Review and compare clustering algorithms for navigation data analysis tasks

Ponomareva A.V., Meyta R.V.
Institute of Cybernetics
National Research Tomsk Polytechnic University
Tomsk, Russia
avp35@tpu.ru

Abstract

This paper presents a study of the possibility of application of cluster analysis methods to the data sets from navigation receivers. The navigation data from the moving objects have a number of features, it makes the application to them of some class of clustering algorithms impossible. Such features include, in particular, the prevalence of clusters of complex shape different from circle.

Keywords

Navigation, cluster analysis, GPS, k-means, c-means, DBSCAN, Data Mining.

I. INTRODUCTION

ata Mining methods are commonly used today to analyze large amounts of data in order to reveal the hidden information and the logical connections in them. The choice of the method is based on the properties of the processed data and the solved problem. One of the types of the problems solved by Data Mining methods are cluster analysis problems. Cluster analysis is successfully used for the research of spatial data sets, for example, the navigation data from satellite systems GPS / GLONASS. Such data are received as a radio signal. On the basis of signal from several visible satellites navigation receiver calculates its location and current time on the Earth. This information is provided by the receiver as a sequence of NMEA-strings. Cluster analysis of navigation data sets can be used to solve a wide range of problems, mainly related to the transport network control. However, as in case of working with the other types of input data, navigation data require the preliminary filtration to improve the results of the processing of the used algorithm by cutting erroneous or distorted input data in advance.

II. STUDIED DATA

NMEA 0183 (National Marine Electronics Association) – is a text based protocol, which is used for the aim of establishing communication between navigation equipment. This protocol describes a large list of different messages, from which we can highlight the twenty which are actively used in a navigational equipment. The messages arrive in groups from navigational receiver at a certain intervals of time, which are usually equal to one second. A navigational context is formed on the basis of the data from all messages, which came simultaneously. Except geographical coordinates, the navigational context consists of such informational, as height above sea level, the exact time, the probability of navigational solution's error, the amount of visible satellites, the direction and the speed of receiver's movement.

Navigation massage may have a length of eighty characters. Additional three characters include a sign start package “$” (hex 0x24) and a group of two final package characters CR / LF (carriage return and line feed, hex codes 0x0D 0x0A and, respectively). After the sign of the massage's beginning, the massage body is composed of fields separated by commas with no spaces. Fields of messages consist of printable ASCII-characters, with code value less than 0x7F. The first field is always the type of message, it appears a few characters of the Latin alphabet. Identifiers “GP” and “GL” represent data from the satellite constellation of GPS and GLONASS, respectively. The number of additional fields defined by the type of navigation message. End of navigation massage can be identified by the getting of a separation symbol “*” (hexadecimal code 0x2A), followed by a checksum length of one byte, which is represented by two hexadecimal digits. The checksum is calculated by using the operation “XOR” for all the bytes of the character string between “$” and “*”.

```
$GPRMC,080941.000,A,5627.77013,N,08457.06117,E,9.5,346.7,300414,,,A*69
$GPGLL,5627.76743,N,08457.06181,E,080941.000,A*36
$GPGGA,080941.000,5627.77013,N,08457.06117,E,1,18,0.6,0111.0,M,-
33.9,M,,*49
```

Fig. 1. An example of navigational messages RMC, GLL and GGA

Highly dense urban areas, which prevents the spread of navigational signal, a small amount of visible satellites, fluctuations of the atmosphere, the errors which occur during the process of triangulation held by navigational receiver – all these are the factors which can cause the appearance of the errors in the navigational context. In this case, noisy navigational signal can lead to appearance of unreliable geopositional data. Moreover, in the worst case it leads to emerge “jumping” navigational points and drastic changes in a measured speed. The example of navigational noise's impact is the drift of stationary object's coordinate on a long period of the measurement time.

For removing knowingly erroneous navigation points different filtering algorithms are used. Depending on the hardware platform, it can be computationally complex methods,

for example, Kalman filter or more simple static filtering techniques. In addition, NMEA protocol contains such parameters as DOP (Dilution of Precision), PDOP (Positional Dilution of Precision), HDOP (Horizontal Dilution of Precision), VDOP (Vertical Dilution of Precision), providing information about reducing the positioning accuracy by navigation receiver. These data can also be used during pre-filtering.

From the viewpoint of the clustering problem, the most interesting are values of the coordinate, time, speed and direction of motion of an individual navigation points. Basing on the above information about the studied data, we present a comparison of the most commonly used clustering algorithms.

III. NAVIGATION DATA CLUSTERING ALGORITHMS

I K-means algorithm

This algorithm is one of the most popular methods of data clustering, because it is fairly simple to implement. K-means algorithm is based on the works of Stuart Lloyd [1], and Hugo Steinhaus [2]. Inputs to the algorithm are the number of clusters and the coordinates of centroids (number of centroids corresponds to the number of clusters). In practice, the most commonly used metric is the Euclidean distance. On each subsequent algorithm iteration, the centers of mass of the clusters are calculated again. The iterative process continues until the centers of mass cease change their coordinates or the number of iterations reaches a determined limit.

K-means algorithm is often used for the preliminary division into groups, the further clustering is conducted by other methods. It is caused by the following algorithm disadvantages:

- It is necessary to know the number of clusters in advance;
- The algorithm is sensitive to choice of initial cluster centers;
- The algorithm is sensitive to noise and ejections.

Considering the algorithm in the context of navigation data clustering, we can conclude that it is not the most appropriate one, because of the following reasons:

- It is not always possible to identify the number of clusters and the initial cluster centers correctly;
- The navigation data contain noise and ejections, which have a great influence on the accuracy of clustering by this method;
- The algorithm poorly allocates clusters of a specific form, and navigation tracks are an example of such clusters.

I Fuzzy c-means algorithm

This algorithm is a modification of k-means algorithm, it is based on the calculation of the probability of an element belonging to a particular cluster [3]. As in k-means algorithm, at the first step the number of clusters and the centers of mass of clusters are set. However, in this algorithm also similarity measure and weight matrix of the element belonging to each of the clusters are set.

At each iteration, taking into account similarity measure and weight matrix, centers of mass of clusters and weights are recalculated.

The advantage of the algorithm is the ability to determine the extent of an element belonging to a particular cluster. This property helps to overcome the difficulties arising when using k-means algorithm. However, there still remains the difficulty of applying the algorithm in the subject area, because of the same reasons as in k-means method. Despite the fact that this algorithm can cope with noise and ejections, there is a complexity of clustering of elements, equidistant from the centers of clusters.

Therefore, we can conclude that fuzzy c-means algorithm cope with its task better than k-means algorithm, but it is not sufficiently suited for navigation data clustering.

I α- quasiequivalence algorithm

This algorithm is based on an evaluation of the distances between the elements [4]. First, it is necessary to construct an α- quasiequivalence scale.

For each element x_i of the initial set X, $i = \overline{1,Q}$ we determine evaluation of its similarity with element x_j from this set, $j = \overline{1,Q}$, according to the formula:

$$\mu_{x_i}(x_j) = 1 - \frac{d(x_i, x_j)}{\max\limits_{k \in [1,Q]} d(x_i, x_j)}. \quad (1)$$

where μ_i – normal similarity measure by the distance from the sample data x_i.

On the basis of the normal measure for each sample x_q from the set X we calculate relative similarity measure of sample data x_i and x_j relatively x_q:

$$\xi_{x_q}(x_i, x_j) = 1 - |\mu_{x_q}(x_i) - \mu_{x_q}(x_i)|, \quad (2)$$

where $i, j = \overline{1,Q}$.

The result is a three-dimensional matrix, which reflects the degree of similarity of each pair of elements with respect to all other ones.

We determine similarity measure of data samples on the set X according to the formula:

$$\xi(a,b) = T(\xi_{x_1}(a,b), ..., \xi_{x_q}(a,b)) = \min_{k \in [1,Q]} \xi_{x_k}(a,b), \quad (3)$$

where $a, b \in X$.

The result is a two-dimensional matrix containing the worst evaluations of similarity measures for each pair of elements.

We construct the transitive closure of the relation of similarity measure of data samples on the set X. For R^1_ξ:

$$R_\xi^1 = R_\xi = \{r_{ij} = \xi(x_i, x_j)\}_{i,j=\overline{1,Q}}. \quad (4)$$

For $q = \overline{2,Q}$:

$$R_\xi^q = R_\xi^{q-1} \cdot R_\xi, \quad (5)$$

where
$\{R_\xi^q\}_{ij} = r_{ij}^q = S(T(r_{ij}^{q-1}, r_{1j}), ..., T(r_{iQ}^{q-1}, r_{Qj})) = \max_{k\in[1,Q]}(\min\{r_{ik}^{q-1}, r_{kj}\})$.

On the last iteration we obtain the ratio of α-quasiequivalence on the set *X*. Then, we construct the quasiequivalence scale in the form of a set of values $R_\xi^{|X|}$, ordered ascending.

Each number α_i from the resulting set is called relation level of α-quasiequivalence. Depending on the chosen level, the initial set will be divided into appropriate clusters.

This algorithm allows to detail the division into clusters depending on the chosen level of quasiequivalence, and also helps to allocate clusters of specific form. However, the computational complexity of this algorithm is quite high. The most resource consuming operation is the construction of the transitive closure, which requires the raising of a square matrix of n order to the n-th power. Even when using binary exponentiation algorithm, the number of operations will be $O(n^3 \log n)$.

I DBSCAN algorithm

The algorithm allows to determine the membership of the element to cluster on the basis of the density distribution of the elements of the original set [5]. The initial algorithm data are distance ε, which determines the radius of the neighborhood of the point, and minPts – the minimum number of points in the neighborhood with the point under consideration.

At each iteration we consider a point from the set. If number of points in its ε-neighborhood is greater than or equal minPts, we add the point to the cluster. Otherwise the point relates to noise. Thus after traversing all the points we form clusters and the array of points attributed to noise.

Suppose minPts = 3. At fig. 2 points A, B, C and D belong to the cluster, because for these points minPts = 3. For points F and E this condition is not satisfied, but they also belong to the cluster, because they are reachable from the points for which this condition is satisfied. Point K doesn't have neighbors and is unreachable from the cluster points, therefore it is not included in the cluster. Points N and M have insufficient number of neighbor points and are unreachable from points, for which the condition minPts = 3 is satisfied.

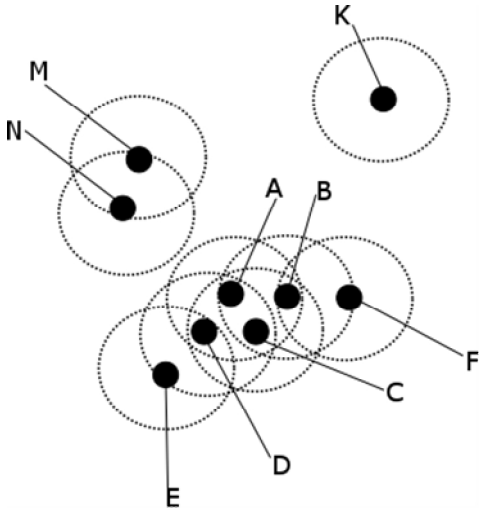

Fig. 2. Clustering by DBSCAN method

In pseudo-code algorithm is implemented as follows [5]:

```
DBSCAN(D, eps, MinPts) {
  C = 0
  for each point P in dataset D {
    if P is visited
      continue next point
    mark P as visited
    NeighborPts = regionQuery(P, eps)
    if sizeof(NeighborPts) < MinPts
      mark P as NOISE
    else {
      C = next cluster
      expandCluster(P, NeighborPts, C, eps, MinPts)
    }
  }
}

expandCluster(P, NeighborPts, C, eps, MinPts) {
  add P to cluster C
  for each point P' in NeighborPts {
    if P' is not visited {
      mark P' as visited
      NeighborPts' = regionQuery(P', eps)
      if sizeof(NeighborPts') >= MinPts
        NeighborPts = NeighborPts joined with NeighborPts'
    }
    if P' is not yet member of any cluster
      add P' to cluster C
  }
}

regionQuery(P, eps)
  return all points within P's eps-neighborhood (including P)
```

The algorithm quite successfully copes with the problem of data clustering. At the same time, it is possible to isolate the noise and carry out division of a set into clusters of specific form, that satisfies the domain requirements. However, the algorithm also has the following disadvantages:

- Belonging of boundary points (points that have the same reachability from two or more clusters) to any cluster is determined by the order of points processing;
- This algorithm does not work well for clusters with a small dot density.

I FN-DBSCAN algorithm

The algorithm is a modification of DBSCAN algorithm and provides an opportunity to assess the degree of membership to ε-neighborhood of the point [6]. In addition to the parameters ε and minPts there is also parameter ε_1, which allows to set the minimum set cardinality. I.e. there is the equation:

$$\varepsilon = d_{\max}(1-\varepsilon_1), \qquad (6)$$

where $d_{\max} = \max_{x_i, x_j \in X} d(x_i, x_j)$.

Fig. 3 shows an example, where at the same value of ε the set cardinality affects whether the considered point is a noise or a part of the cluster.

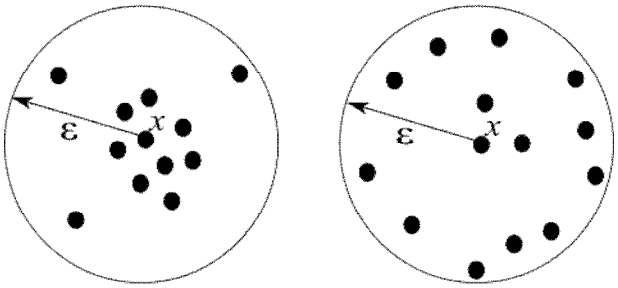

Fig. 3. Influence of the set density on the point membership to cluster in FN-DBSCAN method

Thus, when considering the ε-neighborhood of a point, all points from this neighborhood are checked for membership to ε1-neighborhood of the point. This modification allows to attribute points, outside the ε_1- neighborhood, to noise.

The algorithm is suitable for clustering with the possibility of filtering from noise. Therefore, the algorithm is suitable for working with the navigation data.

IV. CONCLUSION AND FURTHER WORK

Based on the study of clustering algorithms we can conclude that in terms of specificity of the navigation data the most suitable for their analysis is DBSCAN and its derivatives. The main feature of the navigation data is the fact that points in space are grouped in elongated regions that are difficult to cluster by centroid methods. At the same time, the approach implemented in DBSCAN algorithm, allows to find clusters of any shape.

Based on cluster analysis of a set of navigational tracks is solved a lot of problems, mainly related to the transport network control. Some of them are listed below:

- Calculation of the average traffic speed on different days of a week;
- Calculation of the average travel time for passenger transport;
- Assessment of transport infrastructure load throughout the day;
- Prediction of the traffic flow;
- Extraction or elaboration of road graph;
- Automation of the traffic control system through the traffic lights and information boards control.

Such algorithms can process in any weather conditions, at any time, in contradistinction to, for example, a network of distributed sensors. The main condition of their work is the presence of enough stable information channel through which data are transmitted from the navigation receivers. The development of such systems can have a positive economic effect during the improvement of transport infrastructure.

REFERENCES

[1] Stuart P. Lloyd. "Least squares quantization in pcm." IEEE Transactions on Information Theory, vol. 28(2), pp. 129–136, March 1982.

[2] Steinhaus H. "On the division of material bodies into parts" ("Sur la division des corps materiels en parties"). Bull. Acad. Polon. Sci., C1. III – vol IV, pp. 801—804, 1956.

[3] Jan Jantzen «Neurofuzzy Modelling». Electronic publishing.

[4] Barsegyan A.A., Kupriyanov M.S., Stepanenko V.V., Kholod I.I. Methods and models of data analysis: OLAP and Data Mining. – StP.: BKhV-Peterburg, 2004.

[5] Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei. Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M., eds. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). pp. 226–231, 1996.

[6] Ahmet Can Diker, Elvin Nasibov. Estimation of traffic congestion level via FN-DBSCAN algorithm by using GPS data. Problems of Cybernetics and Informatics (PCI), 2012 IV International Conference. pp. 65-68, 2012.