

# Spatio-temporal clustering methods

Matej Senožetnik, Luka Bradeško, Blaž Kažič, Dunja Mladenić, Tine Šubic  
Jožef Stefan Institute  
and  
Jožef Stefan International Postgraduate School  
Jamova cesta 39  
1000 Ljubljana, Slovenia

## ABSTRACT

Tracking a person, an animal, or a vehicle generates a vast amount of spatio-temporal data, that has to be processed and analyzed differently from ordinary data generally used in knowledge discovery. This paper presents existing spatio-temporal clustering algorithms, suitable for such data and compares their running time, noise sensitivity, quality of results and the ability to discover clusters according to non-spatial, spatial and temporal values of the objects.

## Keywords

clustering, spatial-temporal data, density-based algorithms

## 1. INTRODUCTION

Due to emerging field of ICT and rapid development of sensor technologies, a lot of spatio-temporal data has been collected in the past few years. By processing and enriching raw spatio-temporal data we aim at extracting semantic information, which is a basic requirement for the comprehension and later usage of this data. Normally, the very first step of this process is clustering raw GPS coordinates into more distinct groups of points, which already have some semantics, such as whether the points belong to trajectory or stationary point (i.e., stay point).

This paper aims to investigate different methods, used for clustering spatio-temporal data, generated by mobile phones, by collecting timestamped GPS coordinates of the phones' location. By clustering the collected coordinates, we obtain so called stay points (also referred to as points of interest or stop points [15]), which are the points in space where a moving object has stayed within a certain distance threshold for a longer period of time [16]. When the stay points are calculated, we can process the data further, to calculate the most frequently visited locations (i.e., frequent locations), which is the fundamental building block for further advanced analytics use-cases, such as next location prediction [6].

In general, clustering methods are separated into following categories:

- **Partitioning methods** - classify data into  $k$  groups or partitions,
- **Hierarchical methods** - hierarchically decompose given datasets,
- **Density-based methods** - are based on cluster density, where clusters stop growing when neighbourhood density stops exceeding a given threshold,
- **Model-based methods** - definition copied from [1]: "In this method, a model is hypothesized for each cluster to find the best fit of data for a given model. This method locates the clusters by clustering the density function. It reflects spatial distribution of the data points."

The most appropriate methods for clustering spatio-temporal data are density-based methods as they regard clusters as dense regions of objects in a data space that are separated by regions of low density [7]. Thus we will focus on density based methods and their application to cluster GPS coordinates.

## 2. DENSITY-BASED ALGORITHMS

Table 1 shows some of the more common density-based algorithms, used for the detection of stay points. For a successful detection of stay points, it is important that the clustering algorithm uses temporal information alongside bare spatial data. As indicated in Table 1, most of the algorithms do make use of it, except for DBSCAN. Quality of the algorithm also depends on noise sensitivity (the less sensitivity the algorithm has, the better it is), as cellphones' GPS coordinates are frequently noisy due to connection glitches (for instance, when moving through forests, staying inside buildings or in bad weather conditions). Some algorithms return only clusters such as DBSCAN, ST-DBSCAN and OPTICS, but CB-SMoT, SMoT and SPD return a stay point (also referred to as stops) or a path (also referred to as moves).

DBSCAN has an overall average running time  $O(n \log n)$ , and the worst case run time complexity is  $O(n^2)$ . Running time depends on parameter choice and version of implementation. OPTICS has similar time complexity but it is 1.6 seconds slower than DBSCAN. ST-DBSCAN has the same running time as DBSCAN.

Algorithm name	Spatio temporal	Noise sensitivity	Returning stay points/path
DBSCAN	✗	✗	✗
ST-DBSCAN	✓	✗	✗
SMoT	✓	✓	✓
CB-SMoT	✓	✗	✓
SPD	✓	✓	✓
OPTICS	✓	✗	✗

Table 1: The most common density-based algorithms, used for the detection of stay points [14].

## 2.1 Density Based Spatial Clustering of Application with Noise (DBSCAN)

DBSCAN [5] is a density-based clustering algorithm which identifies arbitrary-shaped objects and detects noise in a given dataset. The algorithm starts with the first point in the dataset and detects all neighboring points within a given distance. If the total number of these neighboring points exceeds a certain threshold, all of them have to be treated as part of a new cluster. The algorithm then iteratively collects the neighboring points within a given distance of the core points. The process is repeated until all of the points have been processed.

DBSCAN’s advantages are that it robustly detects outliers, only needs two parameters (Eps and MinPts), is appropriate for large datasets and data input order does not interfere with the results [12].

Numerous research studies have extended DBSCAN, such as in the example of GDBSCAN [13], which is a generalization of the original DBSCAN. GDBSCAN can cluster point objects as well as spatially extended objects. Another one of these extended algorithms is DJ-Cluster [17], used for discovering personal gazetteers. It attaches semantic information to clusters and requires a list of points of interest. Also extension is ST-DBSCAN which is described below.

ST-DBSCAN [4] is another algorithm that is based on DBSCAN. It is making use of its ability to discover clusters with various shapes, while improving some of the weak points of the original algorithm. It adds temporal data to the clustering results, identifies noisy objects if there are various densities of the input data, and more accurately differentiates adjacent clusters.

## 2.2 Stops and Moves of Trajectory (SMoT)

SMoT [2] algorithm divides data into two sets called *moves* and *stops*. The easiest way to understand how SMoT works is by reviewing the example shown in Figure 1. There are three potential stop candidates with geometries  $R_{C_1}$ ,  $R_{C_2}$  and  $R_{C_3}$  and with information about minimum time duration for each of them. From the figure, we can observe that the point  $p_0$  is not inside any of these geometries, therefore it is a candidate for *move* dataset. The next few points are inside the first stop candidate ( $R_{C_1}$ ) and also exceed the minimum time duration which is specified for every geometry. In candidate  $R_{C_2}$ , the point duration does not exceed minimum threshold, therefore  $R_{C_2}$  is not a stop.

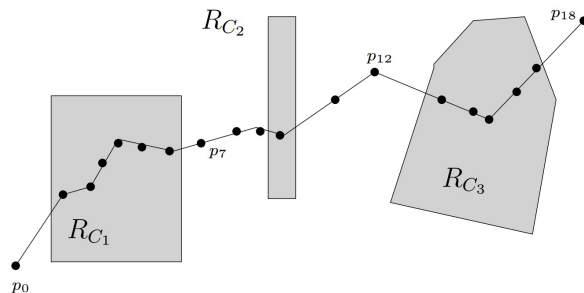


Figure 1: Example of trajectory points with three possible candidate stops [2].

## 2.3 Clustering-Based Stops and Moves of Trajectory (CB-SMoT)

CB-SMoT [11] algorithm is an alternative to the algorithm SMoT and addresses one of its main drawbacks - the incapability to detect stay points that are not predefined by user. It uses clustering methods to automatically detect stay points. The idea behind this method is, that when we move around points of interest (such as museums, monuments, night-clubs, etc.), we move slower than when we are traveling from one place to another. In the first steps, the potential stops (the slower parts of a trajectory) are identified using the variation of the DBSCAN algorithm which considers one-dimensional trajectories and their speed. In the second step, the algorithm identifies the location of the potential stops (clusters) which were found in the first step. The authors report [11] that their algorithm discovers less incorrect stops, compared to SMoT algorithm.

## 2.4 Stay Point Detection (SPD)

The SPD [9] algorithm works by detecting whether the observed entity has spent more than 30 minutes within a radius of 100 meters. If this happens, the region is detected as a stay point. Both threshold time and distance parameters are adjustable and are chosen depending on the use-case and accuracy of raw data.

The main advantages of SPD is its need for predefined structures. It is not computationally demanding, but it is sensitive to the accuracy of data usually generated by GPS receiver. Namely, for different accuracies of GPS it returns slightly different positions for the same location. It is also sensitive to noise, but this can be partially reduced by adjusting the parameters of the algorithm.

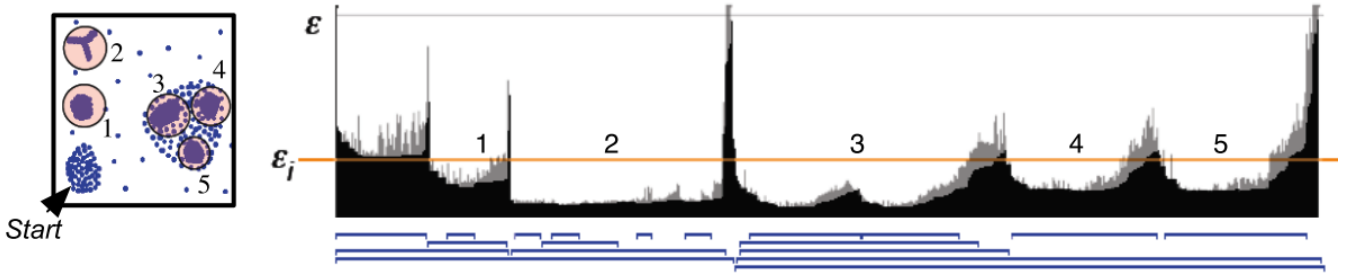


Figure 2: OPTICS result is called reachability plot [10].

## 2.5 Ordering Points to Identify the Clustering Structure (OPTICS)

OPTICS [3] is an algorithm which is used for finding density-based clusters in spatio-temporal data. Though it works in a similar way as DBSCAN, OPTICS improves on DBSCAN’s biggest weakness, the failure to detect clusters when density of the data varies.

In Figure 2, we can observe that OPTICS algorithm generates an easily visualized ordering of points, which can be used to extract partitions and hierarchical clusters [10]. Zheng et al. [16] used OPTICS for clustering stay points. In their article, they present the idea that through the use of a statistically significant number of user-generated GPS trajectories, the correlation between personal geographical locations implicit in a person’s location history enables the provision of valuable services, such as personalized recommendations and target specific sales promotions.

## 2.6 Other algorithms

There exist many other different approaches on dealing with spatial and spatio-temporal data. Some of these are developed as extensions (for example, DB-SMoT [12]) to well known algorithms, while others take an entirely new approach (TRACCLUS [8]). It is important to note that other data can also be used alongside coordinates and timestamps.

## 3. DISCUSSION

The goal of this discussion is to find an algorithm capable of clustering user’s raw historical data of locations, as tracked by a mobile phone. Referring back to Table 1, we want an algorithm which has the following properties:

- is able to cluster spatio-temporal data;
- is noise insensitive;
- returns stay points and a paths.

As we had already stated in the beginning of this paper, besides spatial data, temporal information is one of the most important additional information for stay point detection algorithms, which enables better stay point detection performance and empowers additional time related capabilities, such as detecting time spent at each stay point. By using an algorithm that clusters data only by spatial information, we lose part of the useful information (in our case, the time

spent on a location), as well as the order of stay points on the timeline, thus making us unable to do further analysis, such as future location predictions and plotting frequency graphs. Due to this shortcoming and also different cluster densities, algorithms such as DBSCAN, or its improved version DJ-cluster, are not appropriate for our use case.

In [14], Sander et al. exposed that the problem of DB-SMoT algorithm is that the quantile function requires a priori knowledge of the proportion between points inside potential stops and total points in the dataset. This proportion varies among datasets since users sometimes spend a whole day inside a stop (i.e., the proportion is one), while on different occasions they might be visiting more than ten stops (i.e., the proportion is much smaller than one). Due to this, we need an online algorithm that can function without prior knowledge of stay point to path ratio. The SMoT algorithm uses predefined regions which can be a problem if we want to detect some stay points outside of those zones (such as outdoors). Algorithm SPD, SMoT and CB-SMoT are all sensitive to noise, but have the advantage of returning stay points ordered by timestamps. With other algorithms, such as DBSCAN and OPTICS, we need to find the right sequence ourselves independently of the algorithm. Given our requirements, we propose CB-SMoT, SMoT and SPD as the most suitable for clustering spatio-temporal data collected from a mobile phone.

An efficient algorithm must be insensitive to noise. Currently, GPS has reception problems in narrow valleys, forests or inside buildings, so it’s important to use different data sources (for example, Wi-Fi networks, activity recognition) for accurate stay point detection. Modern mobile phones nowadays also provide rudimentary activity recognition, which we can use to compare our own results against. If the activity recognition system is not detecting movement (the person is standing still) and our collected GPS coordinates are consistently far apart, we may be facing a problem with GPS accuracy. In such cases, additional data sources should be considered for more accurate measurement.

To the best of our knowledge, up to this date SPD algorithm is considered as the best solution for detection of stay points, but suffers from detecting false stay points and paths. This problem can be alleviated by running multiple iterations of the algorithm on resulting dataset. This is already outside of the scope of this paper and will be described in the separate paper which is currently under preparation.

## 4. ACKNOWLEDGMENT

This work was supported by the Slovenian Research Agency and the ICT program of the EC under project OPTIMUM (H2020-MG-636160).

## 5. REFERENCES

- [1] Clustering analysis. [http://www.tutorialspoint.com/data\\_mining/dm\\_cluster\\_analysis.htm](http://www.tutorialspoint.com/data_mining/dm_cluster_analysis.htm).
- [2] Luis Otavio Alvares, Vania Bogorny, Bart Kuijpers, Jose Antonio Fernandes de Macedo, Bart Moelans, and Alejandro Vaisman. A model for enriching trajectories with semantic geographical information. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems, GIS '07*, pages 22:1–22:8, New York, NY, USA, 2007. ACM.
- [3] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod Record*, pages 49–60, 1999.
- [4] Derya Birant and Alp Kut. ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007.
- [5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.
- [6] Sébastien Gams, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. Next place prediction using mobility markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, page 3. ACM, 2012.
- [7] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17:107–145, 2001.
- [8] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: A partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, SIGMOD '07*, pages 593–604, New York, NY, USA, 2007. ACM.
- [9] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '08*, pages 34:1–34:10, New York, NY, USA, 2008. ACM.
- [10] Chris Mueller. Data Preparation. 2005.
- [11] Andrey Tietbohl Palma, Vania Bogorny, Bart Kuijpers, and Luis Otavio Alvares. A clustering-based approach for discovering interesting places in trajectories. In *Proceedings of the 2008 ACM Symposium on Applied Computing, SAC '08*, pages 863–868, New York, NY, USA, 2008. ACM.
- [12] Parya Pasha and Zadeh Monajjemi. a Clustering-Based Approach for Enriching Trajectories With Semantic Information Using Vgi Sources a Clustering-Based Approach for Enriching Trajectories With Semantic Information Using Vgi Sources. 2013.
- [13] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Min. Knowl. Discov.*, 2(2):169–194, June 1998.
- [14] Khoa a Tran, Sean J Barbeau, and Miguel a Labrador. Bacchelor: Automatic Identification of Points of Interest in Global Navigation Satellite System Data : A Spatial Temporal Approach Categories and Subject Descriptors. (January):33–42, 2013.
- [15] Y. Zheng and X. Xie. Learning location correlation from gps trajectories. In *2010 Eleventh International Conference on Mobile Data Management*, pages 27–32, May 2010.
- [16] Yu Zheng. Trajectory Data Mining: An Overview. *ACM Trans. On Intelligent Systems and Technology*, 6(3):1–41, 2015.
- [17] Changqing Zhou, Dan Frankowski, Pamela Ludford, Shashi Shekhar, and Loren Terveen. Discovering personal gazetteers: An interactive clustering approach. In *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems, GIS '04*, pages 266–273, New York, NY, USA, 2004. ACM.