Classification non Supervisée de Trajectoires

Philippe Besse [1] & Brendan Guillouet [2] & Jean-Michel Loubes [3] & François Royer [4]

[1] *Université de Toulouse INSA, Institut de Mathématiques UMR CNRS 5219, philippe.besse@math.univ-toulouse.fr*
[2] *Institut de Mathématiques UMR CNRS 5219, brendan.guillouet@math.univ-toulouse.fr*
[3] *Université de Toulouse UT3, Insitut de Mathématiques UMR CNRS5219, jean-michel.loubes@math.univ-toulouse.fr*
[4] *Datasio, 26-28 rue Marie Magné, 3300 Toulouse, froyer@datasio.fr*

Résumé. Les trajectoires sont des suites de points du plan indexés par le temps. Pour pouvoir les comparer il faut prendre en compte non seulement le point de départ et le point d'arrivée, qui définissent l'itinéraire, mais également leur longueur et leur forme. Ainsi les méthodes usuelles de classification ne permettent pas de bien différencier les observations. Notre objectif est ici de fournir une nouvelle méthodologie de classification non supervisée des données de trajectoires se basant sur l'utilisation d'une distance particulière adaptée à ce type de données.

Mots-clés. Distance liée à la forme des trajectoires, classification non supervisée de trajectoires.

Abstract. Trajectories are modeled as a sequence of consecutive points indexed by time. To compare them we have to consider, not only the start point and the end point but also the length and the global shape of the trajectory. Classical clustering methods are not suitable for this kind of data. Our goal here is to establish a new method to group trajectories. This entails defining a distance to compare trajectories, without time dimension.

Keywords. Shape-based distance for trajectory, Trajectory clustering.

Introduction

Il est aujourd'hui très facile de collecter et stocker des données mobiles en très grandes quantités et pour un coût très faible (GPS, Smartphones,...). Ce type de données spatio-temporelles soulève un ensemble de questions classiques en fouille de données mais plus originales lorsqu'elles sont regroupées pour constituer des trajectoires. Peut-on définir une trajectoire type ? Peut-on détecter des trajectoires anormales ? Comment regrouper les trajectoires similaires ?

Le principe d'une telle étude repose sur la création d'une distance adaptée aux propriétés particulières que présentent les données de trajectoires. De nombreuses distances ont déjà été développées en vue de cet objectif

— Des distances basées sur l'appariement de points similaires. Le *Dynamic Time Warping* (DTW) [3], *Longest Common Subsequence* (LCSS) [10], *Edit distance with Real Penalty* (ERP) [4], *Edit distance on Real Sequences* (EDR)[5], se basent sur une re-paramétrisation de l'index temporel, mais cela ne suffit pas à corriger les distorsions dans le cas des trajectoires géolocalisées.
— Des distances utilisant la géométrie des courbes. La distance proposée par Lee & Han (2007) [7] est une distance de segment à segment, indépendante du temps et permet de détecter les segments similaires entre trajectoires, mais pas leur forme globale.
— Des distances utilisant la forme géométrique des trajectoires en recherchant des déformations. Srivastava et al. (2011) [9] présentent une distance qui permet de détecter très efficacement des formes similaires, mais imposent à celles-ci d'être de même longueur et est indépendante de la distance physique entre deux trajectoires. Lin et Su [8] ont mis en place la distance OWD, *One-Way-Distance*, qui se base uniquement sur la forme des trajectoires.

Nous allons présenter ici une adaptation de la distance OWD à la structure de nos trajectoires. Celle-ci se base sur les critères suivants : la distance physique entre deux trajectoires, la forme des trajectoires (orientation, longueur) et l"indépendance temporelle".

La démarche est illustrée par des données GPS publiques provenant de taxis [1].

1 Distances entre trajectoires

1.1 Définitions & Notations

Une trajectoire T^j est une une fonction qui, à un temps donné, parmi son ensemble de définition $\mathbf{t}^j \in \mathbb{R}$, associe un point dans $\mathbb{R}^2$:

$$\begin{array}{rccc} T^j & : & \mathbf{t}^j & \rightarrow & \mathbb{R}^2, \\ & & t_i^j & \mapsto & T^j(t_i^j) = P_i^j. \end{array}$$

$\mathbf{t}^j = [t_1^j, t_2^j, ..., t_{n_j}^j]$: Temps auxquels la trajectoire est observée,
n_j : Nombre de points constituant la trajectoire T^j,

Une trajectoire T^j composée de n_j points peut être définie comme une succession de points : $T^j = P_1^j, P_2^j, .., P_{n_j}^j$ et de segments : $T^j = S_1^j, S_2^j, ..., S_{n_j-1}^j$. Ou S_i^j est le segment délimité par deux points successifs P_i^j et P_{i+1}^j.

1.2 Distances

La distance entre un point et un segment est la distance utilisée en géométrie euclidienne :

Définition 1 *Soit P^1_{i1} un point et S^2_{i2} un segment,* ***la distance entre un point et un segment****, D_{PS} est définie par :*

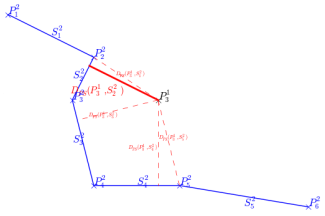
$$D_{PS}(P^1_{i_1}, S^2_{i_2}) = \begin{cases} \overline{P^1_{i_1} P^{1Proj}_{i_1}} & \text{si } P^{1Proj}_{i_1} \in S^2_{i_2}, \\ \min(\overline{P^1_{i_1} P^2_{i_2}}, \overline{P^1_{i_1} P^2_{i_2+1}}) & \text{sinon.} \end{cases}$$

Où $P^{1Proj}_{i_1}$ est la projection du point $P^1_{i_1}$ sur la droite prolongeant le segment $S^2_{i_2}$.

La distance D_{PT} d'un point à une trajectoire, est le minimum des distances entre ce point et les segments constituant cette trajectoire.

Définition 2 *Soit P^1_{i1} un point et T^2 une trajectoire,* ***la distance entre un point et une trajectoire****, D_{PT} est définie ainsi :*

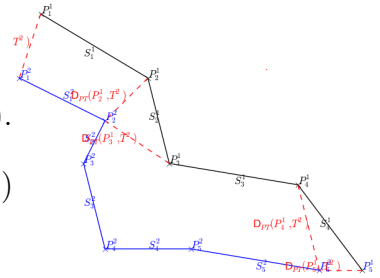
$$D_{PT}(P^1_{i_1}, T^2) = \min_{i_2 \in [0,\dots,n_2-1]} D_{PS}(P^1_{i_1}, S^2_{i_2})$$



Nous définissons dans un premier temps la distance $D_{MOWD}(T^1, T^2)$ de la trajectoire T^1 à la trajectoire T^2, comme la moyenne des distances des points de T^1 à la trajectoire T^2 :

Définition 3 *Soit T^1 et T^2, deux trajectoires, alors $D_{MOWD}(T^1, T^2)$ est définie ainsi :*

$$D_{MOWD}(T^1, T^2) = \frac{1}{n_1} \sum_{i_1=1}^{n_1} D_{PT}(P^1_{i_1}, T^2). \quad (1)$$

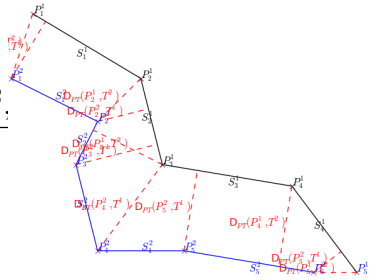


Cette distance n'est pas symétrique. On notera que si T^1 est une sous-trajectoire de T^2, $D_{MOWD}(T^1, T^2) = 0$, tandis que $D_{MOWD}(T^2, T^1) \neq 0$ et peut même être très grande.

La distance **"Symmetrized-MOWD"**, SMOWD, est la moyenne de ces deux distances.

Définition 4 *Soit T^2 et T^2, deux trajectoires, D_{SMOWD} est définie ainsi :*

$$D_{SMOWD}(T^1, T^2) = \frac{D_{MOWD}(T^1, T^2) + D_{MOWD}(T^2,}{2} \quad (2)$$



Ainsi cette distance permet à la fois de comparer les formes de ces trajectoires (en rapprochant d'une part les trajectoires ayant des parties communes), mais également leur distances totales (en les éloignant d'autre part si leurs terminaisons sont éloignées). Cette distance est entièrement indépendante du temps. Elle compare donc uniquement les formes de T^1 et T^2. Elle donne plus d'importance au point de la trajectoire issue des relevées GPS plutôt qu'aux autres points issus de l'interpolation entre ces relevés. La distance $SMOWD$ reflète bien les propriétés que nous souhaitons mettre en exergue sur les trajectoires. Nous allons l'utiliser pour classifier les trajectoires observées en utilisant un algorithme de propagation d'affinité.

2 Algorithme de classification

Le terme *distance* est utilisée abusivement pas soucis de simplification. La distance SMOWD n'en remplit pas les conditions : elle ne satisfait ni l'inégalité triangulaire, ni le principe d'identité des indiscernables. On parle alors d'indice de dissemblance.
Pour cette raison, les algorithmes de classification tel que *K-means* ne sont pas adaptés et PAM trop coûteux en temps.

La méthode de propagation d'affinité est un algorithme basé sur un principe de "*Message Passing*" [6]. Elle prend en paramètre la matrice de similarité S entre les éléments à classer. Elle peut-être utilisée avec n'importe qu'elle indice de similarité. Elle permet également d'élire un individu représentatif de chaque classe : l'*exemplar*. Contrairement à d'autre algorithme hiérarchique tel que *CAH*.
Cet algorithme vise à maximiser la similarité nette S définie comme la somme de l'énergie négative E et une fonction de contrainte δ.

$$S(c) = -E(c) + \sum_{k=1}^{N} \delta_k(c) = \sum_{k=1}^{N} s(i, c_i) + \sum_{k=1}^{N} \delta_k(c), \quad (3)$$

$$\text{où } \delta_k(c) = \begin{cases} -\infty \text{ if } c_k \neq k \text{ but } \exists i : c_i = k, \\ 0 \text{ sinon.} \end{cases} \quad (4)$$

Le terme de contrainte $\delta_k(c)$ oblige le point k à être son propre *exemplar* si au moins un autre point i l'a choisi comme son *exemplar*. A partir de l'équation 3 un "*factor graph*" est construit au sens définie par Frey et al. (2001) [2], puis l'algorithme de "*max-sum*" est appliqué sur ce graphe.

3 Résultats

Notre distance a été calculée sur 2802 trajectoires de taxis San-Franciscains. Toutes les trajectoires sélectionnées ont une origine commune, qui est la gare Caltrain de San-Francisco. L'algorithme de propagation d'affinité a ensuite été appliqué sur la matrice d'affinité $S = D_{max} - D$ ou D est la matrice de distance SMOWD. Nous avons implémenté cette distance sous python. L'algorithme de propagation d'affinité est celui disponible dans la librarie *scikit-learn*. Les résultats de cette application sont les suivants :

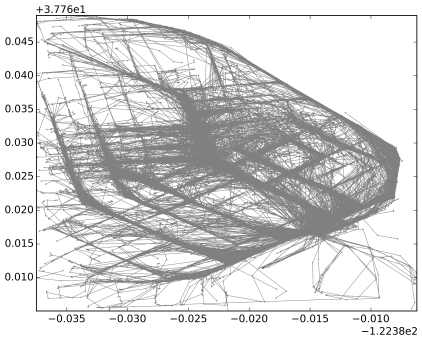


FIGURE 1 – 2802 Trajectoires de taxi

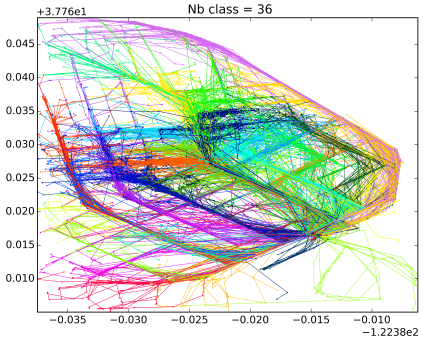


FIGURE 2 – Résultat de la classification

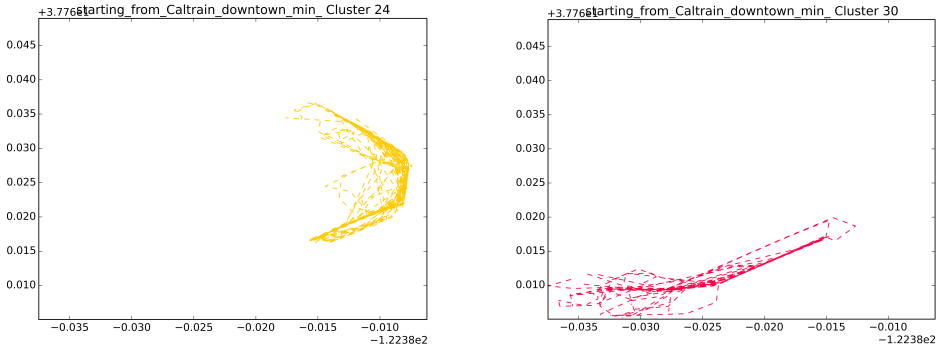


FIGURE 3 – Deux exemples de classes de trajectoires

Trente-six classes de trajectoires ont été obtenues. On peut observer que l'ensemble des trajectoires d'une même classe définissent un itinéraire type.

Lors de notre présentation, nous comparerons les résultats obtenues avec d'autres distances parmi celle utilisée usuellement ainsi que d'autres techniques de classifications. Nous présenterons également une technique permettant d'associer rapidement une classe à chaque nouvelle trajectoire. Cette association nous permettra d'établir une prédiction de l'évolution de cette trajectoire, en utilisant les trajectoires passées de la classe qui lui est associée.

Bibliographie

[1] Cabspotting. http ://www.cabspotting.org. 01-01-2013.

[2] F. R. Kschischang B. J. Frey and H-A. Loeliger. Factor graphs and the sum-product algorithm. *TRANSACTIONS ON INFORMATION THEORY*, 47 :498–519, Février 2001.

[3] D. J. Berndt. Using dynamic time warping to findpatterns in time series. *AAA1-94 Workshop on Knowledge Discovery in Databases*, pages 359–370, Avril 1994.

[4] L. Chen, M. Tamer Ozsu, and V. Oria. Robust and fast similarity search for moving object trajectories. *AProceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502, 2005.

[5] L. Chen and R.Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pages 792–803, 2004.

[6] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *sciencemag*, 315 :972–976, Février 2007.

[7] J-G. Lee and J. Han. Trajectory clustering : A partition-and-group framework. In *In SIGMOD*, pages 593–604, 2007.

[8] B. Lin and J. Su. Shapes based trajectory queries for moving objects. In *Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pages 21–30. ACM Press, 2005.

[9] A. Srivastava, E. Klassen nad S. H. Joshi, and I Jermyn. Shape analysis of elastic curves in euclidean spaces. *Issue No.07 - July (2011 vol.33)*, pages 1415–1428, 2011.

[10] M. Vlachos, G. Kollios, and D.Gunopulos. Discovering similar multidimensional trajectories. *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 673–684, Février 2002.