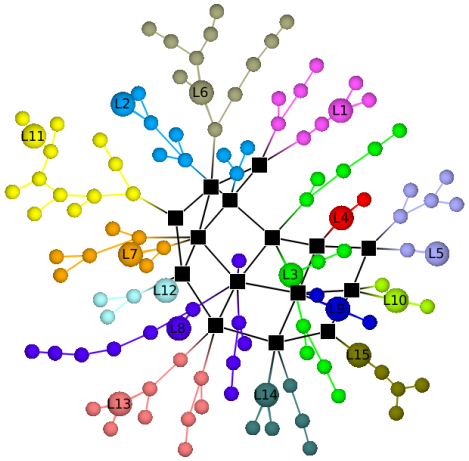
Comme travail futur, nous souhaitons étudier une approche de graphe dynamique et évolutif. En construisant ainsi des sous-parties du graphe de manière incrémentale, nous souhaitons, montrer comment des sous-graphes peuvent évoluer au cours du temps. Comme autre perspective nous soulignons l'importance de se concentrer sur le nouveau concept de points d'intérêt qui caractérisent les points "leader" d'un cluster.

Références

Anonymes (2010). Topological hierarchical tree using artificial ants. In *Proceedings of the 17th international conference on Neural information processing : theory and algorithms -*

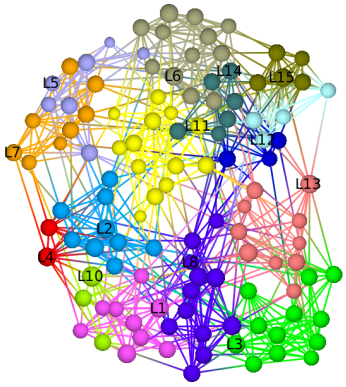


(a) Graphe original : chaque couleur indique la classe réelle.

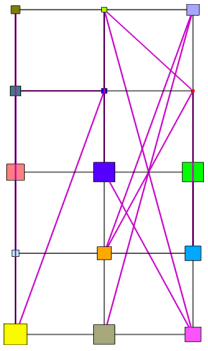


(b) Visualisation multi-niveaux : topologique et hiérarchique. Les "leaders" sont indiqués avec des nœuds de taille plus grande et avec le symbole L.

(c) Visualisation des groupes : chaque couleur indique la classe obtenue après vote majoritaire.

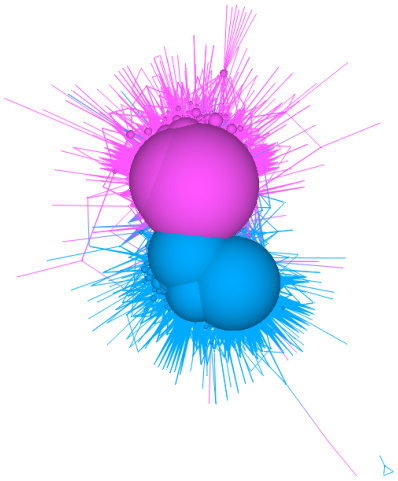


(d) Visualisation des groupes : Le graphe est partitionné en 5×3 groupes.

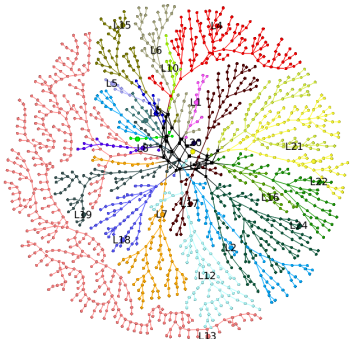


(e) Visualisation topologique : Carte 3×3 avec les liens forts et faibles. La taille des cellules est proportionnelle aux nombres de nœuds. Chaque couleur indique la classe trouvée après vote majoritaire.

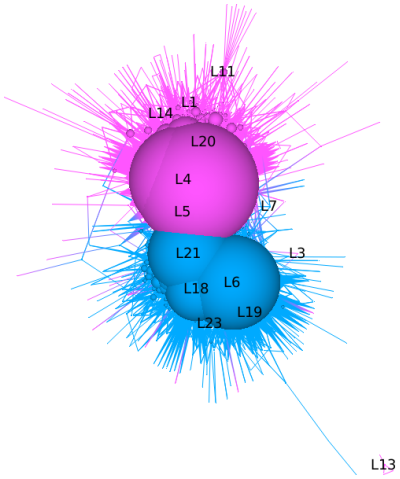
FIG. 2 – *Graph visualization of "Football Teams"*



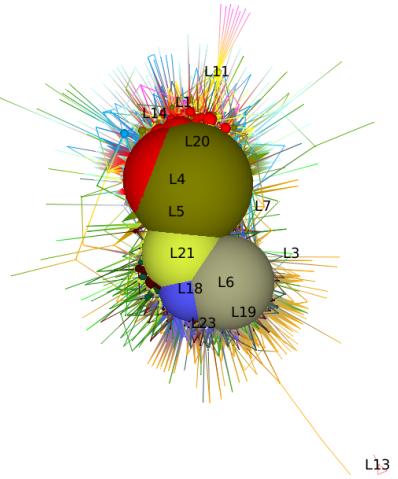
(a) Graphe original : chaque couleur indique la classe réelle.



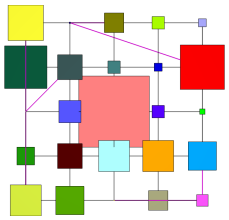
(b) Visualisation multi-niveaux : topologique et hiérarchique. Chaque arbre représente une cellule. Les 'leaders' sont indiqués avec des nœuds de taille plus grande et avec le symbole L.



(c) Visualisation des groupes : chaque couleur indique la classe obtenue après vote majoritaire.



(d) Visualisation des groupes : Le graphe est partitionné en 5×5 groupes.



(e) Visualisation topologique : Carte 3×3 avec les liens forts et faibles. La taille des cellules est proportionnelle aux nombres de nœuds. Chaque couleur indique la classe trouvée après vote majoritaire.

FIG. 3 – *Graph visualization of "Political Blogs"*

Volume Part I, ICONIP'10, Berlin, Heidelberg, pp. 652–659. Springer-Verlag.

Auber, D. (2003). Tulip : A huge graph visualisation framework. In P. Mutzel et M. Jünger (Eds.), *Graph Drawing Softwares*, Mathematics and Visualization, pp. 105–126. Springer-Verlag.

Azzag, H., C. Guinot, et G. Venturini (2006). Data and text mining with hierarchical clustering ants. In *Swarm Intelligence in Data Mining*, pp. 153–189.

Barbara Hammer, Alexander Hasenfuß, F. R. (2009). Median topographic maps for biomedical data sets. *CoRR*, 0909.0638.

Bishop, C. M., M. Svensén, et C. K. I.Williams (1998). Gtm : The generative topographic mapping. *Neural Comput 10*(1), 215–234.

Chung, F. R. K. (1997). *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society.

Clauset, A., M. E. J. Newman, , et C. Moore (2004). Finding community structure in very large networks. *Physical Review E*, 1– 6.

Diday, E. et J. Simon (1976). Cluster analysis. In K. Fu (Ed.), *Digital Pattern Recognition*, pp. 47–94. Springer-Verlag, Berlin.

Dittenbach, M., D. Merkl, et A. Rauber (2000). The growing hierarchical self-organizing map. pp. 15–19. IEEE Computer Society.

Dittenbach, M., A. Rauber, et D. Merkl (2001). Recent advances with the growing hierarchical self-organizing map.

Fabrice Rossi, N. V.-V. ((2010)). Optimizing an organized modularity measure for topographic graph clustering : A deterministic annealing approach. *Neurocomputing 73(7-9)*.

Grygorash, O., Y. Zhou, et Z. Jorgensen (2006). Minimum spanning tree based clustering algorithms. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '06, Washington, DC, USA, pp. 73–81. IEEE Computer Society.

Kohonen, T. et P. Somervuo (2002). How to make large self-organizing maps for nonvectorial data. *Neural Networks 15*(8-9), 945–952.

Koikkalainen, P. et I. Horppu (2007). Handling missing data with the tree-structured self-organizing map. In *IJCNN*, pp. 2289–2294.

Leskovec, J., K. J. Lang, et M. Mahoney (2010). Empirical comparison of algorithms for network community detection. In *WWW '10 : Proceedings of the 19th international conference on World wide web*, New York, NY, USA, pp. 631–640. ACM.

Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing 17*, 395–416.

Macdonald, D. et C. Fyfe (2000). The kernel self-organising map. In *Knowledge-Based Intelligent Information and Engineering Systems*, pp. 317–320.

Newman, M. E. J. (2006a). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) 74*(3), 036104+.

Newman, M. E. J. (2006b). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences 103*(23), 8577–8582.

Peura, M. (1999). The self-organizing map of attribute trees.

Samsonova, E. V., J. N. Kok, et A. P. Ijzerman (2006). Treesom : Cluster analysis in the self-organizing map. neural networks. *American Economic Review 82*, 1162–1176.

Stanoev, A., D. Smilkov, et L. Kocarev (2011). Identifying communities by influence dynamics in social networks. *CoRR abs/1104.5247*.

Summary

In this paper, we present a new approach of graph clustering called GC-SOtree (Graph Clustering using Self-Organizing Trees) based on self-organizing approaches. Our method builds a simultaneous topological and hierarchical partitioning of data. The clustering is presented in 2D grid where each cell of map is represented using tree clustering inspired from self-assembly behavior of artificial ants. The benefit of this novel approach is to represent hierarchical and topological relations in the graph, which gives a good understanding of the underlying problem.

Exploration graphique de données séquentielles

Reto Bürgin, Gilbert Ritschard, Emmanuel Rousseaux

Institut d'études démographiques et du parcours de vie, Université de Genève
et Pôle de recherche national LIVES 'Vulnérabilités et parcours de vie'
{reto.buergin, gilbert.ritschard, emmanuel.rousseaux}@unige.ch,
http://mephisto.unige.ch/

Résumé. Nous proposons une façon originale de représenter graphiquement des données longitudinales catégorielles. La visualisation proposée, inspirée des courbes de séries temporelles, se prête particulièrement bien à la description et à l'exploration de trajectoires individuelles décrites sous forme de séquences d'événements. Outre la description de la méthode de visualisation et des principes qui la président, l'article comprend des exemples d'application et une discussion des propriétés des graphiques produits. De plus, nous expliquons également quelques astuces de spécification permettant d'optimiser le rendu des données.

1 Introduction

L'analyse de données longitudinales a connu un essor important ces dernières années dans de nombreux domaines, par exemple pour suivre les comportements de consommateurs ou d'utilisateurs de services, suivre les développement cognitifs d'individus, contrôler le fonctionnement d'appareils, ou encore pour étudier les carrières professionnelles ou les trajectoires familiales. L'étendue et la portée des applications ne cesse de croître (Frees, 2004), et cette demande a engendré le développement de méthodes d'analyse spécifiques pour données longitudinales. Du côté de la statistique, l'effort a été mis, pour les séquences numériques sur les approches confirmatoires comme les modèles de régression multiniveaux (Hox, 2010), les modèles d'équations structurelles à croissance latente pour données répétées (McArdle, 2009), les modèles de survie, ou encore les modèles de transition de type markovien (Berchtold et Raftery, 2002) et, dans une optique plus exploratoire, sur la caractérisation de configurations typiques de séquences d'états. Du côté de la fouille de données, l'accent a porté sur la recherche de sous-séquences fréquentes et de règles d'association séquentielles (Agrawal et Srikant, 1995; Masseglia, 2002).

L'objectif premier de la présente contribution est de proposer des outils graphiques pour l'exploration de parcours de vie décrits sous forme de séquences d'événements (quitter ses parents, finir les études, se mettre en couple, déménager, etc.). Plus particulièrement, nous nous intéressons aux graphiques qui mettent en évidence les caractéristiques principales de l'ensemble tout en rendant compte des trajectoires individuelles et de leur diversité.

Les graphiques pour données longitudinales que l'on trouve dans la littérature concernent principalement le rendu de séquences d'états (Gabadinho et al., 2011) que l'on visualise par

exemple facilement sous forme de barres empilées dont la couleur des éléments représente l'état et la longueur le temps passé dans l'état. Pour les séquences numériques, on recourt en général aux graphiques superposant les courbes de séries temporelles (Tufte, 2001), parfois appelés 'spaghetti plots'. Parmi les autres formes de représentation de séquences individuelles, on peut citer encore les graphes reliant les états ou événements successifs de chaque séquence (Hébrail et Cadalen, 2000), ou les graphiques qui visualisent sur une ligne ou une barre distincte pour chaque séquence le calendrier d'occurrence des événements. Dans cette optique, les variantes proposées sous forme de 'Life Flow' dans Wongsuphasawat et al. (2011) sont particulièrement intéressantes.

Les séquences d'événements qui nous intéressent s'apparentent à celles que l'on considère dans la fouilles des séquences d'achats de consommateurs (Agrawal et Srikant, 1995). Ces séquences se distinguent des séquences d'états sur deux points qui rendent beaucoup plus difficile leur visualisation. D'une part, les événements interviennent à un moment donné et n'ont donc pas de durée, ce qui exclut l'utilisation de barres de longueur proportionnelle à la durée, et d'autre part, plusieurs événements peuvent intervenir simultanément, par exemple terminer ses études et se mettre en ménage le même mois ou acheter plusieurs produits lors d'une même commande. Ceci explique peut-être le peu de place accordée jusqu'ici à la visualisation de séquencements d'événements. Parmi les rares propositions que l'on trouve dans la littérature figurent les calendriers d'événements et leurs variantes (Wongsuphasawat et al., 2011) déjà cités, et des représentations plus synthétiques comme la superposition des courbes de survie jusqu'à l'occurrence des divers événements (Studer et al., 2010). Ces représentations s'avèrent inadaptées à nos objectifs qui sont :

– visualiser la diversité des trajectoires individuelles observées ;
– identifier les séquencements les plus communs ;
– représenter des séquences individuelles pouvant comprendre des événements simultanés.

En effet, les graphiques de type calendrier se prêtent par exemple mal à la gestion d'événements simultanés et au cas de grands nombres de séquences, tandis que les représentations sous forme de graphes ainsi que les représentation agrégées comme les courbes de survie ne rendent pas compte des parcours individuels observés.

La représentation que nous proposons s'inspire des 'spaghetti plots' de séries temporelles numériques que nous adaptons pour satisfaire les trois critères ci-dessus. Bien que conçu pour visualiser des séquencements d'événements, le graphique proposé s'avère toutefois également utile pour explorer d'autres types de données longitudinales. Nous le verrons en particulier avec l'un des deux jeux de données illustratives considérés qui concerne des séquences d'états.

Un graphique se doit d'être aisément lisible et interprétable. Dans ce but, nous nous sommes attachés à suivre les indications de Diggle et al. (2002) pour une visualisation efficace de données longitudinales, à savoir :

1. montrer les données brutes pertinentes plutôt que des résumés synthétiques ;
2. mettre en évidence les modèles de configuration potentiellement intéressants ;
3. identifier les caractéristiques transversales et longitudinales de l'ensemble ;
4. faciliter l'identification des individus et observations atypiques.

L'article est organisé comme suit. Nous commençons par introduire deux jeux de données qui nous servirons d'illustration. Nous explicitons ensuite les principes de notre méthode et illustrons leur usage. Finalement, nous discutons les avantages et inconvénients de notre proposition avant d'évoquer en conclusion quelques pistes de développements futurs.

2 Données illustratives

Nous considérons deux jeux de données réels. Dans le premier cas, les éléments des séquences sont les états mutuellement exclusifs (pas de simultanéité) d'une variable ordinale. Dans le second exemple, les éléments sont des événements de vie qui peuvent être simultanés.

Anxiété après un séjour aux soins intensifs. 149 patients ayant séjournés dans l'unité de soins intensifs (ICU) de l'Hôpital universitaire de Berne ont été interrogés à trois reprises après leur séjour (Jeitziner et al., 2011). Ils devaient indiquer l'intensité de leur anxiété sur une échelle numérique 0-10 (0 = aucune anxiété, 10 = pire anxiété). Les interviews ont respectivement eu lieu $t_1 = 1$ semaine, $t_2 = 6$ mois et $t_3 = 12$ mois après leur séjour à l'ICU. 117 patients ont répondus les trois fois, et les autres une ou deux fois.

Des analyses préliminaires ont montré qu'il était difficile de voir les trajectoires effectives et d'identifier les séquences typiques avec une représentation sous forme de courbes de séries temporelles, les courbes étant à la fois trop diverses et se masquant les unes les autres. Pour notre illustration, nous discrétisons l'échelle des valeurs en 3 classes : $[0, 2]$, $(2, 6]$ and $(6, 10]$.

id	trajectoire
1	$[0,2]^{t_1} \rightarrow (6,10]^{t_2} \rightarrow (2,6]^{t_3}$
2	$[0,2]^{t_1} \rightarrow [0,2]^{t_2} \rightarrow [0,2]^{t_3}$
3	$[0,2]^{t_1} \rightarrow [0,2]^{t_2} \rightarrow [0,2]^{t_3}$

TAB. 1 – *Extrait de séquences de niveaux d'anxiété, patients 1, 2 and 3.*

Le tableau 1 présente les trajectoires de niveaux d'anxiété de trois patients avec les notations de Studer et al. (2010), où les flèches séparent les éléments successifs et l'indice supérieur reflète la date d'observation, ou, lorsque celle-ci n'est pas disponible, simplement la position dans la séquence. Le premier patient, par exemple, déclare une anxiété de niveau $[0, 2]$ une semaine après son séjour au soins intensifs, $(6, 10]$ six mois plus tard et de niveau $(2, 6]$ après un an.

Evénements de vie familiale. Il s'agit de données tirées de l'enquête biographique réalisée en 2002 par le Panel Suisse des Ménages (`http://www.swisspanel.ch`). Plus précisément, nous utilisons ici les données relatives à 2601 enquêtés qui avaient atteint l'âge de 45 ans au moment de l'enquête, c'est-à-dire ceux né en 1957 ou avant. Les événements retenus sont le départ du domicile parental (Depart), la première mise en couple (Couple), le premier mariage, le premier enfant et le premier divorce. On connaît l'année d'occurrence de ces événements, mais on ne retient ici que leur ordre en considérant comme simultanés les événements survenant une même année. On dispose en tout de 9021 événements pour l'ensemble des 2601 cas retenus.

Le tableau 2 donne à titre d'exemple les trajectoires de vie familiale de trois individus. On y lit notamment que l'individu 2600 a vécu tout d'abord le départ du domicile de ses parents avant de connaître plus tard trois événements au cours d'une même année, à savoir la première mise en couple, le mariage et la naissance du premier enfant.

id	trajectoire
2599	(Depart, Couple, Mariage, Enfant)^1
2600	(Depart)^1 $\rightarrow$ (Couple, Mariage, Enfant)^2
2601	(Depart, Couple, Mariage, Enfant)^1

TAB. 2 – *Trajectoires de vie familiale de trois individus.*

3 Méthode

Le point de départ de la représentation graphique proposée est une simple transposition du principe des 'spaghetti plots' pour séries temporelles numériques au cas de données catégorielles. L'axe des x reflète la date ou position dans la séquence, et l'axe des y la valeur de la variable, soit dans notre cas l'événement ou l'état. On reporte ainsi sur l'axe des ordonnées les divers éléments de l'alphabet catégoriel considéré, dans un ordre arbitraire si l'alphabet est nominal et dans l'ordre établi s'il est ordinal. On peut ainsi visualiser les trajectoires en reportant en regard de chaque position un point à la hauteur de l'événement (ou état) concerné et en reliant ces points-événements successifs par des lignes. En cas d'événements simultanés, on connecte les points du bas vers le haut par un trait vertical. Ceci est arbitraire et une alternative serait par exemple de les connecter de haut en bas.

La Figure 1 illustre le graphique ainsi obtenu pour les données des événements de vie familiale. L'ordre des événements est arbitraire et nous les avons ici ordonnés selon ce qui est encore l'ordre standard en Suisse. Le résultat est un graphe où toutes les paires observées

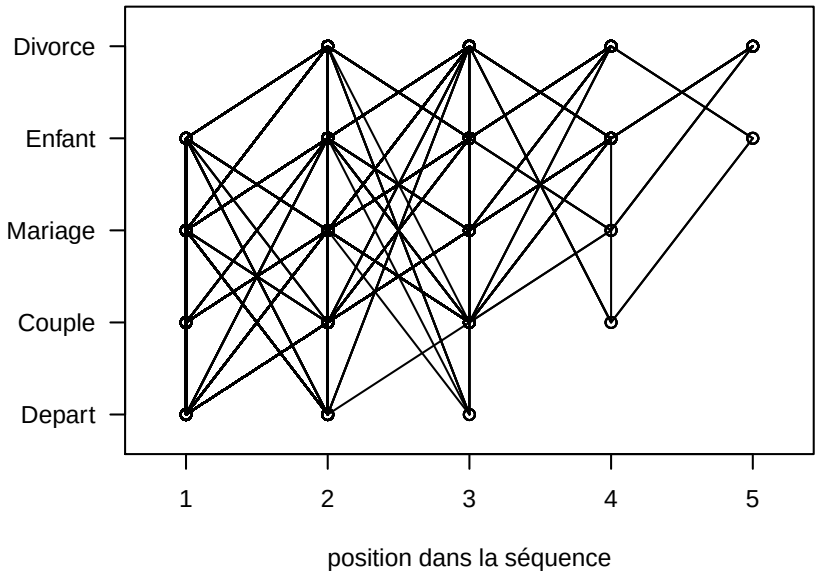


FIG. 1 – *Graphique de type 'séries temporelles' des séquences d'événements de vie familiale.*

d'événements consécutifs ou simultanés sont reliés. Si ce premier graphique permet de rendre compte de la simultanéité d'événements, il reste insatisfaisant car, en raison du caractère discret des événements, les trajectoires tendent à se masquer partiellement ou totalement les unes les autres. Ainsi, il ne permet en général ni de repérer les trajectoires fréquentes, ni a fortiori les trajectoires atypiques, ni de retracer les trajectoires individuelles. Il peut même être trompeur en laissant penser que toute séquence d'événements successivement connectés comme $(\text{first child})^1 \rightarrow (\text{divorce})^2 \rightarrow (\text{first child})^3 \rightarrow (\text{divorce})^4 \rightarrow (\text{first child})^5$ qui est clairement impossible, aurait été observée.

Décalement des courbes. Une première solution pour éviter le recouvrement de séquences consiste à décaler légèrement chaque séquence représentée. Pratiquement, on applique la même translation à chaque point-événement d'une trajectoire.

Afin de faciliter le suivi d'une trajectoire, nous représentons en arrière plan la *zone de translation* sous forme de carrés gris clair (voir figure 2). Le point central de la zone de translation indique la coordonnée d'origine commune des points visibles dans la zone. Tous les points d'une même trajectoire occupent la même position dans leur zone respective de déplacement. Il est ainsi aisé de repérer, par exemple, où se termine une trajectoire donnée quand on connaît la position de son point de départ.

La taille de la zone de translation est paramétrable. Il convient de laisser un espace entre les zones que l'on suggère de fixer à au moins la demi-largeur, respectivement la demi-longueur des zones.

Le seul décalage des courbes reste cependant insuffisant, le graphique devenant rapidement illisible en présence d'un grand nombre de trajectoires.

Trajectoires distinctes. Pour réduire le nombre de courbes, on propose de ne visualiser que les configurations distinctes et de jouer sur l'épaisseur du trait et des points-événements pour rendre compte de la fréquence d'observation des trajectoires représentées. Considérons à titre d'exemple les quatre trajectoires du tableau 3. Comme les séquences 1 et 3 sont identiques, on représente les quatre séquences avec trois courbes (partie gauche de la figure 2). L'épaisseur des traits et points de la trajectoire $(A)^1 \rightarrow (B)^2$ indique qu'elle est deux fois plus fréquente.

Le risque est que les trajectoires les plus fréquentes, qui donnent lieu aux traits les plus épais, cachent les moins fréquentes. On peut éviter en partie cet effet, en variant les couleurs des trajectoires, éventuellement en jouant avec des effets de transparence, et en dessinant en premier les traits les plus épais de sorte à placer les traits plus fins par dessus. Une option permet de fixer le support minimal des séquences à représenter et de visualiser les autres en gris clair pour rendre compte de la diversité.

id	trajectoire
1	$(A)^1 \rightarrow (B)^2$
2	$(A)^1 \rightarrow (A, B)^2$
3	$(A)^1 \rightarrow (B)^2$
4	$(A)^1 \rightarrow (B)^2 \rightarrow (C)^3$

TAB. 3 – *Exemple de trajectoires.*

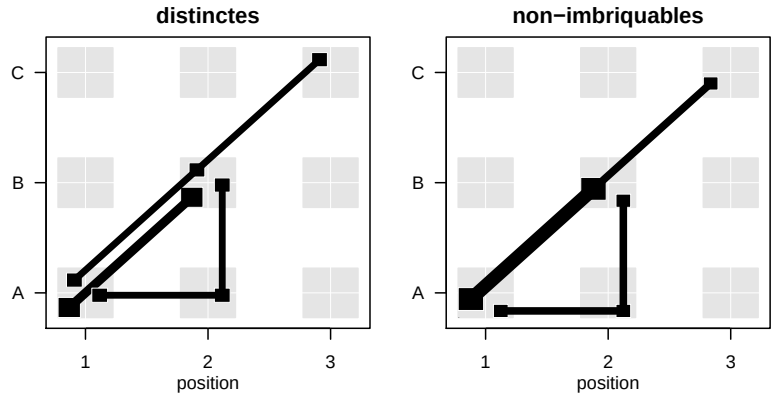


FIG. 2 – *Trajectoires distinctes et trajectoires non-imbriquables, données du tableau 3.*

Trajectoires non-imbriquables. Afin de réduire plus encore le nombre de courbes nécessaires pour rendre compte des séquences individuelles, on propose de ne visualiser que les trajectoires non imbriquées dans une séquence plus longue (voir figure 2 graphique de droite).

Définition. *Une séquence* $s1$ *débutant à la position* t_{s1} *est dite* imbriquée *dans une séquence* $s2$ *si est seulement si* $s2$ *contient de façon contiguë et alignée aux mêmes positions toute la séquence* $s1$*, c'est-à-dire, si* $s2$ *est de la forme* $s2 = sp \rightarrow s1 \rightarrow ss$*, avec des préfixes* sp *et suffixes* ss *éventuellement vides et le début de* $s1$ *en position* t_{s1}.

Par exemple, la séquence $(A)^1 \rightarrow (B)^2$ est imbriquée dans la séquence $(A)^1 \rightarrow (B)^2 \rightarrow (C)^3$, mais n'est pas imbriquée dans $(A)^1 \rightarrow (A, B)^2$. La notion de séquence imbriquée s'apparente à celle de sous-chaîne de caractères (*substring*), mais est plus générale puisqu'ici les caractères peuvent être des transactions ou transitions définies par des ensembles d'événements simultanés. Dans le cas où, comme dans l'exemple ci-dessus, les débuts de séquences sont tous alignés sur la 1ère position, il ne peut évidemment pas y avoir de préfixe sp.

Pour visualiser la présence de séquences imbriquées, on adapte l'épaisseur des points et segments communs indépendamment selon leur fréquence. Ainsi, dans le graphique de droite de la figure 2, on voit clairement qu'il existe des séquences $(A)^1 \rightarrow (B)^2$ imbriquées dans $(A)^1 \rightarrow (B)^2 \rightarrow (C)^3$.

Une difficulté est qu'une même séquence peut parfois s'imbriquer dans plusieurs séquences. La solution retenue est d'affecter son poids entièrement à une seule des séquences non imbriquables concernées, la plus fréquente. Alternativement, on pourrait répartir son poids de façon uniforme entre les diverses séquences non imbriquables concernées.

Si toutes les séquences sont de même longueur (même nombre de transactions dans le cas de séquences d'événements), les trajectoires distinctes et les trajectoires non-imbriquables sont les mêmes.

Notons que s'il est aisé de retracer les séquences individuelles à partir du graphique des séquences distinctes, la démarche demande un peu plus d'efforts dans le cas du graphique des séquences non-imbriquées. L'utilisateur doit dans ce cas repérer et interpréter les variations

d'épaisseurs des divers segments des trajectoires représentées, ce qui reste très intuitif. C'est là le prix à payer pour avoir un graphique plus synthétique sans perte d'information.

4 Illustrations

Anxiété après un séjour aux soins intensifs. La figure 3 présente les 26 trajectoires distinctes parmi les 149 observées. Les couleurs aident à les distinguer et l'épaisseur des traits reflète la fréquence d'observation des trajectoires, si bien qu'il apparaît immédiatement que, et heureusement, le cas le plus fréquent est celui d'individus éprouvant peu ou pas d'anxiété aux trois termes considérés. Comme il s'agit ici de séquences d'états exclusifs, on n'a pas de simultanéités et donc pas de traits verticaux. Par ailleurs, les états sont ordonnés avec la situation la moins désirable au haut de l'échelle. Comme les courbes tendent plutôt à descendre qu'à monter, cela traduit une baisse de l'anxiété au cours du temps. En particulier, on peut observer une fréquence relativement importante dans le carré nord-ouest correspondant aux patients fortement anxieux après une semaine. En suivant les trajectoires qui partent de ce carré, on voit que la plupart de ces individus se déclarent moins anxieux 6 et 12 mois plus tard, que rares sont ceux qui restent fortement anxieux 6 mois plus tard, et qu'aucun ne se déclare encore fortement anxieux après 12 mois.

Outre les séquences sans anxiété, les trajectoires les plus fréquentes identifiables par l'épaisseur des traits sont [(forte)^{t_1} $\rightarrow$ (faible)^{t_2} $\rightarrow$ faible)^{t_3}], [(moyenne)^{t_1} $\rightarrow$ (faible)^{t_2} $\rightarrow$ (faible)^{t_3}],

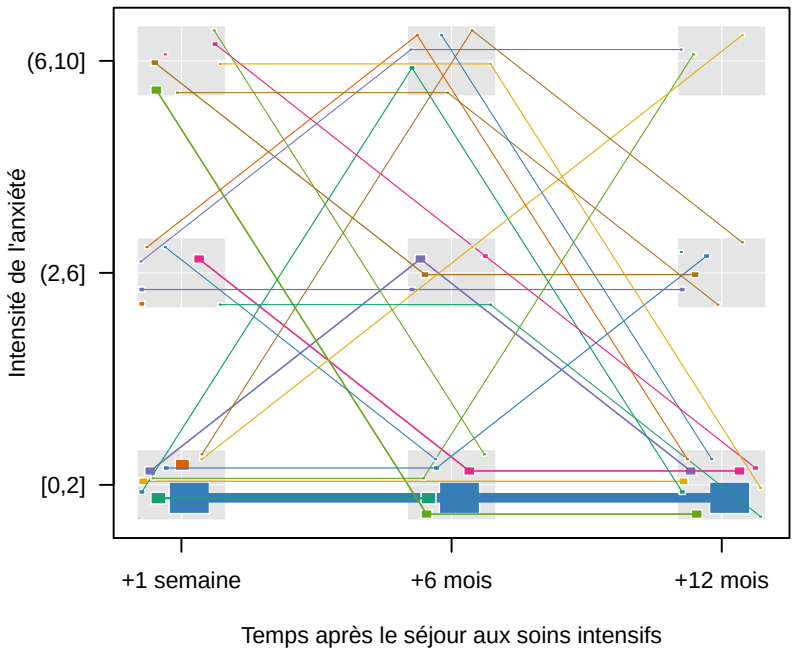


FIG. 3 – *Trajectoires distinctes du niveau d'anxiété de 149 patients.*

et [(faible)^{t_1} $\rightarrow$ (moyenne)^{t_2} $\rightarrow$ (faible)^{t_3}]. Enfin, avec un peu d'attention on peut voir des points-états isolés qui correspondent à des individus qui n'ont répondu qu'une fois, ainsi que des trajectoires de longueur deux, notamment la séquence fréquente [(faible)^{t_1} $\rightarrow$ (faible)^{t_2}], mais aussi une trajectoire plus atypique [(forte)^{t_2} $\rightarrow$ (faible)^{t_3}]. Cette dernière est imbriquée dans la séquence [(forte)^{t_1} $\rightarrow$ (forte)^{t_2} $\rightarrow$ (faible)^{t_3}], et n'apparaîtrait donc pas sous forme distincte dans un graphique des séquences non-imbriquables.

Evénements de vie familiale. Pour notre second exemple, le nombre de trajectoires est beaucoup plus conséquent, c'est pourquoi nous optons pour la représentation des séquences non-imbriquées. La figure 4 montre les 55 trajectoires non-imbriquées de vie familiale identifiables parmi les 2601 trajectoires considérées. A titre de comparaison, le nombre de trajectoires distinctes est de 130 et leur visualisation donnerait un graphique sensiblement plus confus. Le carré noir dans le coin sud-ouest reflète les 52 individus qui n'ont connu aucun événement. Pour faciliter l'identification des séquences les plus communes, les trajectoires dont un segment au moins a un support minimal donné (5% dans notre exemple) sont rendues en couleur, tandis que les trajectoires moins fréquentes apparaissent en gris clair en arrière-plan.

L'examen du graphique révèle que les trajectoires tendent à croître de façon monotone avec la position dans la séquence, ce qui démontre que l'ordre des événements retenu pour l'axe des y est en Suisse la norme, du moins pour les générations nées avant 1958. Deux configurations sont clairement les plus fréquentes : (en bleu) quitter le domicile parental, se mettre en ménage et se marier durant une même année, puis avoir un premier enfant une ou plusieurs années plus tard, et (en vert) quitter d'abord le domicile parental et quelques années plus tard se mettre en ménage et se marier une même année, puis, encore plus tard, avoir le premier enfant. Deux trajectoires un peu moins fréquentes sont de connaître ces quatre événements la même année, ou de les vivre l'un après l'autre. On remarque aussi, que parmi ceux qui ont suivi l'une des quatre trajectoires précédentes, le risque de divorcer est le plus important lorsque l'on expérimente les quatre événements la même année.

De façon non surprenante puisque le divorce doit obligatoirement être précédé d'un mariage, on voit qu'aucune trajectoire ne débute avec le divorce. Plus intéressant est le fait qu'un certain nombre d'individus se marient et/ou ont un premier enfant avant de quitter le domicile parental. On le voit, cette visualisation des séquences permet de rendre compte des normes sociales dans le séquencement des événements de vie tout en informant sur la diversité des situations observées.

5 Discussion

Les illustrations précédentes ont démontré l'utilité de notre représentation graphique pour l'exploration de séquences d'états comme de séquences d'événements. L'intérêt du graphique réside dans sa capacité à visualiser toutes les configurations de trajectoires observées et à faire ressortir clairement les plus courantes d'entre elles.

Nous nous proposons dans cette section, d'une part, d'examiner dans quelle mesure notre représentation graphique répond aux critères de Diggle et al. (2002) que nous avons rappelés en introduction, et d'autre part, de discuter les possibilités d'adapter le graphique aux spécificités des données pour en optimiser le rendu.

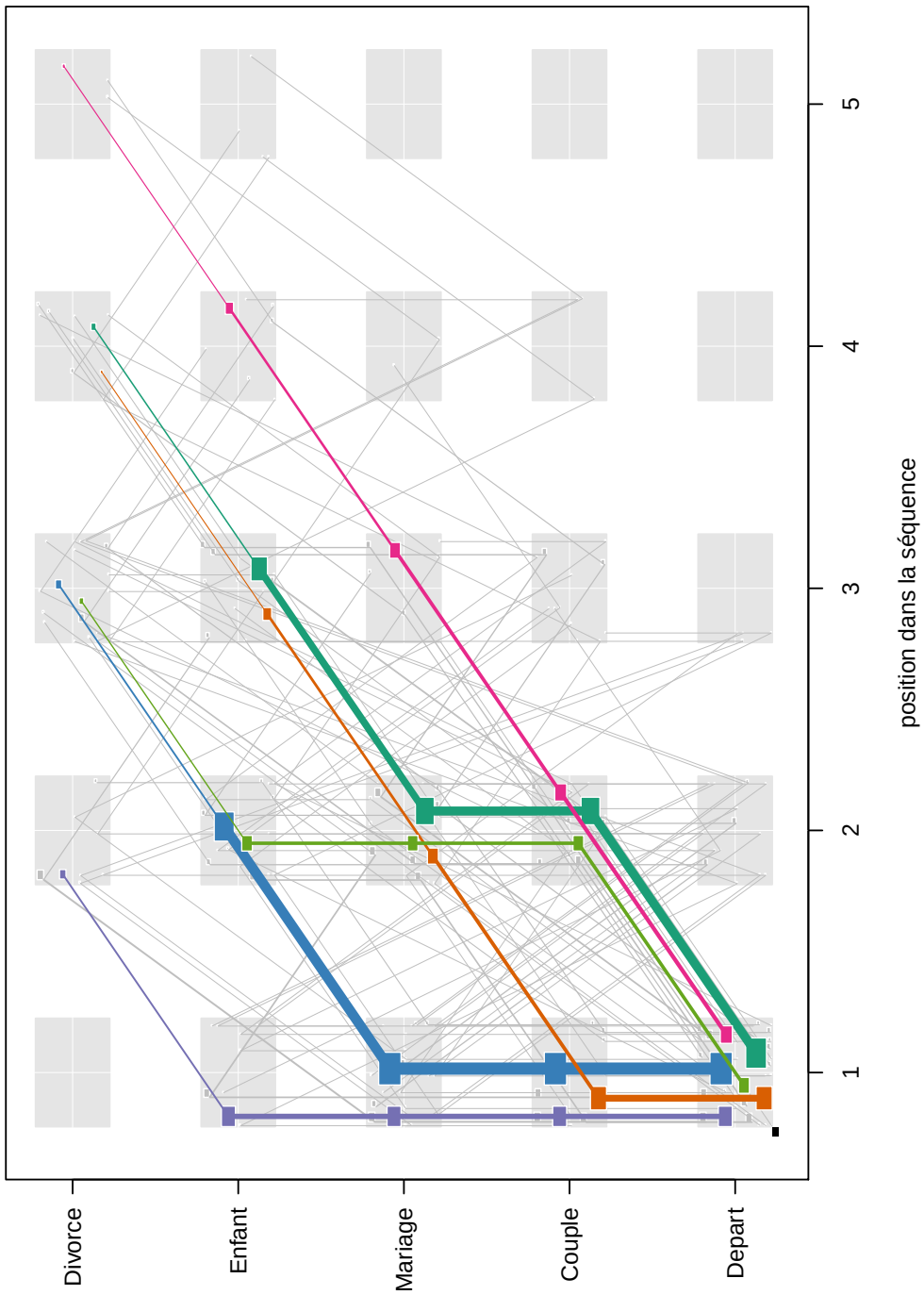


FIG. 4 – *Trajectoires non-imbriquables des 2601 parcours de vie familiale.*

5.1 Critères de Diggle et al. (2002)

Le chapitre 3 de l'ouvrage de Diggle et al. (2002) contient une discussion fort intéressante sur la visualisation de données longitudinales numériques. Les auteurs y énoncent quatre critères qu'un graphique de données longitudinales devrait satisfaire pour être efficient. Bien que notre proposition soit destinée à des données catégorielles plutôt que numériques, il est instructif de l'examiner à la lumière de ces quatre critères.

Montrer les données brutes pertinentes plutôt que des résumés synthétiques. La visualisation proposée conserve l'essentiel de l'information sur les trajectoires individuelles, puisque l'information perdue par l'agrégation des trajectoires identiques ou l'imbrication de trajectoires est restituée par l'épaisseur variable des traits et points correspondants. Il est vrai que telle que présentée la relation entre le graphique et les données n'est pas totalement biunivoque car la figure ne porte pas les identifiants des trajectoires et ne renseigne que de façon relative sur les fréquences des trajectoires distinctes ou segments de trajectoires imbriquées. Si nécessaire, on pourrait afficher ces informations en regard de quelques trajectoires d'intérêt.

Mettre en évidence les modèles de configuration potentiellement intéressants. L'épaisseur différenciée des points et traits permet précisément de distinguer les trajectoires les plus fréquentes. On pourrait évidemment envisager un autre critère d'intérêt que la fréquence, par exemple la centralité (somme des distances par rapport aux autres séquences), et utiliser l'épaisseur des traits pour le refléter.

Identifier les caractéristiques transversales et longitudinales de l'ensemble. Le graphique proposé vise prioritairement à rendre compte des caractéristiques longitudinales. Il visualise cependant aussi la diversité des trajectoires entre individus. De plus, bien que qu'un chronogramme puisse être plus approprié pour cela, il donne également une certaine idée de la distribution transversale des états ou événements en chaque position.

Faciliter l'identification des individus et observations atypiques. A nouveau, c'est ici la taille des points et l'épaisseur des traits qui permet de repérer les trajectoires peu fréquentes : ce sont celles qui sont reproduites avec les traits les plus fins. Comme pour l'identification des trajectoires d'intérêt potentiel, il serait aisé d'utiliser un autre critère que la fréquence pour juger de l'atypicité. Notons cependant que le repérage des lignes les plus fines demande plus d'efforts que l'identification des trajectoires les plus typiques. Notre implémentation prevoit une option pour ne colorier que les trajectoires les moins fréquentes et ainsi mieux les mettre en évidence. Quant à l'identification des individus, on pourrait imaginer d'afficher leur identificateur en regard des trajectoires les moins fréquentes, du moins tant que leur nombre reste limité. L'examen du nombre et de la taille des points représentant à chaque position les états ou événements observés permet d'identifier les états ou événements rares.

Au final, on peut donc affirmer que notre proposition est efficace au sens de Diggle et al. (2002).

5.2 Spécification du graphique et astuces pratiques

En expérimentant la visualisation proposée sur divers jeux de données, nous avons constaté qu'il reste difficile d'extraire des régularités lorsque le nombre de trajectoires distinctes est très

grand. Le résultat est en général d'autant plus brouillé que le nombre d'individus, la taille de l'alphabet des états ou événements, et la longueur possible des séquences sont grands. Plusieurs astuces ou ajustements relativement simples permettent d'améliorer sensiblement les représentations.

En premier lieu, il est souvent très instructif de partitionner les données selon, par exemple, le point de départ ou d'arrivée de la séquence, c'est-à-dire selon le premier ou dernier élément de la séquence. Ceci permet non seulement de réduire le nombre de séquences par graphique, mais aussi la diversité des début ou fin de séquences, ce qui en facilite l'interprétation.

Il est par ailleurs plus facile de repérer des évolutions monotones que des trajectoires irrégulières. Lorsque l'alphabet est nominal, il peut être utile de l'ordonner de sorte à assurer cette monotonie. C'est ce que nous avons fait dans notre illustration avec les événements de vie. Dans le cas d'un alphabet ordinal comme dans notre première illustration, il convient évidemment de respecter l'ordre naturel de l'alphabet.

Des variantes sont possibles également au niveau de l'axe des x. Les options sont ici soit d'aligner les événements ou états sur la position qu'ils occupent dans la séquence, ou alors prendre en compte explicitement la date ou la durée du processus (âge) et aligner sur la date ou la durée. Dans notre seconde illustration, nous avons délibérément ignoré l'information sur les dates d'occurrence des événements en raison de notre intérêt sur leur séquencement plutôt que sur les moments de la vie où ils se réalisent. La prise en compte des âges produit un graphique nettement moins lisible.

Un dernier facteur d'ajustement est lié au décalage des diverses trajectoires ou si l'on veut au positionnement des points dans la zone de translation. Ces positions sont actuellement déterminées aléatoirement mais une idée serait de les choisir de façon à minimiser les croisements.

6 Conclusion

La procédure d'exploration visuelle de données séquentielles proposée dans cet article vient complémenter les nombreux outils développés ces dernières années dans le domaine de la fouille de séquences et de l'analyse de données longitudinales. L'intérêt de l'outil, qui s'applique aussi bien à des séquences d'événements qu'à des séquences d'états, réside dans l'aide qu'il fournit à l'interprétation et à sa capacité à rendre compte simultanément des tendances principales et de la diversité entre individus.

Les démographes et sociologues qui ont eu l'occasion d'expérimenter le graphique sur leurs données y ont vu un outil d'emploi facile, de lecture simple et ouvrant des perspectives nouvelles en particulier pour l'étude de la déstandardisation des parcours de vie. Le graphique sera prochainement disponible dans le cadre de la librairie R TraMineR (Gabadinho et al., 2011).

Les développements futurs devraient permettre d'automatiser l'optimisation du placement des diverses trajectoires dans la zone de translation ainsi que de l'ordre de l'alphabet, ou encore l'ordre de présentation d'événements simultanés.

Références

Agrawal, R. et R. Srikant (1995). Mining sequential patterns. In P. S. Yu et A. L. P. Chen (Eds.), *Proceedings of the International Conference on Data Engeneering (ICDE), Taipei, Taiwan*, pp. 487–499. IEEE Computer Society.

Berchtold, A. et A. E. Raftery (2002). The mixture transition distribution model for high-order Markov chains and non-gaussian time series. *Statistical Science 17*(3), 328–356.

Diggle, P. J., P. Heagerty, K.-Y. Liang, et S. L. Zeger (2002). *Analysis of Longitudinal Data* (2nd ed.). Oxford: Oxford University Press.

Frees, E. W. (2004). *Longitudinal and Panel Data: Analysis and Applications in the Social Sciences*. New York: Cambridge University Press.

Gabadinho, A., G. Ritschard, N. S. Müller, et M. Studer (2011). Analyzing and visualizing state sequences in R with TraMineR. *Journal of Statistical Software 40*(4), 1–37.

Hébrail, G. et H. Cadalen (2000). Visualisation et classification automatique de parcours professionnels. In *Actes des XXXIIe Journées de statistique, Fès, Maroc*, pp. 458–462.

Hox, J. (2010). *Multilevel Analysis: Techniques and Applications* (2nd ed.). Abingdon UK: Routledge.

Jeitziner, M.-M., V. Hantikainen, A. Conca, et J. Hamers (2011). Long-term consequences of an intensive care unit stay in older critically ill patients: Design of a longitudinal study. *BMC Geriatrics 11*(52), 1–7.

Masseglia, F. (2002). *Algorithmes et applications pour l'extraction de motifs séquentiels dans le domaine de la fouille de données : de l'incrémental au temps réel*. Ph. D. thesis, Université de Versailles Saint-Quentin en Yvelines.

McArdle, J. J. (2009). Latent variable modeling of differences and changes with longitudinal data. *Annual Review of Psychology 60*, 577–605.

Studer, M., N. S. Müller, G. Ritschard, et A. Gabadinho (2010). Classer, discriminer et visualiser des séquences d'événements. *Revue des nouvelles technologies de l'information RNTI E-19*, 37–48.

Tufte, E. R. (2001). *The Visual Display of Quantitative Information* (2nd ed.). Cheshire CT: Graphics Press.

Wongsuphasawat, K., J. A. G. Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, et B. Shneiderman (2011). LifeFlow: Visualizing an overview of event sequences. In *Proceedings of the 2011 annual conference on Human Factors in Computing Systems (CHI), Vancouver, Canada, May 7-12, 2011*, pp. 1747–1756. New York: ACM.

Summary

The article introduces an original graphical display for categorical longitudinal data. The visualisation, inspired from the multiple time-series plot, particularly suits to descriptive and exploratory analyses of individual trajectories defined as event sequences. The article includes a description of the visualisation method and of its founding principles, application examples, and a discussion of the properties of the resulting plots. In addition, we explain fine-tuning specifications for optimally rendering given data.

FOLKOVIZ: Visualisation socio-sémantique de folksonomies

A. MOUAKHER, M. DAOUD, S. BEN YAHIA

*Faculté des Sciences de Tunis, Université de Tunis-El Manar, Tunis, Tunisie.
sadok.benyahia@fst.rnu.tn

Résumé. L'essor des sites collaboratifs sur Internet relevant du mouvement participatif que l'on désigne souvent du nom de Web 2.0 a permis la naissance de nouvelles formes d'indexations des contenus du Web créées librement par les usagers et partagés au sein de réseaux sociaux, baptisées sous le nom de folksonomies. Dans ce papier, nous introduisons l'outil FOLKOVIZ pour la Visualisation socio-sémantique de *folksonomies*. Cet outil opère en deux phases :(*i*) Extraction et désambigüisation des concepts triadiques, qui constituent une représentation concise des données folksonomiques; (*ii*) Visualisation 3D en utilisation la métaphore "Ville" avec incorporation de la dimension sociale via les ontologies SIOC et FOAF.

1 Introduction et motivations

Dans les dernières années, le développement impressionnant de l'Internet a renforcé l'apparition d'une énorme quantité d'informations disponibles sur le Web. Cette croissance d'information s'est accompagnée d'une évolution au niveau du web qui est devenu une véritable plate-forme bidirectionelle. Ainsi, dans ce cadre, le social tagging (ou étiquetage social) s'est récemment imposé dans le paysage du web social et collaboratif (Web 2.0) comme un support à l'organisation de ressources partagées en permettant aux utilisateurs de catégoriser leurs ressources en leurs associant des mots clefs, appelés *tags*. Dans de tels environnements, les utilisateurs ont la possibilité d'annoter leurs ressources (pages web, photos, etc.) par le biais de tags. La structure, ainsi créée, est appelée *folksonomie* et peut être considérée comme un hypergraphe tripartite d'utilisateurs, de tags et de ressources. Les sites de folksonomie permettent une exploration plus ouverte et hasardeuse du contenu que les moteurs de recherche. De plus, le vocabulaire évolue parfois rapidement et la folksonomie permet de refléter en temps réel cette évolution. Cette structure spécifique aux folksonomies a rendu leur exploitation d'un grand intérêt pour la recherche d'information dans la mesure où elles permettraient d'identifier et de surveiller l'émergence de nouveaux concepts. Cependant, les moteurs de recherche, qui représentent un passage incontournable pour la recherche d'information sur le Web, ne sont pas adaptés à l'exploration de ces folksonomies. En effet, un certain nombre de contraintes sont à prendre en considération :

1. Les tags sont librement choisis, reflétant les choix des utilisateurs en termes de diction, terminologie et précision, pour l'indexation. Ceci nécessite un traitement adéquat de l'ambiguïté et de la redondance.

2. La taille de la structure induite et le nombre d'utilisateurs sont en plein croissance. Ceci induirait que le passage à l'échelle, en termes de stockage et de temps de réponse, serait une condition *sine qua non* dans tout système de visualisation de folksonomies.

Dans ce papier, nous introduisons l'outil FOLKOVIZ dont l'objectif est d'offrir une visualisation socio-sémantique de données folksonomiques. Ainsi, cet outil se distinguerait de la recherche d'information classique dans la mesure où l'ambition serait de proposer, en réponse à une requête isolée d'un utilisateur, un itinéraire que l'utilisateur devrait parcourir et qui lui donnerait "chemin faisant" les éléments d'une réponse "didactique" à sa question. l'outil FOLKOVIZ opère en deux phases. La première dite en ligne permettra d'extraire une représentation concise des données folksonomiques (dans un souci de passage à l'échelle). Ensuite, une étape de désambigüisation est effectuée pour déterminer la "sémantique" des concepts triadiques extraits. Cette sémantique serait par la suite utilisée dans la phase de visualisation, utilisant la métaphore ville en 3D", pour aider l'utilisateur à bien choisir le "carrefour" qui correspond à son intention de recherche et à visualiser les ressources y afférentes.

Le reste du papier est organisé comme suit. Dans la section 2, nous présentons brièvement les folksonomies ainsi que leurs fondements mathématiques. Ensuite, nous allons présenter dans la section 3, un survol de l'état de l'art sur les métaphores de visualisation 3D. La section 4 est dédiée à une présentation détaillée de l'outil FOLKOVIZ. La dernière section conclut ce papier et dresse les perspectives de recherche des travaux en cours.

2 Folksonomies

Dans cette section, nous allons présenter brièvement quelques notions relatives aux folksonomies.

Definition 1 (CONTEXTE TRIADIQUE) *Selon Ganter et Wille (1999), un contexte triadique d'extraction (ou un contexte d'extraction) est un quadruplet* $\mathcal{K} = (\mathcal{E}, \mathcal{I}, \mathcal{C}, \mathcal{Y})$*, où* $\mathcal{E}$*,* $\mathcal{I}$ *and* $\mathcal{C}$ *sont des ensembles, et* $\mathcal{Y}$ *est une relation ternaire entre* $\mathcal{E}$*,* $\mathcal{I}$ *et* $\mathcal{C}$*, i.e.,* $\mathcal{Y} \subseteq \mathcal{E} \times \mathcal{I} \times \mathcal{C}$*. Les éléments de* $\mathcal{E}$*,* $\mathcal{I}$ *et* $\mathcal{C}$ *sont respectivement appelés objets, attributs, et conditions et* $(e, i, c) \in \mathcal{Y}$*, sous-entend que l'objet* e *est relatif à l'attribut* i *relativement à la condition* c*.*

Un contexte triadique représente exactement la structure d'une *folksonomie* dont la définition est la suivante :

Definition 2 (FOLKSONOMIE) *Hotho et al. (2006) ont définit une folksonomie comme étant un ensemble de tuples* $\mathcal{F} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$ *où*

- $\mathcal{U}$*,* $\mathcal{T}$ *et* $\mathcal{R}$ *sont des ensembles finis dont les éléments sont appelés utilisateurs, tags et ressources.*
- $\mathcal{Y} \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{R}$ *représente une relation triadique dont chaque* $y \subseteq \mathcal{Y}$ *peut être représenté par un triplet :*

$$y = \{(u, t, r) \mid u \in \mathcal{U}, t \in \mathcal{T}, r \in R\}.$$

ce qui signifie que l'utilisateur u *a annoté la ressource* r *par le tag* t*.*

Une *folksonomie* est également appelée *"Social Tagging"*, un processus par lequel de nombreux utilisateurs ajoutent des données sous forme de tags pour partager des ressources. En d'autres termes, il s'agit d'un support du Web 2.0 pour la classification de ressources. L'annotation des ressources par les tags facilite ainsi le partage et la recherche de l'information, e.g., DEL.ICIO.US, FLICKR, CONNOTEA, YOUTUBE, etc.

Example 1 *Le Tableau 1 illustre un exemple d'une folksonomie $\mathcal{F}$ avec $\mathcal{U}$ = {u_1, u_2, u_3, u_4, u_5, u_6, u_7}, $\mathcal{T}$ = {t_1, t_2, t_3, t_4, t_5} et $\mathcal{R}$ = {r_1, r_2, r_3}.*

Notons que chaque "$\times$" représente une relation triadique entre un utilisateur appartenant à $\mathcal{U}$, un tag appartenant à $\mathcal{T}$ et une ressource annotée appartenant à $\mathcal{R}$. Par exemple, l'utilisateur u_1 a taggé la ressource r_1 par le biais des tags t_2, t_3 et t_4.

U/R-T	r_1					r_2					r_3				
	t_1	t_2	t_3	t_4	t_5	t_1	t_2	t_3	t_4	t_5	t_1	t_2	t_3	t_4	t_5
u_1		×	×	×			×	×	×			×	×	×	
u_2		×	×	×		×	×	×	×		×	×	×	×	
u_3		×	×	×		×	×	×	×		×	×	×	×	
u_4						×			×		×			×	
u_5		×	×	×	×		×	×	×	×		×	×	×	
u_6				×	×				×	×					
u_7	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×

TAB. 1 – *Une folksonomie.*

Definition 3 (CONCEPT TRIADIQUE (FRÉQUENT)) *En adaptant au cas triadique la notion d'*"itemsets fermés fréquents" *introduite par Agrawal et al. (1993), nous pouvons définir un concept triadique (ou un tri-concept) d'une* folksonomie *$\mathcal{F}$ = ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{Y}$) comme un triplet (U, T, R) où $U \subseteq \mathcal{U}$, $T \subseteq \mathcal{T}$, et $R \subseteq \mathcal{R}$ avec $U \times T \times R \subseteq \mathcal{Y}$ tel que le triplet (U, T, R) est maximal, i.e., pour $U_1 \subseteq U$, $T_1 \subseteq T$ et $R_1 \subseteq R$ où $U_1 \times T_1 \times R_1 \subseteq \mathcal{Y}$, les ensembles $U \subseteq U_1$, $T \subseteq T_1$, et $R \subseteq R_1$ impliquent toujours (U, T, R) = (U_1, T_1, R_1). Un tri-concept est dit fréquent lorsqu'il est un tri-set fréquent. L'ensemble des tri-concepts de $\mathcal{K}$ est représenté par $\mathcal{TC}_\mathcal{K}$ = {$\mathcal{TC}_i \mid \mathcal{TC}_i$ = (U, T, R) $\in \mathcal{Y}$ est un tri-concept, $i = 1 \ldots n$} où n est le nombre de tri-concepts extraits à partir de $\mathcal{F}$.*

Example 2 *D'après la* folksonomie *du Tableau 1, TC_1={{u_5, u_7}, {t_2, t_3, t_4}, {r_1, r_2, r_3}} est un tri-concept de $\mathcal{F}$: c'est l'ensemble maximal de tags et de ressources partagées par u_5 et u_7.*

Dans ce qui suit, nous allons passer en revue les principales métaphores de visualisation 3D ainsi que leurs principales cadres applicatifs.

3 Les métaphores de visualisation : état de l'art

Selon Card et al. (1999) et Spence (2001), la visualisation d'information consiste à représenter des informations abstraites sous forme visuelle afin d'optimiser la recherche d'information. Elle permet d'exploiter les caractéristiques du système visuel humain pour faciliter la manipulation et l'interprétation des données qui ont tendance à être abstraites. En effet, la recherche d'information est une activité complexe marquée par une surcharge informationnelle et cognitive. L'utilisation des métaphores [1] est alors appropriée afin de concrétiser les informations. Par ailleurs, Bonnel et al. (2005a) affirment que l'utilisation de la 3D est plus proches de

1. Une métaphore consiste à exprimer les concepts d'un domaine dans les termes d'un autre domaine, plus compréhensible et plus familier pour l'utilisateur (Simoff, 2001).

l'esprit humain du point de vue cognitif, offre de nouvelles possibilités d'interactions et facilite la compréhension du système à étudier. En outre, avec la 3D, on peut bénéficier de plus de propriétés graphiques pour les objets et donc de représenter un maximum d'information (Blanchard et al., 2003). Dans ce qui suit, nous allons passer en revue les principales métaphores 3D.

1. **La métaphore système solaire** : Cette métaphore utilise un gestionnaire de disposition orbitale pour placer des éléments dans une orbite autour d'un centre spécifié, de la même façon dont les satellites sont placés autour des planètes. Abel et al. (2000) proposent, dans leur projet *Cybernet*, plusieurs métaphores pour la visualisation des réseaux. Ces derniers ont utilisé la métaphore du système solaire pour la supervision de la performance des stations de travail. Santos et al. (2000) ont utilisé cette métaphore pour aider l'utilisateur à mieux comprendre les informations concernant l'état d'un réseau.Par ailleurs, Graham et al. (2004) ont exploité cette même métaphore pour représenter les métriques d'un logiciel afin d'étudier son évolution dans le temps, d'analyser son code source et de repérer les zones soupçonnées de risques (c.f., figure 2(b)). Récemment, Bloom Studio ont développé le projet *Planetary* [2], qui consiste en un lecteur de musique visuel conçu pour les *iPad*. *Planetary* permet d'explorer la collection de musique dans un univers en 3D crée dynamiquement par des informations sur les artistes.(c.f., figure 2(a)).
2. **La métaphore des arbres coniques** : ces dernières désignent une représentation tridimensionnelle d'une hiérarchie dans laquelle on associe à chaque noeud le sommet d'un cône, et on arrange ses fils autour de la base circulaire du cône. Bonnel et al. (2005b) a affirmé que cette métaphore est l'une des techniques les plus utilisées pour afficher des arbres en 3D. Le but de cette métaphore est de présenter la structure de façon telle que toute la hiérarchie ou du moins la majeure partie puisse être visualisée sans que l'utilisateur ait besoin d'utiliser les barres de defilement. Cependant, elle souffre d'une limitation en termes de nombre de noeuds affichables. L'un des exemples qui ont utilisé cette métaphore autre est *Cat-aCone* Hearst et Karadi (1997), qui utilise une visualisation 3D d'un arbre conique afin d'afficher simultanément les résultats de recherche obtenus et une hiérarchie de catégories. Dans le projet *Cybernet*, Abel et al. (2000) ont utilisé les arbres coniques 3D pour visualiser la structure hiérarchique d'un système (c.f.,figure 2(c)).
3. **La métaphore du paysage naturel** : elle consiste à représenter l'information par des objets posés sur un sol afin d'optimiser leur visibilité et de diminuer les occlusions. Diverses applications ont montré l'efficacité de cette métaphore pour l'exploration de grandes quantités d'information. A titre d'exemple, on peut citer l'explorateur visuel des réseaux bibliographiques conçu par Brandes et Willhalm (2002). En effet, grâce à cette métaphore, cette approche a facilité l'identification des clusters ainsi que les entités les plus importantes de ces réseaux. D'un autre coté, Blanchard et al. (2003) ont exploité cette métaphore à travers leur outil *ArVis* dédié à la fouille anthropocentrée de règles d'association (c.f., figure 2(d)). Par ailleurs, Fabrikant et Skupin (2005) soulignent que cette métaphore doit être accompagnée par une conception cartographique des lignes directives pour garantir à l'utilisateur une meilleur compréhension. Récemment, Fabrikant et al. (2010) ont examiné les hypothèses mises en place par cette métaphore pour

2. Cette application est disponible sur : *http ://planetary.bloom.io/.*

permettre aux non-spécialistes d'accéder exactement à l'information désirée et ceci dans le cadre de la recherche dans les archives des documents en ligne (c.f., figure 2(e)).

4. **La métaphore ville** : peut être vue comme une métaphore de la vie réelle dans laquelle l'utilisateur peut facilement utiliser sa capacité de perception Anslow et al. (2006). Elle fût utilisée, dans le système *MediaMetro* par Chiu et al. (2005), pour la visualisation interactive des documents multimedia. Les auteurs ont utilisé une nouvelle technique de navigation pour cette métaphore afin de faciliter à l'utilisateur la localisation des documents. Par ailleurs, Bonnel et al. (2005b) a utilisé la même métaphore pour la visualisation des résultats de recherche, avec l'objectif d'améliorer l'efficacité de la restitution des résultats. A noter aussi les travaux de Panas et al. (2003); Wettel (2008) qui ont aussi utilisé la métaphore de la ville en présentant des travaux sur la visualisation des logiciels. Le principe de cette métaphore permet aux utilisateurs de transférer des connaissances sur un sujet qu'ils maîtrisent bien (la ville) vers un sujet qui leur est nouveau (la qualité du logiciel).

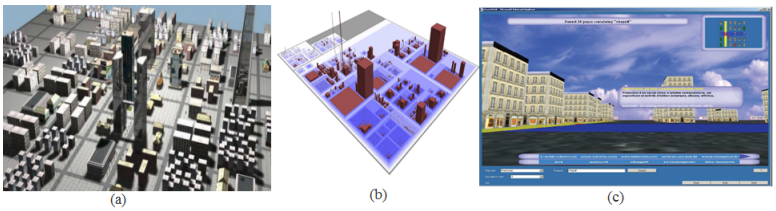


FIG. 1 – *(a) : Le prototype proposé par Panas et al. (2003), (b) : ArgoUML visualisé via* CodeCity*, (c) : Visualisation des résultats de recherche selon le système de Bonnel et al. (2005b)*

Dans la section suivante, nous allons présenter les différentes phases de l'approche FOLKOVIZ.

4 Présentation de l'approche FOLKOVIZ

Dans cette section, nous allons passer en revue notre système de visualisation de données folksonomiques FOLKOVIZ. Comme le met en exergue, la figure 3, FOLKOVIZ opère en deux phases expliquées dans ce qui suit.

4.1 Phase hors ligne

Cette phase opère en étapes décrites dans ce qui suit.

1. **Étape 1 : Extraction d'une représentation concise des données folksonomiques** : Dans cette étape, nous partons d'une folksonomie donnée et nous extrayons tous les tri-concepts en appliquant l'algorithme TRIAS, introduit par Jäschke et al. (2006). L'ensemble des tri-concepts constitue une représentation concise des données folksonomiques. Ensuite, nous procédons à un regroupement des tri-concepts similaires en augmentant le nombre de tags. Cette approche a été proposée par Trabelsi et al. (2011). Ainsi, nous procédons à la fusion des tri-concepts "sémantiquement" similaires afin d'accroître leur

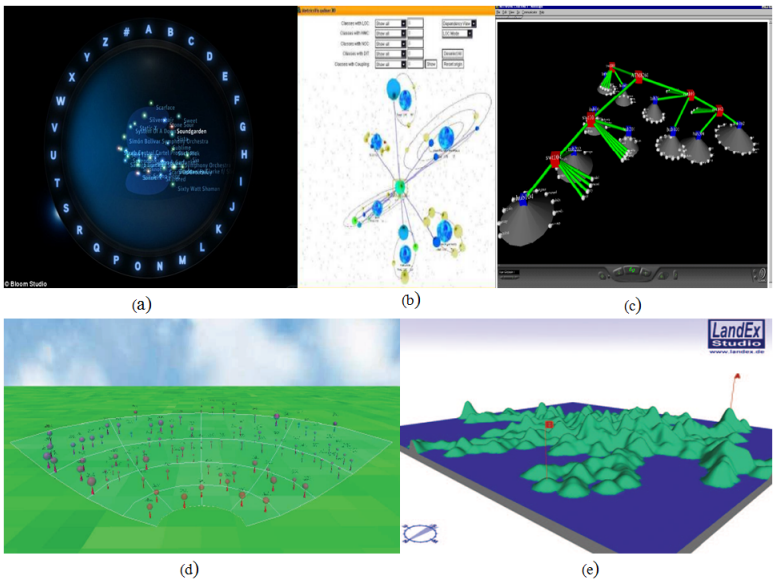


FIG. 2 – *Exemples de métaphores 3D*

modus (l'ensemble de tags). Tout d'abord, l'algorithme commence par comparer des groupes d'utilisateurs de deux tri-concepts donnés et dans le cas où ces derniers sont similaires, il passe à la comparaison des ressources des tri-concepts. Dans cet algorithme, les auteurs utilisent deux mesures de similarité : La similarité entre groupes d'utilisateurs introduite par Yin et al. (2010) et qui est donnée comme suit :

$$Sim(\mathcal{V}_{U_1}, \mathcal{V}_{U_2}) = \frac{\mathcal{V}_{U_1} \times \mathcal{V}_{U_2}}{|\mathcal{V}_{U_1}| \times |\mathcal{V}_{U_2}|}$$

où le vecteur $\mathcal{V}_{U_k} = \{(t_1, n_{U_k}), (t_2, n_{U_k}), \ldots, (t_i, n_{U_k})\}$ et (t_i, n_{U_k}) représente le nombre de fois que le groupe d'utilisateurs U_k a annoté avec le tag t_i.

La deuxième mesure de similarité concerne les ensembles de ressources et se base sur la *PLSC* normalisée. Cette mesure, utilisée par Bach (2006), permet de calculer la longueur de la plus longue sous-chaîne commune entre r_i et r_j, i.e., :

$$Sim_2(r_i, r_j) = \frac{PLSC(r_i, r_j)}{|r_i|}$$

2. **Étape 2 :Désambiguisation** : dans cette étape, nous allons identifier la sémantique associée aux tags. Diverses études ont été menées sur cette problématique(e.g, Garcá-Silva et al. (2009), Benz et al. (2010)). Dans FOLKOVIZ, nous avons utilisé l'algorithme, proposé par Garcá-Silva et al. (2009), qui utilise *Tagora Sense Repositoy* [3] comme un dictionnaire où les tags sont reliés directement aux concepts de *Dbpedia* [4] et les pages de

3. Ce dictionnaire est fourni à : *http ://tagora.ecs.soton.ac.uk/tsr/index.html.*
4. *http ://dbpedia.org/About.*

Wikipedia [5] afin d'extraire les différents sens d'un tag. Par la suite, pour chaque sens, ils identifient les différents termes qui lui sont associés. Une fois que le vocabulaire d'un tag est construit à travers l'union de ces ensembles, ils définissent pou ce vocabulaire un vecteur contexte et un vecteur sens pour chaque sens. Enfin, les auteurs comparent ces deux vecteurs en se basant sur la mesure de similarité pour pouvoir décider lequel, des sens associés au tag, est le plus probable pour définir la sémantique du tag donné.

4.2 Phase en ligne

Le but de notre travail est de visualiser des données folksomiques, que nous avons ramené à la visualisation des concepts triadiques. Dans le domaine, Web2.0 et au meilleur de notre connaissance, aucun des travaux existants ne s'est intéressé à la visualisation des concepts triadiques. Pour notre système FOLKOVIZ, nous avons adopté la métaphore ville. En effet, étant donnée une métaphore du monde réel, les utilisateurs connaissent déjà les stratégies de navigation et peuvent facilement récupérer une information particulière. (Wettel et al., 2011) ont affirmé, après l'évaluation de leur outil *CodeCity*, que la visualisation et l'utilisation de la métaphore ville ont abouti à des résultats meilleurs au niveau de l'exactitude ainsi que le délai de traitement de l'information. En outre, sa structure spatiale est adaptée à notre problématique dans le sens où elle permet la visualisation d'une grande quantité de données en respectant les proximités sémantiques.

Ainsi, il s'agirait d'une ville 3D virtuelle organisée par quartiers et dans laquelle les bâtiments représentent les tri-concepts. Comme illustrée par la figure 4, nous pouvons définir les éléments de cette ville comme suit :

- **Le quartier** : est l'ensemble des ressources présentes dans un tri-concept.
- **Le bâtiment** : le bâtiment représente une ressource avec comme hauteur le nombre de tag utilisés. Chaque bâtiment peut être représenté par des étages dont chacun représente un tag parmi ceux qui figurent dans le tri-concept.
- **Les paraboles** : l'ensemble des utilisateurs qui ont annoté les ressources seront des paraboles en dessous de l'ensemble des bâtiments.
- **Les routes** : seront des chemins qui relient les bâtiments qui sont tagués par des tags similaires. D'où la notion des quartiers voisins. Ceci permet de nous indiquer que deux quartiers voisins contiennent des ressources similaires (des ressources qui ont été taguées par des tags similaires).
- **Les rond-points** : ils indiquent à l'utilisateur le chemin vers un contexte bien déterminé. En effet, un tag peut avoir plusieurs sens, et un tag peut être représenté par plusieurs termes.

Il importe de mettre en exergue la dimension "sociale" de la visualisation proposée par FolkoViz. En effet, le clik sur une parabole permettrait de visualiser l'utilisateur en question et d'en avoir d'amples information sur son réseau social grâce à l'exploration du vocabulaire FOAF (Friend-Of-A-Friend) l'exploration du vocabulaire `foaf`, e.g., `foaf:name`, `foaf:knows`, `foaf:based_near`, `foaf:topic_interest`. Par ailleurs, la connexion à l'application du web sémantique SIOC (Semantically-Interlinked Online Communities) permettrait de découvrir son activité sociale, en le pistant sur les blogs, des forums, des wikis,

5. *http ://www.wikipedia.org/.*

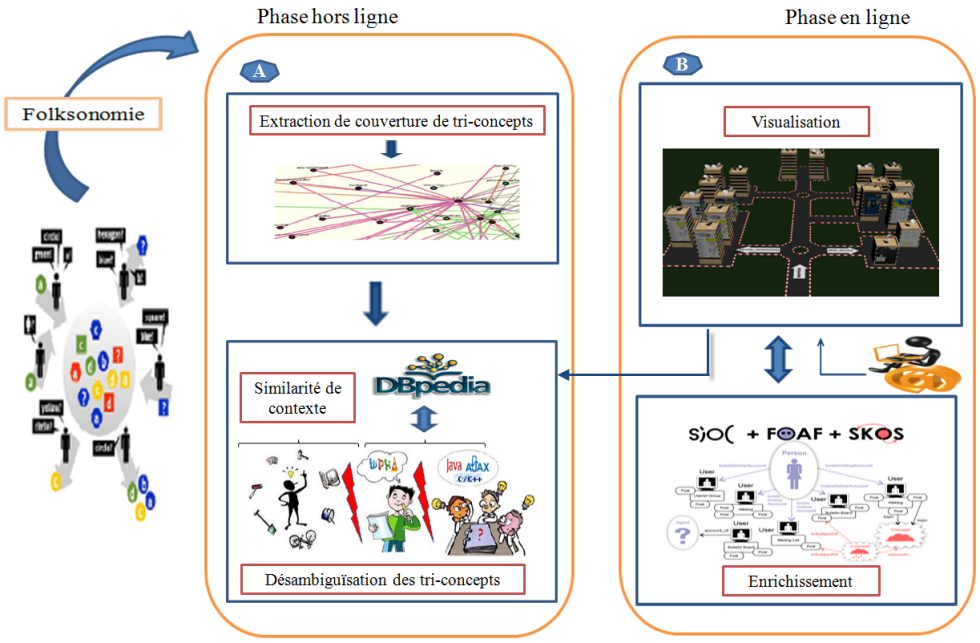


FIG. 3 – *Les différentes phases effectuées par* FOLKOVIZ

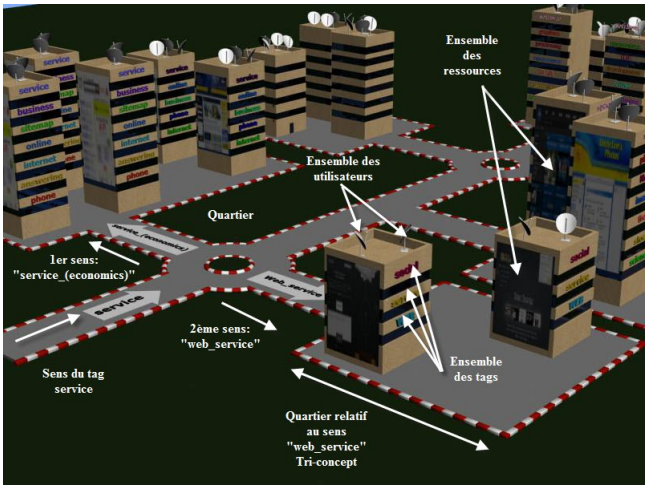


FIG. 4 – *Description de la métaphore proposée*

etc. A cet égard, l'utilisation du concept de OpenID serait d'une grande utilité pour la garantie d'une large exploration de la dimension sociale de l'utilisateur.

5 Exemple illustratif

Nos différentes expérimentations ont été menées sur tri-concepts fréquents extraits d'un jeu de données collecté à partir du système de marque-pages social DEL.ICIO.US et dont la taille réelle s'élève à 10 MB [6] (compressée).

Par définition, deux entités sont considérées comme similaires si leur valeur de similarité dépasse un seuil prédéfini. Dans notre exemple, les seuils de similarité de Sim_1 et Sim_2 sont respectivement fixés à $0,7$ et $0,65$.

	U	{Angie, Lynavo}
TC_1	T	{service, sitemap, answering, phone}
	R	{http ://www.answeringserviceonline.com/answer-service/, http ://www.answeringserviceonline.com/telephone-answering-service/, http ://www.answeringserviceonline.com/answering-service-uk/}
	U	{Krycek, Laurini, Shepard}
TC_2	T	{service, online, business, phone, internet}
	R	{http ://www.easinghelp.info/internet-answering-service/, http ://www.easingassistance.info/telephone-answering-services/}
	U	{Jacob, Brson}
TC_3	T	{service, web, social}
	R	{http ://posterous.com/, http ://pinboard.in/}

TAB. 2 – *Exemples de tri-concepts fréquents extraits à partir de* DEL.ICIO.US.

Tout d'abord, nous procédons à la fusion des tri-concepts similaires et ceci s'effectue par le calcul de similarité entre les groupes d'utilisateurs. La figure 5 représente les deux vecteurs associés à chacun des groupes d'utilisateurs de TC_1 et TC_2, qui correspondent aux différents tags assignés aux différents ressources de TC_1 et TC_2 dans la *folksonomie* $\mathcal{F}$. D'un côté, ces vecteurs vérifient les conditions de similarité étant donné que :

$Sim_1(\mathcal{V}_{U_1}, \mathcal{V}_{U_2}) = \frac{|\mathcal{V}_{U_1} \cap \mathcal{V}_{U_2}|}{\frac{\mathcal{V}_{U_1} \times \mathcal{V}_{U_2}}{|\mathcal{V}_{U_1}| \times |\mathcal{V}_{U_2}|}}$ = 0, 71 $\geq$ 0, 70.

D'un autre côté, ces deux tri-concepts, TC_1 et TC_2, vérifient les conditions de similarité entre les ressources puisque nous avons :

PLSC(**http ://www.a**nswer**ings**ervi**c**eo**nline**.com**/answer-service/**,
http ://www.ea**sing**help.**in**fo/**in**ternet-**ans**wer**ing**-service/**) = 37 $\Rightarrow$ Sim_2 = $\frac{37}{54}$ = 0, 68 $\geq$ 0, 65.

PLSC(**http ://www.a**nswer**ings**ervi**ce**o**n**line.com**/telephone-answering-service/** ,
http ://www.easingassistan**ce**.i**nfo/telephone-answering-service**s**/**) = 39 $\Rightarrow$ Sim_2 = $\frac{51}{66}$ = 0, 77 $\geq$ 0, 65.

Ainsi, nous pouvons fusionner l'ensemble des tags de ces deux derniers. En conséquence, le premier tri-concept devient comme suit :

6. Le jeu de données est librement téléchargeable à l'adresse : *http ://data.dai-labor.de/corpus/delicious/*

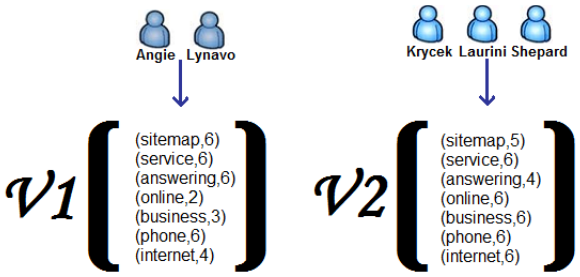


FIG. 5 – *Les Vecteurs correspondant aux différents groupes d'utilisateurs.*

TC_1 = {{Angie, Lynavo}, {service, sitemap, answering, phone, online, business, internet}, {http ://www.answeringserviceonline.com/answer-service/,
http ://www.answeringserviceonline.com/telephone-answering-service/,
http ://www.answeringserviceonline.com/answering-service-uk/}}.

Par la suite, nous passons à l'étape de désambigüisation afin de définir la sémantique des tags. Nous commençons par identifier les différents sens des tags ainsi que les termes qui lui sont associés pour construire le vocabulaire. Finalement, nous pouvons conclure que, pour TC_1, le sens le plus probable du tag service est *dbpedia :Service_(economics)*. Pout TC_3, le sens le plus probable du tag service est *dbpedia : Web_service*. Cette différence au niveau des sens du même tag est illustrée par la figure 6 par le rond-point qui diverge vers deux sens opposés qui sont : *Service_(economics)* et *Web-service*.

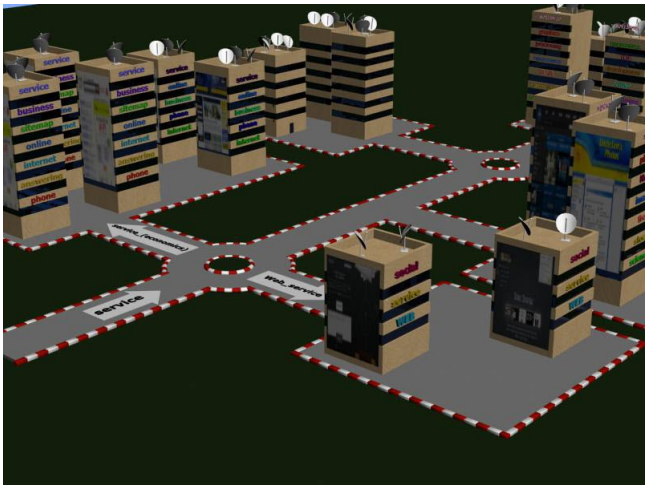


FIG. 6 – *Exemple illustratif*

6 Conclusion et perspectives

Dans ce papier, nous avons proposé l'outil FolkoViz pour une visualisation socio-sémantique des données folksonomiques. Au delà de l'évaluation de l'interface de visualisation, qui déjà en cours de développement nous comptons nous intéresser au cadre théorique de la maintenance incrémentale de la couverture des données folksonomiques. Une meilleure prise en considération du profil de l'utilisateur sera aussi à considérer. Dans ce cadre, une catégorisation des différents types des utilisateurs, empruntée du domaine du E-learning, pourrait être mise à profit pour une adéquation visualisation-utilisateur.

Références

Abel, P., P. Gros, D. Loisel, et C. R. D. Santos (2000). Cybernet : A framework for managing networks using 3d metaphoric worlds. *Annales des télécommunications 55*(3-4).

Agrawal, R., T. Imielinski, et A. Swami (1993). Mining association rules between sets of items in large databases. In *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, Washington D. C., USA, pp. 207–216.

Anslow, C., S. Marshall, et J. Noble (2006). X3d-earth in the software visualization pipeline. In *Proceedings of the X3D Earth Requirements Workshop*, Monterey, California, USA.

Bach, T. L. (2006). *Construction d'un web sémantique multi-points de vue*. Dissertation de doctorat d'état en sciences, École des Mines de Nice, Sophia Antipolis, France.

Benz, D., A. Hotho, et G. Stumme (2010). Semantics made by you and me : Self-emerging ontologies can capture the diversity of shared knowledge. In *Proceedings of the 2nd Web Science Conference (WebSci10)*, Raleigh, NC, USA.

Blanchard, J., F. Guillet, et H. Briand (2003). Une visualisation orientée qualité pour la fouille antropocentrée de règles d'association. *Cognito, Cahiers romans de sciences cognitives 1*(3), 79–100.

Bonnel, N., A. Cotarmanac'H, et A. Morin (2005a). Gestion et visualisation des résultats d'une requête. In *Proceedings of the 3rd workshop Visualisation et Extraction de Connaissances EGC05*, Paris, France, pp. 37–47.

Bonnel, N., A. Cotarmanac'h, et A. Morin (2005b). Visualisation 3d des résultats de recherche : quel avenir ? In *Proceedings of Hypermedias Hypertexts, Products, Tools and Methods, H2PTM'05*, pp. 325–339. Hermes Science Publications.

Brandes, U. et T. Willhalm (2002). Visualization of bibliographic networks with a reshaped landscape metaphor. In *Proceedings of the symposium on Data Visualisation 2002*, Aire-la-Ville, Switzerland, Switzerland, pp. 159–164. Eurographics Association.

Card, S. K., J. D. Mackinlay, et B. Shneiderman (1999). *Readings in information visualization : using vision to think*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

Chiu, P., A. Girgensohn, S. Lertsithichai, W. Polak, et F. Shipman (2005). Mediametro : browsing multimedia document collections with a 3d city metaphor. In *Proceedings of the 13th annual ACM international conference on Multimedia :MULTIMEDIA '05*, New York, NY, USA, pp. 213–214. ACM Press.

Fabrikant, S. I., D. R. Montello, et D. M. Mark (2010). The natural landscape metaphor in information visualization : The role of commonsense geomorphology. *JASIST 61*(2), 253–270.

Fabrikant, S. I. et A. Skupin (2005). Cognitively Plausible Information Visualization. Chapter 35, pp. 667–690. Elsevier.

Ganter, B. et R. Wille (1999). *Formal Concept Analysis.* Springer, Heidelberg.

Garcá-Silva, A., M. Szomszor, H. Alani, et O. Corcho (2009). Preliminary results in tag disambiguation using dbpedia. In *Knowledge Capture (K-Cap 2009)-Workshop on Collective Knowledge Capturing and Representation-CKCaR*.

Graham, H., H. Y. Yang, et R. Berrigan (2004). A solar system metaphor for 3D visualisation of object oriented software metrics. In *Proceedings of the 2004 Australasian symposium on Information Visualisation APVis '04*, Darlinghurst, Australia, Australia, pp. 53–59. Australian Computer Society, Inc.

Hearst, M. A. et C. Karadi (1997). Cat-a-cone : an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '97, Philadelphia, Pennsylvania, US, pp. 246–255. ACM.

Hotho, A., R. Jäschke, C. Schmitz, et G. Stumme (2006). Information retrieval in folksonomies : Search and ranking. In Y. Sure et J. Domingue (Eds.), *Proceedings of The Semantic Web : Research and Applications. LNCS*, Volume 4011, pp. 411–426. Springer, Heidelberg.

Jäschke, R., A. Hotho, C. Schmitz, B. Ganter, et G. Stumme (2006). TRIAS - an algorithm for mining iceberg tri-lattices. In *Procedings of the 6th IEEE International Conference on Data Mining, (ICDM 2006), IEEE Computer Society, Hong Kong*, pp. 907–911.

Panas, T., R. Berrigan, et J. C. Grundy (2003). A 3d metaphor for software production visualization. In *Proceedings International Conference on Information Visualization :IV'03*, pp. 314–319.

Santos, C. R. D., P. Gros, P. Abel, D. Loisel, N. Trichaud, et J. P. Paris (2000). Mapping information onto 3d virtual worlds. In *Proceedings of IEEE International Conference on Information Visualization*, London, UK, pp. 379–386.

Simoff, S. J. (2001). Form-semantics-function - a framework for designing visualization models for visual data mining. In D. A. Keim et S. Eick (Eds.), *Proceedings of Workshop on Visual Data Mining KDD-2001*, San Francisco, California, USA, pp. 29–36.

Spence, R. (2001). *Information visualization*. ACM Press books. Harlow, England : A. Wesley.

Trabelsi, C., N. Jelassi, et S. Ben Yahia (2011). Auto-complétion de requêtes par une base générique de règles d'association triadiques. In *Proceedings of the 8th French Information Retrieval Conference, CORIA'11*, Avignon, France, pp. 9–24. Éditions Universitaires d'Avignon.

Wettel, R. (2008). Scripting visualization with codecity. In *Proceedings of FAMOOSr 2008 (2nd Workshop on FAMIX and Moose in Reengineering)*, Antwerp, Belgium.

Wettel, R., M. Lanza, et R. Robbes (2011). Software systems as cities : a controlled experiment. In *Proceedings of the 33rd International Conference on Software Engineering*, pp. 551–560.

Yin, D., Z. Xue, L. Hong, et B. Davison (2010). A probabilistic model for personalized tag prediction. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, SIGKDD 2010*, Washington, USA, pp. 959–968. ACM Press.

Summary

Recently, social bookmarking systems have received surging an increasing attention in both academic and industrial communities. The main thrust of these Web 2.0 systems is their easy use that relies on simple intuitive process, allowing their users to label diverse resources with freely chosen keywords aka *tags*. The obtained collections are known under the nickname *Folksonomy*. In this paper, we introduce the FOLKOVIZ tool dedicated to the socio-semantic visualisation of the folksonomic data. The tool proceeds

in two phases: (i)Extraction and desambiguisation of triadic concepts; (ii) 3D visualization using the City metaphore integrating the social dimension thanks to an exploration of SIOC and FOAF ontologies.