



HAL
open science

Review and comparison of similarity measures and community detection algorithms for clustering of network constrained trajectories

Noudéhouéno Lionel Jaderne Houssou, Jean-Loup Guillaume, Armelle Prigent

► To cite this version:

Noudéhouéno Lionel Jaderne Houssou, Jean-Loup Guillaume, Armelle Prigent. Review and comparison of similarity measures and community detection algorithms for clustering of network constrained trajectories. MARAMI, Oct 2017, La Rochelle, France. hal-02363974

HAL Id: hal-02363974

<https://hal.science/hal-02363974>

Submitted on 14 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Review and comparison of similarity measures and community detection algorithms for clustering of network constrained trajectories

Noudéhouéno Lionel Jaderne Houssou, Jean-loup Guillaume, Armelle Prigent

Laboratory L3I, University of La Rochelle, Avenue Michel Crépeau, 17042 La Rochelle, France

{noudehouenou.houssou1,jean-loup.guillaume,armelle.prigent}@univ-lr.fr

ABSTRACT. Trajectory analysis is a study field that is experiencing a renewed interest mainly due to the increase and availability of datasets, generated by users either in the real world by the means of sensors such as GPS, or in virtual environments thanks to the digital footprint they leave when visiting a commercial website for example. In this paper we present the process of converting raw trajectories into network-constrained ones and we review similarity measures for trajectories as well as how they can be used to represent trajectories as graphs. We also describe some approaches to cluster graph of trajectories. Then we assess the trajectories similarity measures and the algorithms for graph clustering. In particular we show that some similarity measures are inadequate for clustering graph of trajectories since they are not discriminative enough.

RÉSUMÉ.

KEYWORDS: complex networks, network-constrained trajectories, trajectory similarity, trajectory clustering

MOTS-CLÉS: graphes de terrain, trajectoires contraintes par un réseau, similarité de trajectoires, clustering de trajectoires

DOI:10.3166/RIA..1-13 © 2017 Lavoisier

1. Introduction

Trajectory analysis is a study field that is experiencing a renewed interest mainly due to the increase and availability of mobility datasets generated by sensors such as GPS. The access to that large amount of data allows researchers and data scientists to extract relevant knowledge like user behavior, road traffic and occupation or frequent mobility patterns that can be monetized by companies through value-added applications as intelligent transportation systems. Moreover, this knowledge is necessary to meet populations needs concerning smart and sustainable cities concept and to deal with the fast demographic evolution and the rapid economic development in some regions.

However, trajectories are not limited to human mobility measured with GPS. Other kinds of trajectories describing for example animal displacements in nature or pages to pages moves of website users are also studied. Independently from the way they are conducted, researches dealing with knowledge extraction through trajectory analysis are usually related to definition and comparison of distance or similarity measures, trajectories representation and clustering of trajectories. They also generally concern trajectories that occur in free space while network-constrained trajectories which are quite common has only been considered by a small handful of studies. Accordingly, we choose to focus on trajectories that are constrained by an underlining network (road network, hyperlinks, etc.) in our literature review.

For instance, in (Wang *et al.*, 2013) authors compare several distance measures. They transform taxi trajectories from the city of Beijing, by adding noise, re-sampling (addition, deletion of points) or deforming (points shifting) them. Then they assess the performances of the different distance measures by computing distance between transformed trajectories and their originals. They thus succeed in determining by experimentation the advantages and disadvantages of the studied distance measures.

In (Besse *et al.*, 2016), authors rather define a new distance measure between vehicle trajectories called Symmetrized Segment-Path Distance (SSPD). It has the advantage of being symmetrical and takes account of the geometrical characteristics of the trajectories. The authors compare their new distance to those commonly used in the literature using hierarchical clustering and clustering by affinity propagation and get good results.

(Hwang *et al.*, 2005) propose a method to retrieve similar trajectories from road network space. It consists in a first step to use Points Of Interest (POIs) like roads intersections or places to identify spatially similar trajectories. In the second step they use a temporal distance also based on POIs to refine the trajectory similarity evaluation. Then they apply a clustering using a shifted Hilbert curve to assess their method and get coherent results.

Unlike the previous works, (Guo *et al.*, 2010) define a graph-based representation of trajectories where nodes are trajectories points and where edges link only points that

share the same trajectory. They apply a spatially constrained graph partition method on the obtained graph to cluster trajectories by regions.

In the same way, (Khalil El Mahrsi *et al.*, 2015) propose a new representation of trajectories in graph form. The nodes of this graph correspond to the trajectories and are connected by edges whose weights indicate the similarity between trajectories. The authors also define a new similarity measure between trajectories similar to the TF-IDF (Term Frequency - Inverse Document Frequency) originally used for documents. They then apply a community detection method based on modularity to cluster the trajectories. These new approaches in trajectory representation and clustering has aroused our interest and we propose to enrich it by comparing different similarity measures and various methods of community detection for the clustering of trajectories represented as graph.

This paper present a review of some similarity measures and assess performances of two community detection algorithms used in the context of trajectories clustering. In Section 2 we present some preprocessing steps wich must be performed when confronted with GPS based trajectories that have to be fitted on a network. Then we present several measures of similarity between trajectories in Section 3 which are not dedicated to network-constrained trajectory but can be somehow adapted. In Section 4 we discuss the clustering of trajectories represented as graph. Finally, we present some experimental results using trajectories in the city of Porto in 5.

2. Trajectories preprocessing

When studying trajectories in real environments it is often necessary to associate raw mobility data from GPS to positions on a map. For instance, when studying car movements in a city, we expect that the recorded positions belong to road segments as displacements are mainly carried out on road networks and we therefore want to ensure that the trajectories do not go through buildings. The matching of GPS data to positions on a map is called *map matching* and several approaches have been proposed to deal with this problem. They are categorized in four groups as stated in (Quddus *et al.*, 2007):

- **Geometric approaches:** map matching is done point-to-point, point-to-curve or curve-to-curve. In point-to-point approaches, each recorded point is matched to the closest point on a road segment (Bernstein, Kornhauser, 1998) while for point-to-curve approaches each GPS point is associated to the closest road segment (White *et al.*, 2000) and for curve-to-curve approaches entire pieces of the recorded trajectory are matched to road segments on the map (Phuyal, 2002).

- **Topological approaches:** the similarity between the trajectories and the road network is assessed using the geometry of road segments as well as their topological properties (connectivity, adjacency, containment or contiguity) (Chen *et al.*, 2003; Greenfeld, 2002; Yu, 2006; Quddus *et al.*, 2003).

– **Probabilistic approaches:** an error zone is derived from the error variance associated with the GPS, around each recorded point. If only one road segment intersect the error zone then the point is matched to it, otherwise additional criteria as heading, connectivity and closeness are used to select the best road segment (Ochieng *et al.*, 2003; Zhao, 1997).

– **Advanced approaches:** other concepts such as Kalman filters (Kim *et al.*, 2000), particle filters (Gustafsson *et al.*, 2002), Dempster-Shafer belief theory (El Najjar, Bonnifait, 2005), fuzzy logic (Fu *et al.*, 2004; Quddus *et al.*, 2006) and hidden Markov model (Newson, Krumm, 2009) have also been proposed.

According to (Newson, Krumm, 2009), the hidden Markov model approach correctly identifies up to 99.89% of road segments when the sampling time is less than 30 seconds. Since these performances are better than the ones of other approaches presented in (Quddus *et al.*, 2007), we selected the hidden Markov model for the map-matching of our trajectories. We will now describe it with more details.

2.1. Hidden Markov Model

A Hidden Markov Model (HMM) is a probabilistic model that generates sequences from transitions between the states of a Markov chain (Aggarwal, 2015). The term "hidden" comes from the fact that the states of the chain are not visible to the user but correspond to observations that are visible.

In a Hidden Markov Model used for map-matching, the Markov chain states (nodes) are grouped into layers individually associated with a point on the trajectory to be processed and represent the road segments closest to that point. Each node of a layer has an emission probability quantifying the likelihood that the point associated with the layer belongs to the road segment represented by this node. This probability depends on the measurement error of the GPS used and follows a zero mean Gaussian law (Newson, Krumm, 2009) defined as :

$$p(z_t|r_i) = \frac{1}{\sigma_z\sqrt{2\pi}} e^{-0.5\left(\frac{\|z_t - x_{t,i}\|_{greatcircle}}{\sigma_z}\right)^2} \quad (1)$$

where z_t is the recorded GPS point; r_i is a road segment close enough to z_t ; $x_{t,i}$ is the point associated to z_t on r_i ; $\|z_t - x_{t,i}\|_{greatcircle}$ is the great circle distance between z_t and $x_{t,i}$.

A transition probability is also defined between each pair of nodes coming from two consecutive layers. It indicates the probability of moving from one node to another and therefore from one road segment to another, taking into account the distance between them and the topology of the road network. The transition probability follows an exponential distribution described by (Newson, Krumm, 2009)

$$p(d_t) = \frac{1}{\beta} e^{-\frac{d_t}{\beta}} \quad (2)$$

where d_t is the difference between the great circle distance and the road network distance of two consecutive points of a trajectory, β is an adjustment parameter depending on d_t

Once the model is constructed, it is represented as a network of nodes affected by the emission and transition probabilities on which the Viterbi algorithm is applied to find the best path on the road network corresponding to the inaccurate trajectory recorded on the GPS. Figure 1 illustrates a Hidden Markov Model in which the best path is selected (Goh *et al.*, 2012).

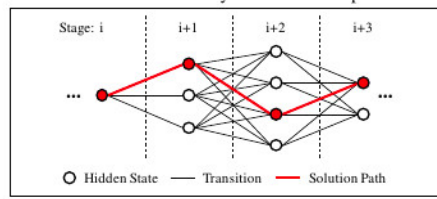


Figure 1. A Hidden Markov Model illustration.

3. Trajectories similarity measures

Since we seek to partition a graph of trajectories into clusters, it is necessary to evaluate the similarity between trajectories. However, the literature most often proposes distance measures between trajectories, which is not adapted to our study. To compensate this lack, we simply convert the distance measures into measurements of similarity using :

$$similarity = \frac{1}{1 + distance} \quad (3)$$

we add 1 to the distance at the denominator, to obtain a normalized similarity between 0 and 1.

The distance measures between trajectories that we have exploited are the Symmetrized Segment-Path Distance (SSPD), Dynamic Time Warping (DTW), Edit Distance with Real Penalty (ERP) and Edit Distance on Real Sequence (EDR). They are among the most widely used in the literature and each has its own unique characteristics. However, none of them is a panacea to the problem of proximity (geographic and / or structural) measurement between trajectories. These distances can be subdivided into two groups: warping distances (DTW, ERP and EDR) that make it possible to compare trajectories of different lengths and take into account the time shifting between trajectories, and shape-based distances (SSPD) that compare the trajectories based on their geometrical characteristics (Besse *et al.*, 2016).

3.1. Dynamic Time Warping (DTW) distance

DTW helps to find the similar patterns between two trajectories of different lengths taking into account the time dimension if necessary. For two trajectories A and B , DTW is computed according to the equation (Yoon *et al.*, 2012):

$$DTW(A, B) = \begin{cases} 0, & \text{if } n = 0 \text{ and } m = 0 \\ \infty, & \text{if } n = 0 \text{ or } m = 0 \\ d(Head(A), Head(B)) + \min \begin{cases} DTW(A, Rest(B)) \\ DTW(Rest(A), B) \\ DTW(Rest(A), Rest(B)) \end{cases} \end{cases} \quad (4)$$

where $d(A,B)$ can be any of the distance functions defined on points and $Head(X)$ (resp. $Rest(X)$) is the first element of X (resp. all but the first element of X). The notations $Head(X)$ and $Rest(X)$ will keep the same meaning in rest of the paper as well as n and m which are the respective lengths of trajectories A and B .

This equation results in a dynamic programming algorithm that evaluates all possible combinations of points between the two trajectories and retains the one where the distance is minimal. It is parameter-free but greedy in calculation time and resources and is also sensitive to noise (Wang *et al.*, 2013).

3.2. Edit Distance with Real Penalty (ERP) / on Real Sequence (EDR)

ERP and EDR are edit distances that can compare trajectories of different sizes with time shifting. An edit distance, also known as the Levenshtein distance between two chains, defines the minimal cost necessary to transform one of the sequences into the other by using a series of transformation operations (Aggarwal, 2015). According to (Zheng, Zhou, 2011) EDR and ERP are defined as :

$$EDR(A, B) = \begin{cases} n & \text{if } m = 0 \\ m & \text{if } n = 0 \\ \min\{EDR(Rest(R), Rest(S)) + subcost, & \text{otherwise} \\ EDR(Rest(R), S) + 1, EDR(R, Rest(S)) + 1\} & \end{cases} \quad (5)$$

where

$$subcost = \begin{cases} 0, & \text{if } d(Head(A), Head(B)) \leq \epsilon \\ 1, & \text{otherwise} \end{cases} \quad (6)$$

$$ERP(A, B) = \begin{cases} \sum_l^n |s_i - g|, & \text{if } m = 0 \\ \sum_l^m |r_i - g|, & \text{if } n = 0 \\ \min \begin{cases} ERP(Rest(A), Rest(B)) + d(Head(A), Head(B)), \\ ERP(Rest(A), B) + d(Head(A), g), & \text{otherwise} \\ ERP(A, Rest(B)) + d(Head(B), g) \end{cases} \end{cases} \quad (7)$$

ERP is based on the L1 norm or Manhattan distance and is a metric while EDR is not. ERP assigns a real penalty to the matched points whereas EDR assigns them a discrete penalty (0 or 1). EDR depends on a matching point parameter which makes it robust to noise while ERP uses a constant called gap to process unpaired points (Zheng, Zhou, 2011). ERP and EDR apply to standardized trajectories (Wang *et al.*, 2013).

3.3. Symmetrized Segment-Path Distance (SSPD)

SSPD is a shape-based distance that use point-to-segment distance. It measures the average of the minimum distances between each point of a trajectory and all segments of a second one and vice versa. It is parameter-free, can compare trajectories of different size and is time insensitive. The definition of SSPD distance is (Besse *et al.*, 2016):

$$SSPD(A, B) = \frac{SPD(A, B) + SPD(A, B)}{2} \quad (8)$$

where

$$SPD(A, B) = \frac{1}{n_A} \sum_{i=1}^{n_A} D_{pt}(p_i^A, B) \quad (9)$$

and

$$D_{pt}(p_i^A, B) = \min_{j \in [0, \dots, n_2-1]} D_{ps}(p_i^A, s_j^B) \quad (10)$$

$D_{ps}(p_i^A, s_j^B)$ is the distance from the point p_i^A to the segment s_j^B .

3.4. Longest Common SubSequence similarity

Besides these distances, we also used the LCSS similarity which is equal to the ratio between the length of the longest common subsequence of two trajectories and the length of the shortest of them. This common subsequence consists of points considered to be matched when the distance between them is less than a predetermined proximity threshold. LCSS is a measure which is robust to noise but does not consider the entire length of the trajectories and is dependent on a parameter. It is calculated as (Zheng, Zhou, 2011):

$$LCSS(A, B) = \begin{cases} 0, & \text{if } n = 0 \text{ or } m = 0 \\ 1 + LCSS(Rest(A), Rest(B)), & \text{if } d(Head(A), Head(B)) \leq \epsilon, \\ & \text{and } |n - m| < \delta \\ \max(LCSS(Rest(A), B), LCSS(A, Rest(B))), & \text{otherwise} \end{cases} \quad (11)$$

Where ϵ is a matching threshold between two points and δ is a parameter used to control how far in time we can go in order to match a given point from one trajectory to a point from another trajectory.

$$Sim_{LCSS}(A, B) = \frac{LCSS(A, B)}{\min(n, m)} \quad (12)$$

4. Trajectories clustering

Using the similarities between the trajectories data, we construct a graph whose nodes are the trajectories themselves and edges bear weights indicating the similarity between two trajectories. This representation of trajectories is inspired by (Khalil El Mahrsi *et al.*, 2015). Once the trajectory graph is obtained, the next step consists in detecting the groups of similar trajectories. It is a well-known problem in graph theory: the detection of communities. The communities in the graphs are considered as groups in which the nodes are densely connected to each other and weakly connected to the nodes of other groups. However, detecting communities and large-scale graphs is a very difficult task classified as Np-hard (Fortunato, 2010). The methods available in the literature for detecting communities are (Fortunato, 2010):

- Traditional methods such as graph partitioning, hierarchical clustering, partition clustering
- Methods based on statistical inference
- Divisive methods
- Spectral methods
- Modularity-based methods
- Method using dynamic algorithms (spin models, random walk)

These methods serve as basis for the design of community detection algorithms, which best performers (Fortunato, 2010) are the Louvain algorithm (Blondel *et al.*, 2008) that makes a direct optimization of the modularity and the InfoMap algorithm (Rosvall *et al.*, 2009) based on map equation and using information theory and random walks. We thus select these two algorithms for the detection of community in our graph of trajectories.

5. Experimentations

5.1. Dataset

The data set used is Taxi Service Trajectory (Moreira-Matias *et al.*, 2013). It contains more than 1000000 trajectories performed by 442 taxis and recorded from 01/07/2013 to 30/06/2014 in the city of Porto (Portugal). Each trajectory is composed of a set of coordinates recorded every 15 seconds from a GPS. For the purposes of our study we will only process 1000 trajectories extracted from this dataset and whose durations are between 5 and 30 minutes. We apply this filter in order to retain only trajectories of intermediate size which are better suited for preprocessing and to prune outliers.

5.2. Trajectories preprocessing

The raw GPS coordinates formatted in JSON (javascript Object Notation) are provided as input to the map matching method using HMM, as described in Section 2, which outputs a trajectory matched to the road network and formatted in GeoJSON format. The trajectories obtained after this first treatment sometimes have loops, that do not exist in the original data, which we eliminate automatically with a script developed for this purpose. We then prune the trajectories points that are not points of interest to reduce trajectories size while retaining their geometric and topological characteristics. All the trajectories resulting from the preprocessing step are recorded in an indexed database.

5.3. Hardware and software used

Our experiments are conducted on an HP Zbook computer, containing an Intel i7-6700HQ octocore processor clocked at 8 * 2.60Ghz with 16Gb of Ram and running under Ubuntu 16.04 LTS. The map used in our study corresponds to the map of the administrative region of the city of Porto in Portugal and was downloaded at <https://www.openstreetmap.org>. We transform it into a routable graph of points of interest whose nodes are most often intersections thanks to the open-source tool Osm2Po. The barefoot open source library (<https://github.com/bmwcarit/barefoot>) written in Java and implementing the hidden Markov model has been used to associate the points of our trajectories with the road segments of the city of Porto. It proposes an "online" map matching, that is, as the data is recorded and an "offline" map matching only possible when all the positions have been recorded. In our case we use the "offline" option. We also use the open-source library "trajectory_distance" written in python (<https://github.com/bguillouet/traj-dist>) to evaluate the distances and similarity described above. The pre-processed trajectories are saved and indexed using the PostgreSQL database management system with its POSTGIS extension.

5.4. Evaluation of the clustering

Obtaining ground truth communities for empirical trajectories is challenging. Since this is a very first comparison of different similarity measures and graph clustering algorithms, we rely on very simple techniques to evaluate the different techniques. Our main indicator of the goodness of a similarity is how well it is to cluster the trajectories. More precisely, community detection algorithm are designed to find clusters of highly connected nodes (or nodes connected with links of high weight) which are connected to each other by few links (or with links of low weight).

Therefore, once the graph of similarity between trajectories is built, we partition it and measure the average intra and inter-cluster weight and also compare it to the average weight on the whole network. If the network can be partitioned well, we should observe that the intra-cluster average weight should be significantly higher than the average weight and the inter-cluster weight.

As indicated in Table 1 and 2 we can observe that EDR and in a lesser extent LCSS are not discriminative enough: intra-cluster, inter-cluster and global average weights are very similar. This is confirmed by the observation of the distribution of the weights (not displayed here by lack of space) which shows for EDR that weights are homogeneously distributed. On the contrary, SSPD and even more DTW and ERP generate similarities much more discriminative: most of them are close to 0 while only few are significant. This is reflected in the ratio of intra/inter cluster average weight that is very high.

The difference between both algorithms is not very significant. In terms of quality, no algorithm gives better results on all datasets.

Table 1. Quality of the partitioning for Louvain and the 5 similarity index: number of clusters obtained, number of links of the weighted graph, global average weight, average weight of intra-cluster links, average weight of inter-cluster links.

Sim.	Nb of clusters	Links	Average similarity	Intra-cluster similarity	inter-cluster similarity
LCSS	8	96586	0.184	0.217	0.123
DTW	19	499500	5.72e-05	0.812	2.68e-05
EDR	5	499500	0.675	0.687	0.672
ERP	17	499500	4.24e-05	0.438	1.39e-07
SSPD	16	499500	4.79e-04	0.401	3.33e-04

Table 2. Quality of the partitioning for Infomap and the 5 similarity index: number of clusters obtained, number of links of the weighted graph, global average weight, average weight of intra-cluster links, average weight of inter-cluster links.

Sim.	Nb of clusters	Links	Average similarity	Intra-cluster similarity	inter-cluster similarity
LCSS	14	96586	0.184	0.319	0.121
DTW	26	499500	5.72e-05	0.629	2.26e-05
EDR	1	499500	0.675	0.675	Nan
ERP	12	499500	4.24e-05	6.9e-03	1.82e-08
SSPD	23	499500	4.79e-04	0.64	4.53e-04

6. Conclusion

In this paper we presented a first comparative study of different similarity measure and graph-based clustering algorithms for network-constrained trajectories. While some trajectories are already mapped to a network, most of them require some pre-processing. It is for instance the case for GPS based trajectories that must be first mapped to a road network from which a set of interest points are extracted to form the nodes of the network. We also presented several state-of-the-art distance or similarity measures that can be used to compare trajectories as well as community detection algorithms. Finally, we presented some comparative results of clustering of trajectories based on these similarities and we have shown that some of them are inadequate for clustering, at least from a network point of view.

Many extensions still possible for this work. First, a deeper analysis of the results has to be performed so as to give precise conclusions on the usefulness of the similarity measures in this context. Next, a more proper validation of the results of the clustering should be done. For instance we might be able to label the different communities that might correspond to commuting trips, travels from the city to the airport, etc. Another solution might be to use an application case, for instance trajectory prediction, to check whether the clusters are meaningful or not. Finally, the last short term perspective is to take time constraints into account in particular to evaluate the similarity of trajectories.

References

- Aggarwal C. C. (2015). *Data mining: the textbook*. Springer.
- Bernstein D., Kornhauser A. (1998). An introduction to map matching for personal navigation assistants.
- Besse P. C., Guillaudet B., Loubes J.-M., Royer F. (2016). Review and perspective for distance-based clustering of vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 11, pp. 3306–3317.
- Blondel V. D., Guillaume J.-L., Lambiotte R., Lefebvre E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, Vol. 2008, No. 10, pp. P10008.
- Chen W., Yu M., Li Z., Chen Y. (2003). Integrated vehicle navigation system for urban applications.
- El Najjar M. E., Bonnifait P. (2005). A road-matching method for precise vehicle localization using belief theory and kalman filtering. *Autonomous Robots*, Vol. 19, No. 2, pp. 173–191.
- Fortunato S. (2010). Community detection in graphs. *Physics reports*, Vol. 486, No. 3, pp. 75–174.
- Fu M., Li J., Wang M. (2004). A hybrid map matching algorithm based on fuzzy comprehensive judgment. In *Intelligent transportation systems, 2004. proceedings. the 7th international ieee conference on*, pp. 613–617.

- Goh C. Y., Dauwels J., Mitrovic N., Asif M. T., Oran A., Jaillet P. (2012). Online map-matching based on hidden markov model for real-time traffic sensing applications. In *Intelligent transportation systems (itsc), 2012 15th international ieee conference on*, pp. 776–781.
- Greenfeld J. S. (2002). Matching gps observations to locations on a digital map. In *81th annual meeting of the transportation research board*, Vol. 1, pp. 164–173.
- Guo D., Liu S., Jin H. (2010). A graph-based approach to vehicle trajectory analysis. *J. Location Based Services*, Vol. 4, No. 3&4, pp. 183–199. Retrieved from <https://doi.org/10.1080/17489725.2010.537449>
- Gustafsson F., Gunnarsson F., Bergman N., Forssell U., Jansson J., Karlsson R. *et al.* (2002). Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, Vol. 50, No. 2, pp. 425–437.
- Hwang J., Kang H., Li K. (2005). Spatio-temporal similarity analysis between trajectories on road networks. In *Perspectives in conceptual modeling, ER 2005 workshops aois, bp-uml, comogis, ecomo, and qois, klagenfurt, austria, october 24-28, 2005, proceedings*, pp. 280–289.
- Khalil El Mahrsi M., Guigourès R., Rossi F., Boullé M. (2015). Co-clustering network-constrained trajectory data. *arXiv preprint arXiv:1511.01281*.
- Kim W., Jee G.-I., Lee J. (2000). Efficient use of digital road map in various positioning for its. In *Position location and navigation symposium, ieee 2000*, pp. 170–176.
- Moreira-Matias L., Gama J., Ferreira M., Mendes-Moreira J., Damas L. (2013). Predicting taxi-passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14, No. 3, pp. 1393–1402.
- Newson P., Krumm J. (2009). Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th acm sigspatial international conference on advances in geographic information systems*, pp. 336–343.
- Ochieng W. Y., Quddus M., Noland R. B. (2003). Map-matching in complex urban road networks. *Revista Brasileira de Cartografia*, Vol. 2, No. 55.
- Phuyal B. P. (2002). Method and use of aggregated dead reckoning sensor and gps data for map matching. In *Ion gps 2002: 15 th international technical meeting of the satellite division of the institute of navigation*.
- Quddus M. A., Noland R. B., Ochieng W. Y. (2006). A high accuracy fuzzy logic based map matching algorithm for road transport. *Journal of Intelligent Transportation Systems*, Vol. 10, No. 3, pp. 103–115.
- Quddus M. A., Ochieng W. Y., Noland R. B. (2007). Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation research part c: Emerging technologies*, Vol. 15, No. 5, pp. 312–328.
- Quddus M. A., Ochieng W. Y., Zhao L., Noland R. B. (2003). A general map matching algorithm for transport telematics applications. *GPS solutions*, Vol. 7, No. 3, pp. 157–167.
- Rosvall M., Axelsson D., Bergstrom C. T. (2009). The map equation. *The European Physical Journal-Special Topics*, Vol. 178, No. 1, pp. 13–23.
- Wang H., Su H., Zheng K., Sadiq S., Zhou X. (2013). An effectiveness study on trajectory similarity measures. In *Proceedings of the twenty-fourth australasian database conference-volume 137*, pp. 13–22.

- White C. E., Bernstein D., Kornhauser A. L. (2000). Some map matching algorithms for personal navigation assistants. *Transportation research part c: emerging technologies*, Vol. 8, No. 1, pp. 91–108.
- Yoon H., Zheng Y., Xie X., Woo W. (2012). Social itinerary recommendation from user-generated digital trails. *Personal and Ubiquitous Computing*, Vol. 16, No. 5, pp. 469–484.
- Yu M. (2006). *Improved positioning of land vehicle in its using digital map and other accessory information*. Unpublished doctoral dissertation, The Hong Kong Polytechnic University.
- Zhao Y. (1997). *Vehicle location and navigation systems*. norwood, ma: Artech house. Inc.
- Zheng Y., Zhou X. (2011). *Computing with spatial trajectories*. Springer Science & Business Media.