Spatio-temporal clustering methods classification

Hadi Fanaee Tork
info@fanaee.com

Abstract. Nowadays, a vast amount of spatio-temporal data are being generated by devices like cell phones, GPS and remote sensing devices and therefore discovering interesting patterns in such data became an interesting topics for researchers. One of these topics has been spatio-temporal clustering which is a novel sub field of data mining and Recent researches in this area has focused on new methods and ways which are adapting previous methods and solutions to the new problem. In this paper we first define what the spatio-temporal data is and what different it has with other types of data. Then try to classify the clustering methods and done works in this area based on the proposed solutions. classification has been made based on this fact that how these works import and adapt temporal concept in their solutions.

Keywords: Spatial Clustering, Spatio-temporal Clustering, Data Mining, GIS

1 Spatio-Temporal Data

Before bringing a clear definition about spatio-temporal clustering, we should explain more about the nature of spatial and spatio-temporal data and its difference with classical data. As it can be seen on figure 1 which shows a sample of classical data, each points is represented by its x and y values in a 2-D space and it doesn't show anything about the spatial or temporal situation of the points.

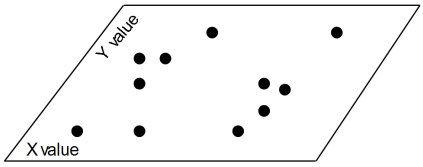

Figure 1 – a Simplified classical data in 2-D space

In Spatial data, the data item is representing by its spatial location (usually on earth) and it doesn't provide any information about other features of that item. Figure 2 shows a sample of spatial data. In this case we don't also have any temporal

information for each data item which cause spatial data to be different from spatio-temporal data.

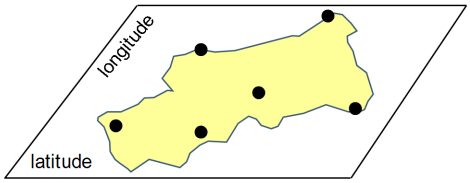

Figure 2 – A sample of spatial data

Spatio-Temporal data is more complicated, because time factor can be involved in different ways. By the way if we had a temporal information for each data item we are facing with spatio-temporal data and not spatial data. We have three types of spatio-temporal data:

Events : if there is no correlation between data items and data set doesn't include any identification for each data item or at least its not important for us. A sample of such data set is presented in table 1. As it can be seen there are 18 objects that for each object we have both spatial and temporal data. So for example <X6,Y7,2> implies on object 6 and it shows that object 6 has occurred in time = 2 and its Longitude and Latitude are X6 and Y7 respectively.

Table 1. A sample data set of Events data

Longitude	*Latitude*	*Time*
X1	Y1	1
.	.	1
.	.	
.	.	
X6	Y6	1
X7	Y7	2
.	.	2
.	.	
X12	Y12	3
.	.	3
.	.	
X18	Y18	3

Geo-Referenced data items : such data items are objects that in addition to their spatial and temporal position, a non-spatial value related to them is added to data item. A sample of such data type is presented on Table 2. for example each data items in this case can be a weather station location and corresponding temperature value at the different time sequences. For instance <X6,Y6,2,V62> implies on object number 6 with Longitude X6 and Latitude Y6 and 2 shows that this object has occurred In time=2 and V62 is for example temperature

value related to this object. Likewise V61 is the temperature value related to object 6 but at time=1.

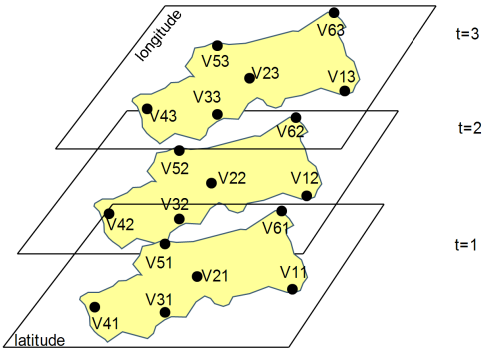

Figure 3 – Geo-Referenced data items

Table 2. A sample data set of Geo-Referenced data items

Longitude	*Latitude*	*Time*	*Value*
X1	Y1	1	V11
X2	Y2	1	V12
.	.	1	.
.	.		.
.	.		.
X6	Y6	1	V16
X1	Y1	2	V12
X2	Y2	2	V22
.	.	2	.
.	.		.
.	.		.
X6	Y6	2	V62
X1	Y1	3	V13
X2	Y2	3	V23
.	.	3	.
.	.		.
.	.		.
X6	Y6	3	V63

Moving data items: In such data sets, data items are moving and are not static. For identification of data items entity they should have an ID to be able to trace their movement during the time. For example figure 4 shows some moving points data set during the time between t=1 and t=3. As it can be seen from figure, green object is moving to left, orange object has not moved at least from t=1 to t=3, blue and yellow objects are being closer together and white object is moving to right. Also we have red object in t=1 and we don't have this object in the next time sequences. It means we have not had any information about the red object location

at that periods. It can be due to lack of GPS data at that time (e.g. user has entered to the inside of a house). As we observed, the important thing that enabled us to identify the moving behavior of objects was their color or better say their identification parameter. So in such data sets we have all ID, spatial location and temporal information as records. Table 3 shows a sample of such data sets.

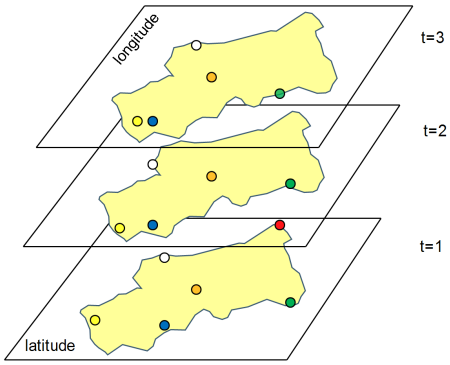

Figure 4 – Moving data items

Table 3. A sample data set of Moving data items

ID	*Longitude*	*Latitude*	*Time*
GREEN	X1	Y1	1
GREEN	X2	Y2	2
GREEN	X3	Y3	3
RED	X4	Y4	1
RED	---	---	2
RED	---	---	3
ORANGE	X5	Y5	1
ORANGE	X6	Y6	2
ORANGE	X7	Y7	3
BLUE	X8	X8	1
BLUE	X9	Y9	2
BLUE	X10	Y10	3
WHITE	X11	Y11	1
WHITE	X12	Y12	2
WHITE	X13	Y13	3
YELLOW	X14	Y14	1
YELLOW	X15	Y15	2
YELLOW	X16	Y16	3

2 Spatial data Clustering

Spatial data clustering is not a new task and we had already the same concept in classical data clustering. The only difference is the difference between the nature of input values. In classical data as it has been shown in figure 1, we have values which can be presented in multi-dimensional vectors and therefore <x,y> represents two distinctive values in a 2-d space. With this definition spatial clustering can be simplified as a vector with two values like x,y but this time instead of values x and y, the longitude and latitude a object can be replaced. With this assumption the spatial clustering problem is exactly like clustering of 2-d vectors.

This clustering also can be done via density-based methods or distance-based methods. distance-based methods have two weakness which leads to be not suitable for spatial data clustering, first they need a number of clusters as an input and second they allocate all objects to the clusters and never identify noises. By the way there are some related works like[13] which firstly transform spatial or spatio-temporal data to same length multi-dimensional vectors and then apply a generic clustering algorithm like k-mean on the data.

Density-based clustering of spatial data mostly is based on two well-known density-based algorithms DBSCAN (Density-Based Spatial Clustering of Applications with Noise) and OPTICS(Ordering Points To Identify the Clustering Structure). In both of these algorithms there is a density function which compute the distance of objects in order to allocate them into the clusters. Two input parameters of these algorithms are distance threshold and minimum number of neighbors which can make objects a member of cluster or a noise. The most important property of density function that distinguish them from other algorithms is flexibility of that distance function so that we can configure this distance function according to our specific purpose. Despite of similarity in login between OPTICS and DBSCAN the most important difference between them is related to the order of visiting of the objects in data set. Contrary to DBSCAN, OPTICS visit the objects based on their ordered distances from the visited objects, thus its no such sensitive to the input parameters and also have good tolerance to the noises. Also DBSCAN can not recognize hierarchical clusters and does not perform well in case the clusters have different densities, while OPTICS can do both.

3 Spatio-temporal data Clustering

As a matter of fact, spatio-temporal data clustering is not much difference with spatial data clustering. All the related works are trying to import time concept in the data or algorithms as a threshold or by the distance function or even they transform the spatio-temporal clustering problem to a multi-sequence spatial data clustering. So with this explanation, we can say there are the following possible strategies:

- Different distance functions

- Importing time to the spatial data
- Transform spatio-temporal data to the new objects
- Configure the algorithm
- Progressive clustering
- Performing clustering task on each time sequences
- Thresholds-based clustering
- Spatio-temporal pattern discovery

In the remained part of the paper we explain more about the above strategies.

3.1 Different distance functions

In this strategy, different distance functions are employed according to the specific goal of analysis. In a part of analysis we might use one specific distance function and in other part we might use another one. This strategy can be applied to two different ways with respect to the type of the given data :

Events : When we are dealing with events data (like data set of table 1), we have some events like crime points which are occurred in a specific times. In this case, we like to apply a density-based clustering algorithm like DBSCAN on data set.s However, in the stage of search for neighbors of a point in DBSCAN we need a function that could calculate the distance between the give point and other points and then retrieve the neighbors of the given point. This distance function can be selected according to the goal of analysis. We have three possible scenario as follows:

Spatial distance function: in this case, we ignore the temporal part of data items and apply for example DBSCAN on all data items. In this case our problem is changed to spatial data clustering. The goal could be finding the regions of the city that have the most crime activates. But yet we don't know in which periods of time these crimes are occurring more.

Temporal distance function : in this case, we ignore the spatial part of the data items and apply clustering on all data items. Our problem will be changed to temporal clustering. As the final result we can understand in which periods of the year, month, week or time , depending on the type of temporal part(year, month, week, stime,…) the crime amounts are the high or low.

Spatio-Temporal distance function : Some works like [13],[14] benefit from this method and created modified algorithm namely ST-DBSCAN while other works like[4] a spatio-temporal distance function with spatial and time threshold is using in order to discover the interdiction process over the three years. In both ways the goal is to discover the spatio-temporal behavior of events. For example we might like to discover the spatio-temporal regions of crime spots. More precisely as the results, we may understand that downtown of the city between hours 19 to 23 and a specific

zone of city between 12 to 13 are crime spots. So these spatio-temporal regions can be reported to the police to have more patrol there during that time periods. In order to cluster such data we need a little modification in one stage of algorithm and that is when algorithm is going to calculate the distance between points. As well as spatial threshold like 5km We need to add a time threshold like 1.5h to filter the points that has not enough distance to the given point. In fact firstly when for example DBSCAN searches for objects that have lower distance than 5km to the given object. Then before starting the counts of the retrieved objects we add another filter step. In this new filter step, we check that whether these objects have lower time distance of 1.5h to the given object or not. If they didn't satisfy the threshold they will be removed from the retrieved list. Then we count the final retrieved list after filtering. If the number of objects was greater than MinPts (minimum number of points required to form a cluster) a cluster is created and that point is a core point. Another section of the algorithm is exactly like normal one.

Trajectories or Moving points: In this case, we are dealing with moving points data usually called trajectories. As we already discussed, the most important different between moving points data and events data is that in events data, objects are not moving and they are constant points without any direct correlation to other points. But In terms of moving points always each point is identified with its ID and it has direct relation to other points, so that some points generate a trajectory. So all the related points of this trajectory have same ID and they are connected together. We can not see such relations in events data. So the nature of these two data is different from perspective of analysis. However as like as event data we also are able to use different distance function according to the goal of analysis or in progressive clustering according to the stage of analysis. in[3] some distance functions like similar routes , similar destinations, similar source, similar route and destinations, similar directions are mentioned and are employed in time of the analysis. For example in similar destination or similar source as depicted in figure 5-a problem is transferred to a single points clustering. Because we ignore the whole trajectories points and we just take their start or end points. In case of similar routes as shown in figure 5-c we need to compare whole trajectories. So the problem of comparison is comparing of two polylines. The determination of selected point per se is a complicated problem as will be discussed more in section 3-3. However a direct method for such comparison is computing the Euclidian distance between the corresponding points.

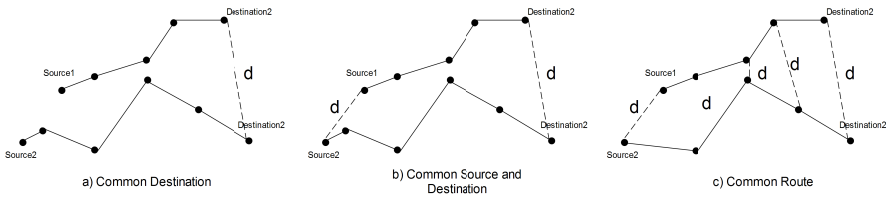

Figure 5 – Similarity calculation of two trajectories based on different goals a) common destinations b) common sources and destinations c) common routes

Concerning the distance functions like similar source and destination (figure 5-b) we need to discover some groups of trajectories which they are coming from the same root and go to the same destination. These trajectories might be a short direct way from the source to the destination or ones which pass zigzag path to reach to the destination. In all such distance functions we need to modify our density based algorithm to adapt the clustering task with our desired goal. For example in [7] a new algorithm namely TRJ-DBSCAN is proposed to find trajectories within *e* of given object at a fixed time *t* which employs route similarity distance function.

3.2 Importing time to data and define a new threshold

In this strategy, a new time dimension is added to the 2-d vector of spatial data [15] and then problem will be transformed to clustering of a 3-d vectors of <x,y,t> which can be performed by both distance-based (e.g. k-mean) and density based algorithms(e.g. DBSCAN or OPTICS). In both algorithms a standard distance measure such as the Euclidean distance (equation 1) will be used as distance measure criteria between two objects <x1,y1,t1> and <x2,y2,t2>.

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2 + (t2 - t1)^2} \quad (1)$$

3.3 Transform spatio-temporal data to the new objects

In this strategy which in general is using for trajectories, we make a new object from some spatio-temporal data items so that in nature has the both spatial and temporal concept inside of itself. Then we apply clustering algorithms on the new data objects[1,2,3,4,5,6,7,8,10,11]. Also In this case for comparing the new objects we need a new distance function that could be as above a Euclidean distance which is able to compute the distance between two new spatio-temporal objects. For instance, regarding the green and blue points on figure 4 and data items in table 3, according to the spatial location visited by green point during the time between t=1 to t=3 we can transform all data items related to this point to a single object like Trj_Green: (X1,Y1)->(X2,Y2)->(X3,Y3) and likewise for the blue point we have : Trj_Blue: (X8,Y8)->(X9,Y9)->(X10,Y10). So with a Euclidean distance function we can compute the similarity of these two new spatio-temporal objects. sometimes comparing two trajectories by just comparing their spatial locations is not reasonable because it neglects the temporal aspect. Some works like [16] consider the properties of moving objects in road network space and define temporal similarity as well as spatio-temporal similarity between trajectories based on POI (Points of Interest) and TOI (Times of Interest) on road networks. Comparing trajectories is not always easy

as mentioned above, in some circumstances it's a difficult task especially when in a real-world problems two trajectories length may completely be different. Also if we consider each point of trajectories, it would be an expensive task for long trajectories. Some trajectory simplification techniques like Douglas-Peucker(DP) algorithm[7,11,17] are used to reduce the trajectory comparison computation costs. The goal of trajectory simplification is transforming a polylines to another polylines with less points. In [7] another two algorithm DP+ and DP* is also presented for increasing of efficiency. Goal of all of these methods is having less points while we are comparing trajectories and therefore less computation.

3.4 Progressive Clustering

Motivation of all works done here [1,2,3,4,5] is doing filtering on the data set to first reduce the computing costs and second get better results according to the specific goals. In some works like [2,3,4] , multiple distance functions and different input parameters are employed in a progressive way so that according to the goal of analytics, suitable inputs and distance function are being used and in some other work like [1] authors are trying to filter the input data with focusing on time periods which cause the clustering result to be same as using whole data set. For instance if the goal is understanding the traffic patterns in a city, instead of applying clustering to whole data set we can first identify which time periods give us the result we are looking for and then filter the data set according to that time periods and then perform clustering. In this case first we are facing with a smaller data set and second the result will be more meaningful because for example the traffic patters in city in weekends is completely different with weekdays and by mixing the both data items the result may not what we like to know. Also in [5] a heuristic method is used for clustering of very large spatio-temporal data set. The idea behind that is this fact that dense regions in data set remain dense also in a sample subset. So if we were able to discover the fundamental frame we can classify other objects to the discovered clusters more easier and cheaper. In order to do this, firstly a proper subset is selected and a density based clustering like OPTICS is applied to that subset, then an analytics do modification and revision on the results, then build a classifier and employ that classifier to allocate each new objects to the obtained clusters.

3.5 Performing clustering task on each time sequences

Sometimes we perform clustering on each time sequences separated, for example in [4] landing events are clustered irrespective of time distance threshold for three years of 2005,2006 and 2007 and then clustering results of these years is useful to find out how pattern of landings has been changed over these years. Also this strategy is employed in the moving clustering problem[6]. Moving clusters are group of objects which enter or leave the cluster during some time intervals but having the portion of common objects higher than predefined threshold. For discovering of moving clusters it needs to perform clustering on each time sequences.

3.6 Thresholds-based clustering

In this strategy, generally spatial and temporal threshold is used for grouping objects and in general no clustering algorithm is involved. Such strategy is used in [9] for discovering of important places from trajectories, every new location is compared to the previous location. If the distance is less than a threshold, the new location is added to the previously created cluster. Otherwise, the new candidate cluster is created with the new location. The candidate cluster becomes a cluster of important places when the time difference between first point in a cluster and the last point is greater than the threshold. In another work like [8], they benefit from this strategy for discovering of flocks. Flocks are group of trajectories that stay together within a specific disk size for the duration of a given time. For example if a disk includes two trajectories in three consecutive sequences it can be a flock with thresholds M=2 and K=3.

3.7 Spatio-temporal pattern discovery

Spatio-temporal patterns which in general is not a clustering problem its more a pattern discovery problem, describe a general spatio-temporal behavior of a group of objects[10,11]. For example as a result of trajectory pattern mining on travelers trajectories we might have patterns like $Sation \xrightarrow{7\,\min} Downtown \xrightarrow{15\,\min} beach$ and $Sation \xrightarrow{2\,\min} Bridge \xrightarrow{8\,\min} University$ so that the first one implies the travel pattern of tourists and second one could be a travel pattern of students. In order to discover such patterns, four steps is needed[10], first a set of input trajectories over given grid and using spatial neighborhood or radius to allocate the members. Second computing the point of interests from trajectories points by a density based clustering algorithm and then mining the frequent patterns.

4 Conclusion

Several methods and techniques introduced during the paper in terms of spatio-temporal data clustering, however summary of all methods can be summarized in three ways, first those methods who try to transform spatio-temporal data to a condition which can be used in classical clustering algorithms mostly density-based, second trying to change a bit in classic and spatial-suited algorithms to enable algorithm handle the new data type. The third method are related to event type data which both change in algorithm and data can is not sufficient to solve the problem and should be seen separately and is a collection of tasks in progressive ways.

References

1. Nanni M, Pedreschi D (2006) Time-focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems 27(3):267–289
2. Rinzivillo S, Pedreschi D, Nanni M, Giannotti F, Andrienko N, Andrienko G (2008) Visually driven analysis of movement data by progressive clustering. Information Visualization 7(3):225–239
3. Andrienko G, Andrienko N (2008) Spatio-temporal aggregation for visual analysis of movements. In: Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST 2008), IEEE Computer Society Press, pp 51–58
4. Andrienko G, Andrienko N (2009) Interactive cluster analysis of diverse types of spatiotemporal data. ACM SIGKDD Explorations
5. Andrienko G, Andrienko N, Rinzivillo S, NanniM, Pedreschi D, Giannotti F (2009) Interactive Visual Clustering of Large Collections of Trajectories. VAST 2009
6. Kalnis P, Mamoulis N, Bakiras S (2005) On discovering moving clusters in spatio- temporal data. Advances in Spatial and Temporal Databases pp 364–381
7. Jeung H, Yiu ML, Zhou X, Jensen CS, Shen HT (2008) Discovery of convoys in trajectory databases. Proc VLDB Endow 1(1):1068–1080
8. Vieira MR, Bakalov P, Tsotras VJ (2009) On-line discovery of ?ock patterns in spatio-temporal data. In: GIS ’09: Proceedings of the 17th ACMSIGSPATIAL In- ternational Conference on Advances in Geographic Information Systems, ACM, New York, NY, USA, pp 286–295
9. Kang JH,WelbourneW, Stewart B, Borriello G (2004) Extracting places from traces of locations. In: WMASH ’04: Proceedings of the 2nd ACM international work- shop on Wireless mobile applications and services on WLAN hotspots, ACM, New York, NY, USA, pp 110–118
10. Giannotti F, Nanni M, Pinelli F, Pedreschi D (2007) Trajectory pattern mining. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, p 339
11. Kang J, Yong HS (2009) Mining Trajectory Patterns by Incorporating Tempo- ral Properties. Proceedings of the 1st International Conference on Emerging Databases
12. Reades J, Calabrese F, Sevtsuk A, Ratti C (2007) Cellular census: Explorations in urban data collection. IEEE Pervasive Computing 6(3):30–38
13. Birant D, Kut A (2007) St-dbscan: An algorithm for clustering spatial-temporal data. Data Knowl Eng 60(1):208–221
14. Wang M, Wang A, Li A (2006) Mining Spatial-temporal Clusters from Geo- databases. Lecture Notes in Computer Science 4093:263
15. Roberto Trasarti , Mastering the Spatio-Temporal Knowledge Discovery Process, PhD Thesis , University of Pisa Department of Computer Science.
16. J. Hwang, H. Kang, and K. Li. Searching for similar trajectories on road networks using spatio-temporal similarity. In Proc. of the East-European Conference on Advances in Databases and Information Systems, pp. 282--295, 2006.
17. D. Douglas and T. Peucker. Algorithms for the reduction of the number of points required to represent a line or its character. The American Cartographer, 10(42):112–123, 1973