Clustering par optimisation de la modularité pour trajectoires d'objets mobiles

Mohamed K. El Mahrsi, Télécom ParisTech, Département INFRES
46, rue Barrault 75634 Paris CEDEX 13, France
Fabrice Rossi, Équipe SAMM EA 4543, Université Paris I Panthéon-Sorbonne
90, rue de Tolbiac 75634 Paris CEDEX 13, France

Résumé : *L'analyse et la fouille des traces de mobilité produites par divers objets mobiles est un sujet de recherche qui sollicite un grand intérêt depuis quelques années. Dans le présent article, nous présentons une approche de classification (ou clustering) adaptée aux données de véhicules se déplaçant sous contraintes d'un réseau routier. Une mesure de similarité est proposée pour comparer les trajectoires étudiées entre elles en tenant compte des contraintes de déplacement imposées par le réseau. Cette mesure est exploitée pour construire un graphe traduisant les différentes relations de similarité entretenues par les trajectoires entre elles. Nous partitionnons ce graphe à l'aide d'un algorithme utilisant la notion de modularité comme critère de qualité afin de découvrir des communautés (ou clusters) de trajectoires qui sont fortement liées et qui présentent un comportement commun. Nous avons implémenté et testé l'approche proposée sur plusieurs jeux de données synthétiques à travers lesquels nous montrons son fonctionnement.*

Abstract : *Analysis and data mining of moving objects trajectories have gained a considerable amount of interest in the last few years. In this article, we present a clustering approach tailored for trajectories of vehicles moving on a road network. First, we introduce a similarity measure that makes it possible to compare such trajectories while taking into account the constraints of the underlying network. Then, this measure is used to construct a graph that models the interactions among the trajectories w.r.t. their similarity. A community detection algorithm based on modularity optimization is applied to the graph in order to discover groups of trajectories that behaved similarly and that moved along the same portions of the road network. We implemented the proposed approach and tested it on multiple synthetic datasets in order to show its feasibility and its efficiency.*

Mots-clés : trajectoires, objets mobiles, similarité, classification.
Keywords : trajectories, moving objects, similarity, clustering.

1 Introduction

Les grands progrès, réalisés ces dernières années, en technologies de géolocalisation et de télécommunications ont contribué à la disponibilité sur le marché grand public d'appareils capables de retrouver leur position géographique, de la sauvegarder et de la partager. Il devient ainsi envisageable de construire des bases de données pour stocker et traiter des données en provenance de divers types d'objets mobiles (oiseaux migrateurs, véhicules équipés de GPS, piétons avec leurs smartphones et PDA, etc.). La disponibilité de ce nouveau type de données et leur nature spatiotemporelle si particulière (décrivant une position spatiale évoluant constamment au fil du temps) ont soulevé de nouveaux problèmes et ont motivé la recherche sur la gestion, la fouille et l'analyse des bases de données d'objets mobiles [11].

La gestion et l'analyse du trafic routier figure parmi les domaines d'application dans lesquels les données d'objets mobiles se montrent très intéressantes à étudier. En effet, la congestion du trafic est devenue, de nos jours, un problème qui affecte quotidiennement l'activité humaine résultant constamment en des délais dans les transports et en de sérieux problèmes environnementaux. La surveillance et l'analyse de l'état du réseau routier est habituellement effectuée en déployant des capteurs dédiés qui mesurent les taux d'exploitation des portions routières sur lesquelles ils sont situés. Cependant, les coûts élevés du déploiement et de la maintenance de ces capteurs limitent leur exploitation au réseau routier principal (c.à-d. les autoroutes et les grandes routes seulement) et l'état du réseau dans sa globalité ne peut pas être étudié. Une approche alternative peut consister à récolter directement les traces GPS provenant de divers véhicules empruntant le réseau routier. Ces données peuvent être

alignées avec le plan du réseau avec un algorithme de map matching [4] et l'état du réseau à un instant donné peut être déduit. Elles peuvent également être stockées en vue d'effectuer des tâches de fouille plus poussées.

Dans cet article, nous nous intéressons à la problématique du clustering de trajectoires de véhicules se déplaçant sur un réseau routier. Le clustering vise à partitionner un jeu de données en plusieurs groupes (clusters) regroupant des individus à comportement similaire. Dans notre contexte, ceci se traduit par des groupes de trajectoires qui se sont déplacées de façon similaire. Cette tâche est effectuée en post traitement sur un historique regroupant un nombre important de trajectoires. Elle permet de découvrir des patterns de mouvement de groupe qui sont invisibles à l'échelle de trajectoires individuelles ou en analysant seulement les taux d'occupation des segments routiers. Ces patterns sont très utiles dans diverses applications parmi lesquelles nous citons :

— La planification et le réaménagement de l'infrastructure routière : la découverte de clusters de trajectoires permet de mieux comprendre la dynamique d'exploitation du réseau routier et permet d'évaluer l'adéquation de ce dernier à l'usage qui en est fait. Les choix de planification et de construction de nouvelles routes peuvent également bénéficier de cette connaissance.
— Le covoiturage : les clusters de trajectoires font apparaitre des opportunités de covoiturage qui, si elles sont saisies, permettent de réduire les coûts des déplacements et leur impact sur l'environnement.

Plusieurs travaux de recherche se sont intéressés à la problématique de clustering de trajectoires d'objets mobiles. Toutefois, la plupart supposent un mouvement non contraint et ne tiennent donc pas compte des contraintes qui peuvent régir les déplacements des objets mobiles du fait de la présence d'un réseau sous-jacent. Or, la topologie de ce réseau joue un rôle clé dans la caractérisation des trajectoires et leur regroupement sous des profils similaires.

En résumé, les contributions de cet article sont les suivantes :

— La définition d'une nouvelle mesure de similarité qui permet de comparer des trajectoires entre elles en exploitant le réseau sous-jacent au déplacement ;
— La proposition d'une modélisation des relations entre les trajectoires sous forme d'un graphe de similarité ;
— L'utilisation d'une méthode de clustering de trajectoires basée sur la détection de communautés dans les graphes. Cette méthode repose sur un critère de qualité bien défini et qui a fait ses preuves dans le cadre du clustering de graphes larges. Elle fournit une hiérarchie de clusters qui permet une analyse sur différents niveaux de détail.
— La validation de l'approche proposée sur plusieurs jeux de données synthétiques, générés en utilisant la carte d'un réseau routier réel.

Le reste de l'article est organisé comme suit : la section 2 présente les travaux existants autour du clustering de trajectoires. Dans la section 3 nous formalisons notre problématique et nous présentons le modèle de données adopté. La section 4 détaille la mesure de similarité que nous employons pour comparer les trajectoires entre elles ainsi que notre approche de clustering. Notre étude expérimentale est présentée dans la section 5. Nous concluons cet article dans la section 6.

2 État de l'art

Les travaux liés à notre problématique peuvent être classés en deux grandes catégories : i. l'étude de similarité entre trajectoires ; et ii. la conception d'algorithmes de clustering adaptés aux trajectoires.

2.1 Similarité entre trajectoires

Plusieurs distances sont proposées pour comparer des trajectoires en mouvement libre entre elles. La distance Euclidienne peut être utilisée pour comparer deux trajectoires mais impose que celles-ci soient de longueur égale. La distance DTW (*Dynamic Time Warping*) [3] permet de comparer des trajectoires de longueurs différentes mais n'est pas robuste face à la présence de bruit dans les données. Vlachos et al. [24] exploitent le principe de LCSS (*Longuest Common Subsequence*) pour proposer

un ensemble de distances et de mesures de similarité qui sont robustes face à la présence de données aberrantes dans les trajectoires analysées. Les distances ERP (*Edit distance with Real Penalty*) et EDR (*Edit Distance on Real sequence*) [8] sont deux adaptations de la distance d'édition au cas des trajectoires. EDR est robuste en présence du bruit tandis que ERP ne l'est pas. Lin et Su [19] proposent la distance OWD (*One-Way Distance*) pour comparer des trajectoires en se basant seulement sur leur forme spatiale. Toutes ces distances sont inadaptées au cas de trajectoires contraintes par un réseau puisqu'elles se basent sur la distance euclidienne et ignorent les restrictions topologiques imposées par le réseau.

Dans [14], Hwang et al. introduisent l'une des premières mesures de similarité adaptées au mouvement contraint. Les trajectoires sont filtrées en appliquant une similarité spatiale sur le réseau (passage par des points d'intérêt prédéfinis) puis le résultat est raffiné en appliquant un critère temporel. D'autres distances spatiotemporelles sont également présentées par Tiakas et al. [23] et Chang et al. [7].

2.2 Algorithmes de clustering de trajectoires

La problématique de clustering a été étudiée de façon exhaustive dans le cas de données statiques. Plusieurs approches ont été proposées dans la littérature, que ce soit des méthodes par partitionnement (ex. *k*-means), des méthodes hiérarchiques (ex. BIRCH [25]) ou encore des méthodes basées sur la densité (ex. DBSCAN [9] et OPTICS [1]).

Plusieurs approches de clustering sont proposées pour le cas de trajectoires d'objets mobiles. Ces approches diffèrent, généralement, selon le choix de la représentation des données (géométrique ou symbolique), les dimensions prises en compte (spatiale, temporelle ou les deux) et la granularité du clustering (trajectoires entières, portions de trajectoires, etc.). Nanni et Pedreschi [20] adaptent l'algorithme OPTICS au cas des trajectoires : la variante T-OPTICS regroupe des trajectoires entières alors que la variante TF-OPTICS regroupe des sous-trajectoires. Dans [2], les auteurs introduisent la notion de *flock patterns* qui consistent en des groupes d'objets mobiles se déplaçant ensemble dans un disque de rayon donné. La notion de *convoy pattern* est introduite dans [15] et utilise l'algorithme DBSCAN pour regrouper des objets mobiles sur plusieurs instants temporels consécutifs. Lee et al. [18] proposent l'algorithme TRACLUS : les trajectoires sont d'abord simplifiées avec un algorithme de type MDL (*Minimal Description Length*) puis des sous-trajectoires sont regroupées ensemble avec une adaptation de l'algorithme DBSCAN. Tous ces algorithmes font l'hypothèse d'un mouvement libre et utilisent des distances euclidiennes pour les comparaisons.

Dans [16], les auteurs proposent un algorithme pour découvrir des chemins denses résultants des déplacements sur un réseau routier en exploitant le principe de densité introduit par DBSCAN. Roh et Hwang [21] proposent l'algorithme NNCluster qui exploite les calculs de plus court chemin dans le réseau routier pour évaluer la distance entre les trajectoires et les partitionner.

Une nouvelle tendance dans le clustering de trajectoires a vu le jour très récemment. Elle consiste à utiliser des techniques issues de l'analyse et du clustering de graphes en les adaptant au contexte des trajectoires. Dans [5], les auteurs construisent un graphe modélisant le nombre de "rencontres" entre les trajectoires et calculent différentes statistiques là dessus. Une autre approche de ce genre est proposée par Guo et al. [12] où un graphe est construit avec comme noeuds les points constituant les trajectoires et comme arcs le nombre de trajectoires ayant passé à la fois par les deux points reliés. Ce graphe est partitionné pour découvrir des zones d'intérêt regroupant des points souvent visités conjointement. Bien que les auteurs de ces travaux citent des applications dans le cas contraint, leurs méthodes exploitent des calculs de distance euclidienne entre points de trajectoires non contraints.

3 Problématique et modèle de données

Un réseau routier peut être modélisé sous forme d'un graphe orienté $G = (N, S)$. L'ensemble des nœuds N désigne les d'intersections routières et l'ensemble des arcs S correspond aux segments routiers reliant ces intersections entre elles. L'orientation d'un segment donné $s = (n_i, n_j)$ reliant deux

noeuds n_i et n_j indique que le sens de déplacement autorisé sur ce segment routier est du nœud n_i au nœud n_j et non pas inversement.

Une trajectoire T se déplaçant sur ce réseau peut être modélisée avec une représentation symbolique comme étant la suite ordonnée des segments visités [1] :

$$T =< id, \{s_1, ..., s_i, ..., s_n\} >$$

Avec *id* l'identifiant de la trajectoire T et n le nombre de segments la constituant.

Étant donné un ensemble de trajectoires $\mathscr{T}$ qui se sont déplacées sur un réseau routier modélisé par un graphe G, le problème de clustering de trajectoires contraintes par un réseau consiste à découvrir des sous-ensembles de trajectoires (appelés clusters) $\mathscr{C} = \{C_1, C_2, ..., C_m\}$ ayant exhibé un comportement similaire. Les trajectoires appartenant à un même cluster doivent se ressembler le plus que possible et les trajectoires appartenant à des clusters différents doivent être aussi différentes que possible. Notre notion de similarité et la façon de découvrir ces clusters sont expliquées dans la section suivante.

4 Clustering basé sur la maximisation de la modularité pour les trajectoires contraintes

Nous présentons une nouvelle mesure pour calculer la similarité entre trajectoires en tenant compte des contraintes du réseau routier. Puis, nous expliquons comment celle-ci est exploitée pour effectuer la classification des trajectoires.

4.1 Mesure de similarité

Dans notre approche, nous considérons les trajectoires selon un paradigme *bag-of-segments* (inspiré du modèle *bag-of-words* [13]) : les trajectoires sont comparées segment par segment de façon individuelle (quand un segment est analysé, la présence d'autres segments ainsi que l'ordre du segment dans la trajectoire n'ont aucune influence). Ce choix découle de deux raisons :

— Dans une optique d'analyse du trafic routier, les situations de congestion apparaissent d'abord sur des segments routiers individuels et isolés puis s'étendent avec le temps aux segments adjacents. Suivre une logique similaire où l'on compare les trajectoires segment par segment est suffisante pour tenir compte de ces situations ;

— Bien que l'ordre est négligé lors de la comparaison individuelle des segments, le fait que le mouvement des objets mobiles est effectué sur un graphe orienté permet de bien tenir compte du sens de parcours et de bien séparer des trajectoires qui ont circulé sur les mêmes routes mais dans des sens opposés.

Nous adaptons la notion de TF-IDF (*Term Frequency - Inverse Document Frequency*), qui est largement utilisée en informatique documentaire, au cas des trajectoires afin de pondérer les différents segments routiers selon leur pertinence par rapport à l'ensemble de trajectoires analysées. Nous définissons la fréquence spatiale (*spatial segment frequency*) qui mesure l'importance d'un segment s par rapport à une trajectoire T comme étant le rapport entre la longueur de ce segment ($\text{length}(s)$, multipliée par son nombre d'occurrences dans la trajectoire $n_{\text{occ}}(s)$) et la longueur totale de T :

$$\text{ssf}_{s,T} = \frac{n_{\text{occ}}(s) \cdot \text{length}(s)}{\sum_{s' \in T} \text{length}(s')}$$

La fréquence de trajectoire inversée (*Inverse Trajectory Frequency*) d'un segment s mesure l'importance de ce segment dans tout l'ensemble de trajectoires analysées $\mathscr{T}$:

$$\text{itf}_s = \log \frac{|\mathscr{T}|}{|\{T_i : s \in T_i\}|}$$

1. La dimension temporelle peut être intégrée, si nécessaire, en étiquetant chaque segment s_i par la date d'entrée t_i de la trajectoire sur ce segment.

Avec $|\mathscr{T}|$ le nombre total de trajectoires dans $\mathscr{T}$ et $|\{T_i : s \in T_i\}|$ le nombre de trajectoires qui contiennent le segment s.

Ainsi, le poids du segment s dans une trajectoire T s'obtient par :

$$\omega_{s,T} = \mathrm{ssf}_{s,T} \cdot \mathrm{itf}_s$$

L'attribution de ces poids permet de pénaliser les segments routiers qui sont communs et très fréquents sur tout l'ensemble de trajectoires ainsi que d'accorder plus d'importances aux segments qui sont peu fréquents et qui seront, par conséquent, plus discriminants pour caractériser les trajectoires et les partitionner.

Finalement, pour comparer deux trajectoires T_i et T_j entre elles, nous calculons leur similarité cosinus :

$$\mathrm{Similarity}(T_i, T_j) = \frac{\sum_{s \in S} \omega_{s,T_i} \cdot \omega_{s,T_j}}{\sqrt{\sum_{s \in S} \omega_{s,T_i}^2} \cdot \sqrt{\sum_{s \in S} \omega_{s,T_j}^2}}$$

Cette approche ne considère que la dimension spatiale dans l'analyse des trajectoires. En effet, dans le contexte de gestion du trafic routier, la prise de décision peut avoir un impact sur le routage du trafic sur certaines portions du réseau (ex. interdire la circulation dans un sens donné) ce qui affecte toutes les trajectoires utilisant cette portion de route quel que soit leur temps de passage.

4.2 Construction du graphe de similarité entre trajectoires

Nous modélisons les relations de similarité entre les trajectoires par un graphe de similarité pondéré et non-orienté $G_S = (\mathscr{T}, E, W)$. Chaque trajectoire dans $\mathscr{T}$ correspond à un sommet dans G_S. Une arête $e \in E$ existe entre deux trajectoires T_i et T_j si et seulement si $\mathrm{Similarity}(T_i, T_j) > 0$ (Autrement dit, si les deux trajectoires ont au moins un segment routier en commun). Dans ce cas, la similarité entre les deux trajectoires est attribuée en tant que poids $\omega_e \in W$ à l'arête en question.

Cette représentation en graphe (qui est largement utilisée dans d'autres domaines tels que les réseaux sociaux, la biologie, etc.) est un moyen naturel de décrire les interactions entre les trajectoires. Par ailleurs, elle permet de mettre l'accent sur le fait que des trajectoires totalement différentes ne doivent, a priori, pas être regroupées ensemble du fait de l'absence d'une arête qui les relie directement.

4.3 Clustering du graphe de similarité

Le clustering est une tâche de classification qui est effectuée sur un volume important de données. Ainsi, le graphe de similarité construit comme indiqué dans la section précédente tend à avoir un très grand nombre de sommets. En plus, du fait qu'il suffit que deux trajectoires partagent au moins un segment pour qu'une arête les reliant existe dans le graphe, les sommets tendent à avoir un fort degré. La détection de communautés par maximisation de la modularité est l'un des moyens les plus efficaces et les plus pertinents pour effectuer le clustering de ce type de graphes [10]. Pour un graphe $G = (V, E, W)$, de sommets $V = \{v_1, v_2, ..., v_n\}$, d'arêtes E pondérées par les poids $(\omega_{ij})_{ij}$ (où $\omega_{ij} \geq 0$ et $\omega_{ij} = \omega_{ji}$) et étant donné une partition des sommets en K clusters (appelés dans ce contexte "communautés") $C_1, C_2, ..., C_K$. La modularité de cette partition est exprimée par :

$$\mathscr{Q} = \frac{1}{2m} \sum_{k=1}^{K} \sum_{i,j \in C_k} \left(\omega_{ij} - \frac{d_i d_j}{2m} \right)$$

Avec $d_i = \sum_{j \neq i} \omega_{ij}$ et $m = \frac{1}{2} \sum_i d_i$.

La modularité mesure la qualité de la classification en inspectant la disposition des arêtes au sein des communautés de sommets. Une modularité élevée indique que les arêtes à l'intérieur des communautés sont plus nombreuses (ou possèdent des poids plus élevés) que dans le cas d'un graphe où les arêtes sont distribuées de façon aléatoire.

Pour effectuer le clustering de notre graphe de similarité G_S nous utilisons l'algorithme décrit dans [22]. L'algorithme prend en entrée le graphe G_S et se charge de trouver une partition optimale (c.à-d. une partition qui maximise la mesure de modularité). Cette partition est validée en mesurant sa "significativité" (en comparant sa modularité à la modularité obtenue sur les partitions optimales de graphes aléatoires ayant une structure similaire à celle du graphe G_S). Si la partition est valide, l'algorithme est repris de façon récursive sur chacune des communautés (autrement dit, le sous-graphe formé par les sommets de la communauté et leurs arêtes internes est isolé et le clustering est effectué sur celui-ci). La récursivité s'arrête lorsqu'aucune des sous partitions ne peut être partitionnée davantage. Le résultat est une hiérarchie de clusters imbriqués qui peuvent être explorés niveau par niveau ou de façon gloutonne (en éclatant à chaque fois le cluster qui cause la plus faible perte de modularité).

5 Étude expérimentale

Nous présentons ici notre étude expérimentale où nous comparons notre algorithme au clustering hiérarchique ascendant classique.

5.1 Données utilisées

Des jeux de données de trajectoires peuvent être construits à partir de plusieurs sources réelles (telles que des flottes de véhicules équipés de GPS). Cependant, la majorité de ces données sont propriétaires et ne sont pas disponibles pour un usage autre que celui pour lequel elles étaient prévues initialement. Une alternative consiste, donc, à utiliser des données de synthèse qui sont générées en simulant les déplacements des objets mobiles sur un réseau routier réel. Dans notre étude expérimentale, nous utilisons deux types de jeux de données synthétiques.

Nous utilisons un jeu de données constitué de 10000 trajectoires générées à l'aide du générateur de Brinkhoff [6]. Ce dernier permet de générer des trajectoires se déplaçant sur un réseau routier en tenant compte de plusieurs facteurs (ex. vitesse limite et occupation des segments routiers, types des véhicules, etc.). Le but de ce premier scénario est d'observer le comportement des algorithmes testés sur un grand jeu de données où aucune classification n'est connue a priori.

En plus du jeu de données de Brinkhoff, nous utilisons plusieurs jeux de données de plus petite taille que nous avons générés de façon à faire apparaitre des clusters. Chaque cluster est généré en choisissant (sur le réseau routier utilisé) une zone de départ et une zone d'arrivée. Puis, des trajectoires sont générées en choisissant un nœud de départ (resp. d'arrivée) parmi les nœuds inclus dans la zone de départ (resp. d'arrivée) et en se déplaçant en suivant le plus court chemin entre ces deux nœuds. Le but de ce second scénario est de voir si les algorithmes testés sont capables de bien séparer les clusters et retrouver les classes originales des trajectoires.

Tous les jeux de données susmentionnés sont générés en utilisant le réseau routier de la ville d'Oldenburg dont le graphe est constitué de 6105 nœuds et 7035 segments routiers (non orientés).

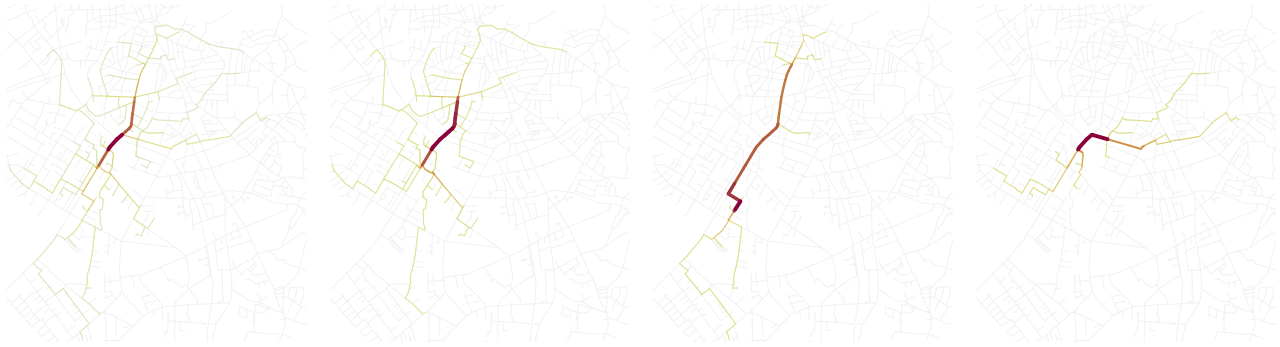
5.2 Résultats sur les données du générateur de Brinkhoff

Le clustering du jeu de données générées par le générateur de Brinkhoff avec notre approche révèle l'existence de six niveaux de hiérarchie avec seulement neuf clusters au plus haut niveau et 648 clusters au niveau le plus bas (cf. tableau 1). La figure 1 illustre le jeu de données original (Figure 1(a)) ainsi que quelques clusters du niveau le plus haut (La couleur des segments routiers correspond à leur taux d'usage au sein du cluster. Plus un segment est visité plus sa couleur est foncée.).

Le fait de disposer de plusieurs niveaux hiérarchiques à des résolutions plus ou moins grossières peut être très utile pour analyser des jeux de données de trajectoires de grande taille. L'utilisateur peut commencer avec le niveau de clustering le plus haut pour comprendre les tendances générales du mouvement puis accéder, par zooms successifs, à des détails de plus en plus fins dans les clusters qui l'intéressent (Figure 2).

(a) (b) (c) (d)

FIG. 1 – *Jeu de données original (a) et quelques clusters découverts par application du clustering basé sur la modularité.*



(a) (b) (c) (d)

FIG. 2 – *Un cluster (a) contenant 38 trajectoires ainsi que ses trois sous clusters (b) (c) et (d) composés respectivement de 22, 8 et 8 trajectoires.*

Pour chacun des niveaux trouvés, nous comparons notre approche de clustering hiérarchique basé sur la modularité au clustering hiérarchique ascendant classique (CHA) avec saut minimal (*simple linkage*) et saut maximal (*complete linkage*). Pour ce faire, nous arrêtons la procédure de CHA au même nombre de clusters et nous comparons les *taux de chevauchement* intra-clusters atteints par les deux méthodes. Le taux de chevauchement intra-clusters permet de mesurer la ressemblance entre les trajectoires regroupées ensembles. Plus il est élevé plus les trajectoires dans les clusters sont similaires. Nous définissons le taux de chevauchement intra-clusters pour un ensemble $\mathscr{C}$ de clusters de trajectoires selon la formule :

$$\text{taux de chevauchement} = \sum_{C \in \mathscr{C}} \frac{1}{|C|} \sum_{T_i, T_j \in C} \frac{\sum_{s \in T_i, s \in T_j} \text{length}(s)}{\sum_{s \in T_i} \text{length}(s)}$$

Le tableau 1 résume les performances réalisées par les deux approches. Comme attendu, plus on descend dans les niveaux de la hiérarchie plus les trajectoires rassemblées dans un même cluster sont similaires. Notre approche est celle qui a obtenu les meilleurs taux de chevauchement sur tous les niveaux, suggérant ainsi que c'est celle qui a le mieux classé les trajectoires. Notons que le clustering hiérarchique avec simple linkage n'a pas réussi à effectuer le clustering du jeu de données et a résulté en des clusters très disproportionnés (ce qui s'est traduit par ses taux de chevauchement si bas par rapport à ceux obtenus par notre approche et le clustering hiérarchique avec complete linkage).

5.3 Résultats sur les données avec clusters prédéterminés

Comme indiqué dans la section 5.1, nous utilisons également des jeux de données que nous avons créés pour faire apparaitre des clusters générés aléatoirement. Ces clusters peuvent interagir de façon naturelle entre eux. Par exemple, deux clusters peuvent avoir une zone d'arrivée (ou de départ) commune et peuvent donc partager une portion routière commune. Le but de ce second scénario et de

Niveau de hiérarchie	Nombre de clusters	Taux de chevauchement		
		CHA (S)	CHA (C)	Modularité
1	9	110.65	122.85	**608.19**
2	45	148.92	349.38	**1767.56**
3	159	270.12	1877.42	**3121.02**
4	419	568.87	3681.85	**4264.11**
5	621	822.10	4419.30	**4779.68**
6	648	881.02	4419.30	**4823.44**

TAB 1 – *Taux de chevauchement intra-clusters réalisés par le clustering hiérarchique ascendant (CHA) classique (en S : simple linkage et C : complete linkage) et le clustering hiérarchique par optimisation de la modularité.*

voir si les algorithmes de clustering peuvent bien séparer les trajectoires et retrouver la classification établie a priori.

Nous testons le clustering hiérarchique classique et notre clustering hiérarchique basé sur la modularité sur dix jeux de données contenant dix clusters chacun (chaque cluster contient entre 20 et 50 trajectoires). Pour évaluer les performances des algorithmes testés, nous mesurons la pureté et l'entropie [26] des clusters générés. La pureté d'un ensemble de clusters $\mathscr{C}$ est donnée par :

$$\text{purity} = \frac{1}{|\mathscr{T}|} \cdot \sum_{C \in \mathscr{C}} \max_i n_C^i$$

Avec $|\mathscr{T}|$ le nombre de trajectoires dans le jeu de données $\mathscr{T}$ et $\max_i n_C^i$ le nombre d'individus appartenant à la classe (originale) majoritaire représentée par le cluster.

L'entropie est donnée par :

$$\text{entropy} = \sum_{C \in \mathscr{C}} \frac{|C|}{|\mathscr{T}|} \cdot E(C)$$

Où $E(C)$ est l'entropie du cluster C, exprimée par :

$$E(C) = \frac{1}{\log q} \sum_{i=1}^{q} \frac{n_C^i}{|C|} \log \frac{n_C^i}{|C|}$$

q étant le nombre de classes originales dans le jeu de données et n_C^i le nombre d'individus de la classe i qui ont été regroupés dans le cluster C.

Jeu de données	Nombre de trajectoires	Clusters trouvés	Pureté			Entropie		
			CHA (S)	CHA (C)	Mod.	CHA (S)	CHA (C)	Mod.
1	341	8	0.76	0.81	**0.84**	0.17	0.16	**0.12**
2	387	13	0.69	0.74	**0.76**	0.22	0.20	**0.18**
3	349	9	0.63	**0.80**	**0.80**	0.29	0.14	**0.13**
4	328	11	0.76	0.79	**0.85**	0.19	0.15	**0.11**
5	390	14	0.76	0.58	**0.92**	0.19	0.32	**0.06**
6	309	9	0.72	0.73	**0.90**	0.21	0.20	**0.07**
7	312	9	0.70	0.70	**0.73**	0.25	0.23	**0.20**
8	332	12	0.77	0.82	**0.94**	0.19	0.13	**0.05**
9	323	9	0.76	0.83	**0.93**	0.19	0.14	**0.05**
10	355	14	**0.94**	0.87	0.87	**0.04**	0.09	0.09

TAB 2 – *Performances réalisées sur les jeux de données avec clusters connus a priori.*

Le tableau 2 indique pour chaque jeu de données le nombre de trajectoires qui le composent ainsi que le nombre de clusters trouvés par notre approche quand elle y est appliquée. Le nombre de clusters est utilisé pour comparer les trois approches étudiées à pied d'égalité (c.à-d. nous arrêtons le clustering

hiérarchique classique avec ce même nombre pour calculer les indicateurs de performances). Notre méthode a réalisé le meilleur clustering sur la grande majorité des jeux de données analysés. Les résultats montrent également que le nombre de clusters trouvés (et leurs contenus) coïncide rarement avec le nombre de clusters générés. La méthode peut décider que certains clusters se ressemblent trop (par exemple, les clusters convergeant vers une destination communes) et les fusionne en un seul cluster qui n'est pas raffiné par la suite. Elle peut également décider qu'un cluster original doit être raffiné et partitionné davantage en plusieurs sous-clusters.

6 Conclusion et perspectives

Dans cet article, nous avons présenté une nouvelle approche de classification adaptée aux trajectoires d'objets mobiles se déplaçant en suivant un réseau routier. L'idée clé de l'approche proposée consiste à modéliser les relations de similarité entre les différentes trajectoires analysées sous forme d'un graphe. Ce graphe est ensuite partitionné en utilisant un critère de qualité qui a fait ses preuves afin de faire apparaitre des groupes de trajectoires qui se sont déplacées de façon similaire.

Notre approche se distingue par le fait qu'elle ne nécessite pas des paramètres de configuration contrairement à la majorité des méthodes existantes, notamment les méthodes basées sur la densité dont les résultats varient de façon significative en fonction du choix des valeurs des seuils *minPts* et ε. En plus, au lieu d'offrir un clustering plat sur un seul niveau, notre approche permet d'obtenir une hiérarchie de clusters imbriqués qui s'étalent sur plusieurs niveaux. Ceci est un atout majeur lors de l'analyse de grands jeux de données de trajectoires : l'utilisateur peut commencer avec un nombre limité de clusters avec une simplification grossière pour comprendre les tendances générales de mouvement puis accéder, par zooms successifs, à plus de détails dans les clusters qui l'intéressent. L'une des limitations de notre approche concerne la gestion du bruit (c.à-d. les trajectoires qui ne sont pas suffisamment pertinentes pour appartenir à un quelconque cluster). Notre étape de clustering par la maximisation de la modularité ne permet pas de détecter ce genre de trajectoires et de les éliminer. Par conséquent, la présence de trajectoires aberrantes peut dégrader la qualité des clusters découverts.

Il serait intéressant de coupler la méthode proposée avec un outil de visualisation qui permet de faciliter l'exploitation des résultats et la navigation dans les clusters de façon interactive. Une deuxième perspective consiste à essayer de construire d'autres types de graphes de similarité à partir des trajectoires analysées et d'en faire l'étude. Par exemple, nous pouvons construire un graphe de similarité entre les différents segments routiers en fonction de leur utilisation mutuelle et en effectuer le clustering pour faire apparaitre des groupes de segments qui sont souvent visités ensemble. Enfin, la dégradation des clusters en cas de présence de trajectoires aberrantes peut être corrigée avec une étape de post traitement qui consiste à inspecter les clusters individuels et en supprimer les trajectoires qui ne sont pas suffisamment Aussi, l'approche présentée ici repose sur un modèle et une similarité purement spatiaux, ce choix étant justifié par le fait que, dans un contexte d'analyse et de réaménagement des infrastructures routières, la prise de décision touche normalement toutes les trajectoires passant par une portion routière donnée quels que soient leurs temps de passage par cette portion. Cependant, dans d'autres contextes (ex. recherche d'opportunités de covoiturage) la prise en compte du temps serait essentielle pour corréler correctement les trajectoires. Une piste possible pour arriver à cette fin serait de suivre une approche similaire à celle présentée dans [17] et qui consiste à augmenter le modèle de données avec la dimension temporelle (comme indiqué brièvement dans la section 3) et à diviser l'intervalle du temps couvert par le jeu de données en plusieurs sous-intervalles d'intérêt sur lesquels le clustering spatial est effectué séparément.

7 Remerciements

Les auteurs tiennent à remercier les relecteurs dont les remarques ont permis d'améliorer considérablement la qualité de cet article et de clarifier davantage certains aspects de ce travail.

Références

[1] M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics : ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2) :49–60, 1999.

[2] M. Benkert, J. Gudmundsson, F. Hübner, and T. Wolle. Reporting flock patterns. In *ESA'06 : Proceedings of the 14th conference on Annual European Symposium*, pages 660–671, London, UK, 2006. Springer-Verlag.

[3] D. J. Berndt and J. Clifford. Finding patterns in time series : a dynamic programming approach. pages 229–248, 1996.

[4] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, pages 853–864. VLDB Endowment, 2005.

[5] I. R. Brilhante, J. A. F. de Macedo, C. Renso, and M. A. Casanova. Trajectory data analysis using complex networks. In *Proceedings of the 15th Symposium on International Database Engineering & Applications*, IDEAS '11, pages 17–25, New York, NY, USA, 2011. ACM.

[6] T. Brinkhoff. A framework for generating network-based moving objects. *Geoinformatica*, 6 :153–180, June 2002.

[7] J.-W. Chang, R. Bista, Y.-C. Kim, and Y.-K. Kim. Spatio-temporal similarity measure algorithm for moving objects on spatial networks. In *Proceedings of the 2007 international conference on Computational science and its applications - Volume Part III*, ICCSA'07, pages 1165–1178, Berlin, Heidelberg, 2007. Springer-Verlag.

[8] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *SIGMOD '05 : Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502, New York, NY, USA, 2005. ACM.

[9] M. Ester, H.-p. Kriegel, J. S, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231, 1996.

[10] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5) :75–174, 2010.

[11] F. Giannotti and D. Pedreschi, editors. *Mobility, Data Mining and Privacy - Geographic Knowledge Discovery*. Springer, 2008.

[12] D. Guo, S. Liu, and H. Jin. A graph-based approach to vehicle trajectory analysis. *J. Locat. Based Serv.*, 4 :183–199, September 2010.

[13] Z. Harris. Distributional structure. *Word*, 10(23) :146–162, 1954.

[14] J.-R. Hwang, H.-Y. Kang, and K.-J. Li. Spatio-temporal similarity analysis between trajectories on road networks. In *ER (Workshops)*, volume 3770 of *Lecture Notes in Computer Science*, pages 280–289. Springer, 2005.

[15] H. Jeung, H. T. Shen, and X. Zhou. Convoy queries in spatio-temporal databases. In *ICDE '08 : Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, pages 1457–1459, Washington, DC, USA, 2008. IEEE Computer Society.

[16] A. Kharrat, I. S. Popa, K. Zeitouni, and S. Faiz. Clustering algorithm for network constraint trajectories. In *SDH*, Lecture Notes in Geoinformation and Cartography, pages 631–647. Springer, 2008.

[17] A. Kharrat, I. S. Popa, K. Zeitouni, and S. Faiz. Caractérisation de la densité de trafic et de son évolution à partir de trajectoires d'objets mobiles. In D. Menga and F. Sedes, editors, *UbiMob*, volume 394 of *ACM International Conference Proceeding Series*, pages 33–40. ACM, 2009.

[18] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering : a partition-and-group framework. In *SIGMOD '07 : Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604, New York, NY, USA, 2007. ACM.

[19] B. Lin and J. Su. Shapes based trajectory queries for moving objects. In *GIS '05 : Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pages 21–30, New York, NY, USA, 2005. ACM.

[20] M. Nanni and D. Pedreschi. Time-focused clustering of trajectories of moving objects. *J. Intell. Inf. Syst.*, 27(3) :267–289, 2006.

[21] G.-P. Roh and S.-w. Hwang. Nncluster : An efficient clustering algorithm for road network trajectories. In *Database Systems for Advanced Applications*, volume 5982 of *Lecture Notes in Computer Science*, pages 47–61. Springer Berlin - Heidelberg, 2010.

[22] F. Rossi and N. Villa-Vialaneix. Représentation hiérarchique d'un grand réseau à partir d'une classification hiérarchique de ses sommets. *Journal de la Société Française de Statistique*, 152 :34–65, In press 2011.

[23] E. Tiakas, A. N. Papadopoulos, A. Nanopoulos, Y. Manolopoulos, D. Stojanovic, and S. Djordjevic-Kajan. Trajectory similarity search in spatial networks. In *Proceedings of the 10th International Database Engineering and Applications Symposium*, pages 185–192, Washington, DC, USA, 2006. IEEE Computer Society.

[24] M. Vlachos, D. Gunopoulos, and G. Kollios. Discovering similar multidimensional trajectories. In *ICDE '02 : Proceedings of the 18th International Conference on Data Engineering*, page 673, Washington, DC, USA, 2002. IEEE Computer Society.

[25] T. Zhang, R. Ramakrishnan, and M. Livny. Birch : an efficient data clustering method for very large databases. *SIGMOD Rec.*, 25(2) :103–114, 1996.

[26] Y. Zhao and G. Karypis. Criterion functions for document clustering : Experiments and analysis. Technical report, 2002.