

Clustering de trajectoires contraintes par un réseau

Ahmed KHARRAT*, Karine ZEITOUNI*
Sami FAIZ**

* Laboratoire PRISM 45, avenue des états unis
78035 Versailles

Ahmed.kharrat@prism.uvsq.fr, Karine.Zeintouni@prism.uvsq.fr
<http://www.prism.uvsq.fr/users/karima/>

**INSAT, Tunisie
Sami.Faiz@insat.rnu.tn

Résumé. Cet article se place dans le cadre de la fouille de données spatiotemporelle. Il propose une méthode de classification automatique des trajectoires d'objets mobiles dont le mouvement est contraint par un réseau routier. Cette méthode est basée sur la densité du réseau. Elle se compose de deux phases : la première découvre des chemins denses, puis la seconde s'appuie sur ces chemins denses pour former des groupes de trajectoires - ou plus exactement de sous-trajectoires - qui leurs sont similaires. Des paramètres donnés par l'utilisateur permettent de définir le seuil de densité pour les chemins denses ainsi que le seuil de similarité au sein des clusters.

1 Introduction

La gestion d'objets mobiles a connu un regain d'intérêt ces dernières années, encouragée par la disponibilité d'outils de localisation, à travers les téléphones cellulaires, les GPS ou récemment le RFID (Radio Frequency IDentification). Il devient possible de générer des bases de trajectoires d'objets mobiles à une large échelle. Par exemple, des applications de contrôle et de prévision du trafic utilisent des flottes de véhicules équipées de GPS comme des sondeurs du trafic. Leurs relevés GPS (appelés *Floating Car Data*) alimentent une base de trajectoires d'objets mobiles. Dans ce cas, comme dans la majorité des cas, le mouvement de l'objet mobile est contraint par le réseau. On parle de trajectoire contrainte par un réseau.

Ces applications ont généré de nouveaux problèmes qui ont motivé la recherche sur la gestion d'objets mobiles, en général et sur la fouille de données spatio-temporelle, en particulier. Le clustering des trajectoires fait partie de ces recherches. C'est une technique de fouille de données (ou datamining) largement utilisée dans des applications telles que l'étude de marché, l'analyse financière ou encore le traitement d'images. Plusieurs types d'algorithmes de clustering ont été proposés dont K-means (Lloyd, 1981), BIRCH (Zhang et al., 1996), DBSCAN (Ester et al., 1996) et OPTICS (Ankerst et al., 1999). Les recherches récentes sur le clustering de trajectoires d'objets mobiles ont utilisé ces algorithmes en les adaptant au domaine étudié (Lee et al., 2007).

Notre travail s'inspire notamment de l'algorithme DBSCAN (Ester et al., 1996). L'idée clé derrière notre approche est que la connaissance de la densité du trafic sur le réseau per-

Clustering de trajectoires contraintes par un réseau.

mettrait de guider le clustering des trajectoires. Dans un premier temps, nous définissons la similarité entre sections de routes et l'utilisons pour les grouper en chemins denses. Dans un deuxième temps, nous proposons une mesure de similarité entre trajectoires, puis nous nous servons pour former des groupes autour des chemins denses. Comme dans (Lee et al., 2007), le temps n'est pas considéré dans notre approche (on parle de temps relaxé ou *time relaxed*). Sommairement, les contributions de cet article sont les suivantes :

- Nous proposons une méthode de clustering de trajectoires d'objets mobiles complètement novatrice basée sur la densité du réseau. Tout comme l'approche de (Lee et al., 2007), cette méthode ne considère pas l'aspect temporel - bien qu'elle respecte le sens de parcours des trajectoires - et regroupe des sous-trajectoires plutôt que des trajectoires entières. Ainsi, une trajectoire peut appartenir à plusieurs clusters. Cependant, la méthode proposée tire profit des propriétés du réseau pour générer des groupes pertinents.
- Nous implémentons notre approche par un algorithme en deux phases permettant dans la première de grouper les sections de route et dans la seconde de grouper les trajectoires.
- Nous définissons de nouvelles fonctions de similarité.

Le reste de cet article est organisé comme suit. Nous détaillerons dans la section 2 un état de l'art sur la similarité et le clustering des trajectoires. Nous expliquerons ensuite la démarche de clustering dans la section 3. Nous présenterons dans la section 4 la première phase de l'algorithme - appelé NETSCAN - pour le clustering des sections de route. Nous décrirons ensuite la deuxième phase de l'algorithme. Enfin, nous conclurons cet article à la section 6 et proposerons des pistes pour la poursuite de cette recherche.

2 Etat de l'art

La recherche sur le clustering de trajectoires d'objets mobiles est étroitement liée à trois sujets qui sont la représentation de trajectoires, la similarité et les algorithmes de clustering proprement dits. De manière orthogonale, nous distinguons la prise en compte des critères suivants : l'aspect contraint ou libre de la trajectoire, l'aspect temporel, le respect du sens de parcours et enfin la prise en compte d'une partie ou de la totalité de la trajectoire dans le clustering. Cette section décrit les principaux travaux traitant de ces trois sujets en les situant par rapport aux critères ci-dessus.

Concernant la représentation de trajectoires, celle-ci peut-être soit géométrique (Lee et al., 2007), soit symbolique (Hadjieleftheriou et al., 2005) pour les trajectoires évoluant dans un réseau. En effet, si l'on connaît d'avance la géométrie et la topologie du réseau, une représentation naturelle de la trajectoire peut être donnée par la liste des sections traversées avec éventuellement l'instant auquel l'objet est passé d'une section à une autre si l'on représente l'aspect temporel. Cette représentation est très précise au niveau spatial, mais peut-être moins précise au niveau temporel. Néanmoins, elle peut suffire dans de nombreux cas et particulièrement dans notre contexte où le temps n'est pas pris en compte.

Concernant les travaux sur la similarité de trajectoires d'objets mobiles, nous citons d'abord les travaux dans le contexte de trajectoires libres, puis pour des trajectoires contraintes.

Yanagiswa et al. (2003) proposent d'abord une modélisation par la forme 2D. L'aspect spatial y est représenté par des vecteurs de même longueur, tandis que la description spatio-temporelle est obtenue par des vecteurs de longueurs différentes, mais à intervalle de temps régulier. Ainsi, la même formule de distance euclidienne entre ces vecteurs permet de définir la similarité spatiale et spatio-temporelle. Shim et Chang (2003) considèrent la similarité des sous-trajectoires et ont proposé un algorithme de distance 'K-warping'. Lin et al. (2005) ne considèrent que l'aspect spatial dans leur calcul de similarité et proposent une nouvelle distance OWD (One Way Distance). On trouve des travaux similaires dans Valachos et al. (2003), Sakurai et al. (2005), et Chen et al. (2005). Valachos et al. (2002) se sont intéressés aux valeurs extrêmes (outliers) qui dégradent les performances de la distance euclidienne et la déformation du temps (le time warping). Par conséquent, ils proposent l'utilisation d'une fonction de distance non métrique et qui se base sur l'algorithme de la plus longue sous séquence commune (LCSS) avec la conjonction de la fonction SigmoidMatch pour aligner (matching) les séquences de deux trajectoires. Zeinalipour-Yazti et al. (2006) introduisent une recherche de similarité spatiotemporelle distribuée basée sur la mesure de distance LCSS et proposent deux nouveaux algorithmes offrant de bonnes performances.

Toutes ces méthodes ne sont pas appropriées au calcul de similarités sur le réseau routier puisqu'elles se basent sur la distance euclidienne et non de la distance réelle, c'est à dire par la route, sur le réseau routier. C'est ce dernier point qui a motivé la proposition de Hwang et al. (2005) qui ont été les premiers à proposer une mesure de similarité basée sur la distance spatiotemporelle entre deux trajectoires utilisant la distance par la route. L'algorithme de recherche de trajectoires similaires consiste en deux étapes : l'étape de filtrage basée sur la similarité spatiale sur le réseau routier et l'étape de raffinement pour la recherche de trajectoires similaires à base de distance temporelle. Tiakas et al. (2006) et Chang et al. (2007) utilisent également une distance spatiotemporelle, basée sur le réseau routier, dans leur algorithme de recherche de trajectoires similaires.

Pour les travaux sur le clustering de trajectoires, nous citons les deux suivants :

Gaffney et Smyth (1999) ont soutenu que la représentation de trajectoire à base de vecteurs est inadéquate dans plusieurs cas. Pour surmonter cette lacune ils ont introduit un modèle de mixtures de régression probabiliste et ils ont montré comment l'algorithme EM pourrait être utilisée dans le clustering des trajectoires. Cette approche considère des trajectoires non contraintes. De plus, elle groupe les trajectoires entières et non des sous-trajectoires dans un cluster.

Lee et al. (2007) proposent eux un algorithme appelé TRACCLUS qui groupe des sous-trajectoires similaires dans chaque cluster. Il consiste en deux phases : le partitionnement des trajectoires sous forme de segments de droites et puis le groupage de ces derniers selon leurs similarités. Néanmoins, ce travail suppose toujours un mouvement non contraint de trajectoires. De plus, il ne considère pas le temps.

Aucun des travaux précédents ne permet le clustering de sous-trajectoires contraintes par le réseau.

3 Procédure de clustering

Le clustering consiste à former, à partir d'une base de données, des groupes d'objets similaires (Han et Kamber 2006). On entend généralement par procédure de clustering l'ensemble des étapes devant être réalisées depuis la mise à disposition d'un jeu de données jusqu'au traitement des résultats obtenus suite au partitionnement de l'espace des données en clusters. Classiquement, les procédures de clustering comprennent les étapes suivantes dans l'ordre : (i) Représentation des données, (ii) Définition d'un critère de similarité, (iii) Clustering, (iv) Abstraction des données (v) Evaluation de la qualité du clustering.

Nous proposons une méthode de clustering en deux phases. La première permet de grouper les sections de route. Nous parlerons de clustering de sections par abus de langage. La seconde phase effectue véritablement le clustering des trajectoires. Étant donné que ces types de clustering portent sur des objets complexes, nous devons préciser pour chaque phase la représentation, la similarité et l'algorithme spécifique de clustering.

3.1 Représentation des données

La représentation du réseau est donnée par l'ensemble de sections de route. Par ailleurs, connaissant l'ensemble des trajectoires, nous calculons une matrice de transitions associée au réseau routier (cf. FIG. 1). Celle-ci fournit des statistiques sur les passages aux carrefours et les mouvements tournants, en reportant le nombre d'objets mobiles qui transitent d'une section à une section connexe. Cette matrice sera notée par la suite M et une cellule $M(i,j)$ dénote le nombre d'objets mobiles traversant de la section S_i vers la section S_j .

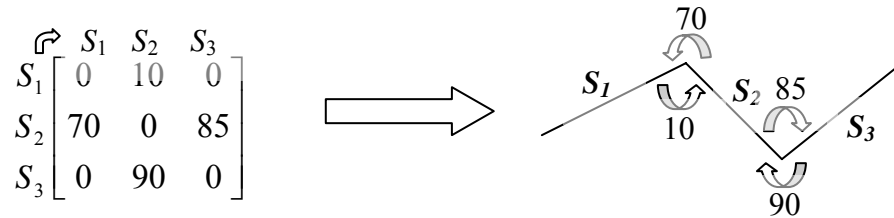


FIG.1 - Matrice de transitions attribuée à un réseau.

Nous adoptons une représentation symbolique des trajectoires (Du Mouza, 2005), (Wan et Zeitouni, 2005). Dans cette représentation, une trajectoire d'objets mobiles est une séquence de symboles dont chacune se réfère à une section de route.

$$TR = \langle S_{i1}, \dots, S_{in} \rangle$$

L'ordre de ces symboles détermine le sens du déplacement.

3.2 Critères de similarité

La similarité représente la base de l'opération de clustering. Nous définissons la similarité à deux niveaux. Au niveau du réseau, la similarité est calculée entre deux transitions

comme la différence de leurs valeurs de densité. À noter que cette mesure ne concerne que des transitions consécutives.

$$Sim_route (M(i,j), M(j,k)) = |M(i,j) - M(j,k)|$$

Au niveau des trajectoires, nous définissons une mesure de similarité entre deux trajectoires dont une est la référence. Cette mesure reflète une ressemblance à un objet et n'est pas symétrique. Elle permet de comparer les trajectoires effectives à une trajectoire fictive type. Pour cela, la similarité est calculée comme le rapport entre la longueur commune entre une trajectoire et la référence d'un côté, et la longueur de la trajectoire référence de l'autre. La longueur d'une trajectoire ou de la partie commune des trajectoires est égale au nombre de sections qu'elle contient.

$$Sim_traj = \text{Longueur (partie_commune)} / \text{longueur (traj_réf)}$$

Par comparaison aux travaux cités dans la section 2, la similarité y est basée sur la distance euclidienne et/ou de formes (Yanagisawa et al., 2003). Ceci vient du fait qu'ils représentent les trajectoires par leur géométrie et leur forme. Notre travail adopte un critère tout à fait différent car il se place dans le contexte contraint. Il utilise l'information disponible sur la densité du réseau, d'une part et la représentation symbolique des trajectoires permettant de se ramener à des similarités de séquences comme dans Chen et al. (2005), d'autre part.

3.3 Clustering

L'étape du clustering correspond à la phase de regroupement proprement dite et mène à l'obtention d'une partition de la base de données aussi pertinente que possible. C'est à ce niveau, que nous proposons un algorithme de clustering nommé NETSCAN en deux étapes. Dans une première étape, il génère les clusters de sections de route les plus fréquentées formant chacun un chemin dense sur le réseau routier. La seconde étape permet de classifier les trajectoires d'objets mobiles selon ces chemins denses. La figure 2 résume les principales étapes de NETSCAN. Notre algorithme partage les mêmes caractéristiques que l'algorithme DBSCAN et se met en oeuvre en deux étapes :

1. Groupement de sections : Trouver les chemins du réseau les plus denses en termes d'objets mobiles qui y transitent.
2. Groupement des trajectoires : Pour chacun de ces chemins, regrouper la trajectoire qui lui est similaire.

Clustering de trajectoires contraintes par un réseau.

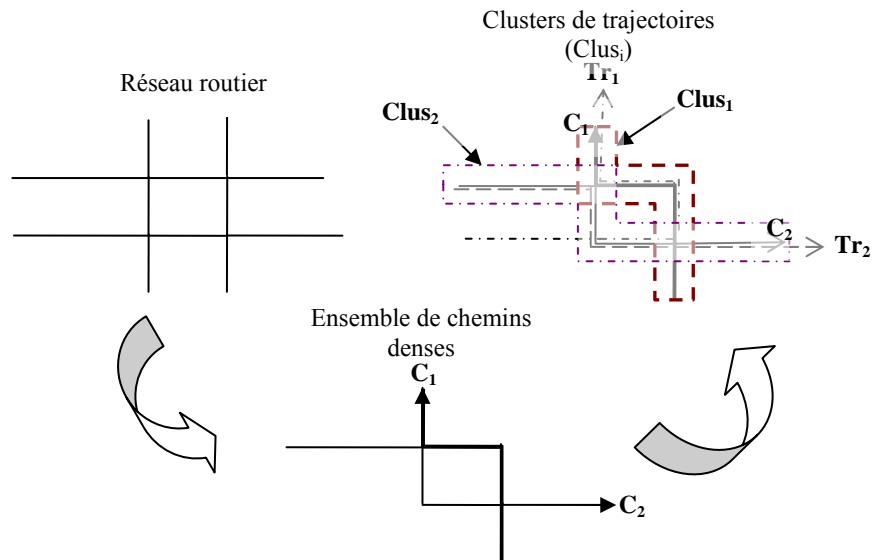


FIG.2 – Exemple de clustering de trajectoire via l'algorithme NETSCAN

4 Clustering des sections

L'algorithme proposé s'appelle NETSCAN. Il se compose de deux parties. La première, NETSCAN-PHASE 1 décrite ici, effectue le clustering des sections. Cette première phase s'inspire du principe de densité introduit dans l'algorithme DBSCAN (Ester et al., 1996) en l'appliquant sur des sections de routes. Elle prend en entrée l'ensemble de sections qui constituent le réseau routier, la matrice de transitions entre les sections de routes (cf. section 3.1), un seuil α de densité et un seuil ε de similarité entre les densités des transitions. Lors de cette phase, l'algorithme regroupe les sections où transite le maximum d'objets mobiles en premier (les transitions denses). Il groupe ensuite les transitions connexes dont les densités sont similaires constituant ainsi des chemins denses.

Pour ce faire, le processus commence par la transition ayant la densité maximum. Ensuite, le groupage s'étend aux transitions connexes dans les deux sens pour chercher celles ayant une densité proche à ε près. De cette façon, de proche en proche, on génère un « chemin dense ». Pour assurer la non-réutilisation des transitions figurant dans des chemins denses, on les marque à la première affectation.

L'extension du chemin dense se fait dans les deux sens si les conditions sont remplies, à savoir le non marquage de la transition à rajouter, le respect des seuils de densité α et de similarité ε . Les clusters de sections obtenus correspondent aux chemins les plus denses sur le réseau. La figure 3 décrit la première phase de l'algorithme NETSCAN.

Tout comme les trajectoires, les chemins denses sont représentés sous forme de séquences de sections (cf. section 3.1). Chaque section est donnée par un symbole qui l'identifie.

 Algorithmme NETSCAN-PHASE 1 /* Découverte d'itinéraires denses*/

Entrée :

- Ensemble de sections $S = \{S_1, S_2, \dots, S_{nbsections}\}$
- Matrice de transitions M .
- Seuil ε -- écart maximal de densité entre sections voisines.
- Seuil α -- densité minimum requise dans une transition d'un chemin.

Sortie :

- Ensemble ordonné de chemins $O = \langle C_1, C_2, \dots, C_{nbchemins} \rangle$.

Algorithmme :

1. $O \leftarrow$ vide -- Initialisation
2. $k \leftarrow 0$
3. Tant qu'il existe des transitions $M(i,j) \geq \alpha$ et non marquées
4. $k++$
5. $M(d,f) = \max (M [i, j])$
6. $C_k \leftarrow \langle S_d, S_f \rangle$ -- on génère un nouveau chemin dense avec cette transition
7. Marquer la transition $M(d,f)$
8. Tant qu'il existe u tel que $M(f,u) \geq \alpha$ et non marquée – extension en avant
9. Sélectionner $M(f, f_succ)$ telle que $|M(d,f) - M(f,u)|$ est minimum
10. Si $|M(d,f) - M(f, f_succ)| \leq \varepsilon$
11. Insérer queue (C_k, S_{f_succ})
12. Marquer $M(f, f_succ)$
13. $D \leftarrow f ; f \leftarrow f_succ$ -- étendre le chemin
14. Sinon Sortie de la boucle
15. Tant qu'il existe u tel que $M(u,d) \geq \alpha$ et non marquée -- extension en arrière
16. Sélectionner $M(d_préd, d)$ telle que $|M(d,f) - M(u,d)|$ est minimum
17. Si $|M(d,f) - M(u,d)| \leq \varepsilon$
18. Insérer tête $(C_k, S_{d_préd})$
19. Marquer $M(d_préd, d)$
20. $F \leftarrow d ; d \leftarrow d_préd$ -- étendre le chemin
21. Sinon Sortie de la boucle
22. Ajout du chemin C_k à l'ensemble O
23. Retourner O

 FIG.3 – Algorithmme NETSCAN-Phase 1 : clustering des sections en chemins denses.

5 Clustering des trajectoires

Cette section présente la deuxième partie de l'algorithme NETSCAN correspondant au clustering de trajectoires. Celle-là se base sur le résultat obtenu par la première partie de l'algorithme présentée dans la section précédente. En effet, les chemins denses sont considérés comme des centres naturels de clusters de trajectoires d'objets mobiles.

Clustering de trajectoires contraintes par un réseau.

L'algorithme de clustering de trajectoires consiste à grouper les trajectoires selon leurs similarités avec chacun des chemins denses générés lors de la phase 1 de NETSCAN. La mesure de similarité utilisée est celle définie dans la section 3.2. L'algorithme prend en entrée l'ensemble de chemins denses, l'ensemble des trajectoires et le seuil σ . Il parcourt dans l'ordre les chemins denses et calcule pour chacun sa similarité avec chaque trajectoire d'objets mobiles. Si cette similarité répond au seuil σ , alors la trajectoire est retenue dans le cluster. Plus exactement, la partie commune de la trajectoire avec le chemin dense est ajoutée au cluster. Le nombre de clusters retournés à la fin de l'algorithme est égal au nombre de chemins denses.

 Algorithme NETSCAN-PHASE 2

Entrée :

- Ensemble de chemins récupérés de l'algorithme NETSCAN-PHASE 1
 $O = \langle C_1, C_2, \dots, C_{nbchemins} \rangle$
- Ensemble de trajectoires d'objets mobiles $TR = \langle TR_1, TR_2, \dots, TR_{nbtrajectoires} \rangle$
- Seuil σ -- seuil minimal de similarité

Sortie :

- Ensemble de clusters $Clus = \{Clus_1, \dots, Clus_{nbchemins}\}$

Algorithme :

1. Pour ($i=0$; $i < nbchemins$; $i++$)
2. Pour ($j=0$; $j < nbtrajectoires$; $j++$)
3. Calculer la séquence $Sc = C_i \cap Tr_j$ -- ensemble des sections communes
4. Calculer la somme des longueurs des sections commune

$$Lc = \sum_{\substack{i=0 \\ nbde \\ communes}}^{i=0} longueur (Sc)$$

5. Mesurer la similarité entre C_i et Tr_j en utilisant la formule : $Sim = Lc / longueur(C_i)$
6. Si $sim \geq \sigma$
7. Ajouter Sc à $Clus_i$
8. Ajouter $Clus_i$ à l'ensemble $Clus$
9. Retourner $Clus$

 FIG.4 – Algorithme NETSCAN-Phase2 : Clustering des trajectoires.

6 Conclusion et Perspectives

Cet article se situe dans le cadre de la fouille de données spatio-temporelles. Plus précisément, il se focalise sur le clustering pour l'adapter aux objets mobiles contraints par le réseau routier. Nous avons proposé un algorithme de clustering en deux phases. La première traite des sections de routes pour obtenir les chemins les plus denses sur le réseau et la seconde traite des trajectoires afin d'obtenir des classes de trajectoires similaires.

Comme futur travail, nous allons implémenter notre algorithme pour évaluer ses performances. Dans sa forme actuelle, l'algorithme de clustering de trajectoires parcourt la base de données autant de fois qu'il n'y a de chemins denses. Une solution plus optimale devra être mise en œuvre afin de réduire la complexité algorithmique, notamment en terme d'entrées / sorties. L'utilisation d'une solution par index spatial comme le TTR-tree proposé dans Wan et Zeitouni (2006) permettrait un gain significatif de performances.

Nous étudierons ensuite la prise en compte du temps, ainsi que d'autres mesures rattachées à la localisation des trajectoires. Pour cela, nous garderons et nous étendrons notre représentation symbolique par référence aux sections du réseau routier. Cela implique l'extension de la matrice de transition par la dimension temporelle ou de mesure. La similarité devra également être étendue et les algorithmes adaptés et optimisés.

Références

- Ankerst M., M. M. Breunig, H.-P. Kriegel et J. Sander (1999) *OPTICS: Ordering Points to Identify the Clustering Structure*, In Proc. 1999 ACM SIGMOD Int'l Conf. on Management of Data, Philadelphia, Pennsylvania, pp. 46-60.
- Chang J-W., R. Bista, Y-C. Kim et Y-K Kim (2007) *Spatio-temporal Similarity Measure Algorithm for Moving Objects on Spatial Networks*. ICCSA 2007, pp.1165-1178.
- Chen L., M.T. Ozsü et V. Oria (2005) *Robust and Fast Similarity Search for Moving Object Trajectories*. In: ACM SIGMOD, pp. 491-502. ACM Press, New York.
- Du Mouza C. (2005) *Patterns de mobilité*. Thèse, Chapitre 3 : Patterns de mobilité, p. 51-66.
- Ester M., H.-P. Kriegel, J. Sander et X. Xu (1996) *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, In Proc. 2nd Int'l Conf. on Knowledge Discovery and Data Mining, Portland, Oregon, pp. 226-231.
- Gaffney S. et P. Smyth, (1999) *Trajectory Clustering with Mixtures of Regression Models*, In Proc. 5th ACM SIGMOD Int'l Conf. on knowledge Discovery and Data Mining, San Diego, California, pp. 63-72.
- Hadjieleftheriou M., G. Kollios, P. Bakalov, V. Trotras (2005) *Complex Spatio-Temporal Pattern Queries*. In Proc. of the 31st VLDB Conference.
- Han J. et M. Kamber (2006) *Data Mining: Concepts and Techniques*, 2nd ed., Morgan Kaufmann.
- Hwang J-R., H-Y. Kang et K-J. Li (2005) *Spatio-temporal Analysis Between Trajectories on Road Networks*. ER Workshops 2005, LNCS 3770, pp. 280-289.
- Lee J-G, J. Han et K-Y. Whang (2007) *Trajectory Clustering: A Partition-and-Group Framework*. In Proc.SIGMOD'07, Beijing, China.
- Lin B., J. Su (2005) *Shapes Based Trajectory Queries for Moving Objects*. GIS, pp. 21-30.
- Lloyd S. (1982) *Least Squares Quantization in PCM*, IEEE Trans. on Information Theory, 28(2): 129-137.

Clustering de trajectoires contraintes par un réseau.

- Sakurai Y., M. Yoshikawa et C. Faloutsos (2005) *FTW: Fast Similarity Search Under the Time Warping Distance*. In: PODS, pp. 326-337.
- Shim C-B et J-W Chang (2003) *Similar Sub-Trajectory Retrieval for Moving Objects in Spatiotemporal Databases*. In: Proc. of the 7th EECADIS, pp.308-322.
- Tiakas E., A. N. Papadopoulos, A. Nanopoulos et Y. Manolopoulos (2006) *Trajectory Similarity Search in Spatial Networks*. In : Proc. of the 10th IDEAS, pp. 185-192.
- Vlachos M., D. Gunopulos et G. Kollios (2002) *Robust Similarity Measures of Mobile Object Trajectories*. In: Proc. of the 13th Intl. Workshop on DEXA, IEEE Computer Society Press, Los Alamitos pp. 721-728.
- Vlachos M., G. Kollios et D. Gunopulos (2002) *Discovering Similar Multidimensional Trajectories*. In: Proc. Of the 18th ICDE. IEEE Computer Society Press, Los Alamitos pp. 673-684.
- Wan T., K. Zeitouni (2005) *Modélisation d'objets mobiles dans un entrepôt de données*. EGC 2005: pp. 343-348.
- Wan T., K. Zeitouni (2006) *Représentation et indexation d'objets mobiles contraints par le réseau dans un entrepôt de données*, 2^{ème} Conférence Entrepôts de données et Analyse en ligne, EDA 2006, Revue Nouvelles Technologies de l'Information (RNTI), Cépaduès Edition, 2006, pp. 139-154.
- Yanagisawa Y., J. Akahani, T. Satoh (2003) *Shape-Based Similarity Query for Trajectory of Mobile Objects*. In : Proc. Of the 4th Intl. Conf. On MDM, pp. 63-77.
- Zeinalipour-Yazti D., S. Song Lin, D. Gunopulos (2006) *Distributed Spatio-Temporal Similarity Search*. CIKM, pp. 14-23.
- Zhang T., R. Ramakrishnan, et M. Livny (1996) *BIRCH: An Efficient Data Clustering Method for Very Large Databases*. In Proc. ACM SIGMOD Int'l Conf. on Management of Data, Montreal, Canada, pp. 103-114.

Summary

In this paper, we proposed a new clustering density based method into moving object trajectories database. The proposed algorithm (NETSCAN) is two steps: the first one computes dense paths in the network while the second step clusters the moving object trajectories with the dense paths using an adapted similarity measure. Our work assumes time is relaxed constraint. The advantage of this framework is to discover groups of common sub-trajectories from a trajectory database.