Bartłomiej GAJEWSKI, Tomasz MARTYN
WARSAW UNIVERSITY OF TECHNOLOGY, INSTITUTE OF COMPUTER SCIENCE
15/19 Nowowiejska St., 00-665 Warsaw, Poland

Spatial data clustering in independent mobile environment

Abstract

Most geolocation applications for mobile devices assume a constant connection with the network and high computational power nodes. However, with ever-developing devices it now becomes possible to establish peer-to-peer networks in case when the network can be unreachable due to special circumstances (like conflicts or natural disasters). In this paper, a method for clustering spatial data in mobile environment is discussed. A simple solution based on OPTICS algorithm with lexical distance is proposed for grouping the observations.

Keywords: peer-to-peer, data clustering, OPTICS, mobile, lexical distance.

1. Introduction

Rapid development and improvement of mobile devices technology have led to the possibility of creation of durable, affordable and relatively easily extendable augmented reality hardware. These possibilities have also been noted by military equipment developers that now have to catch up with commercial technology. It has also been spotted by military personnel in combat situations - many examples show that soldiers (e.g. in Iraq, Afghanistan, Ukraine) prefer to use their own cellular phones with civilian applications over military hardware [1].

Recently a few prototype military applications have been created for enhancing commanding and situation awareness for soldiers. A notable example is the mCOP application developed by the Military University of Technology in Warsaw [11]. This application can be used by a soldier as an interactive map that displays the current military situation. It can be also used as an interface to input a scouting data and observations. Although connectivity of this application with the centralized network is ensured by various, both military and civil communication interfaces, it also has an option of working independently. In the future, there is a possibility of creating a decentralized peer-to-peer network between the devices.

One of the common task of creating reliable situation awareness is to ensure that the input observations are not duplicated or are properly grouped. For example, when two distant users spot a vehicle near a same place, then it is not sure if it is the same vehicle or there are two different ones. Moreover, not only human users can input data, but those can also be provided by radars, detectors, satellites, etc. On a larger scale or a higher level of command, it is also desired to view the reports grouped in larger units (brigade, divisions, and so on). In the case of a small mobile device, efficient data grouping and presenting is also important.

Although there are various military algorithms that can integrate and validate the data, usually a human operator is also present. In this paper, we discuss a possibility of creating for those purposes an algorithm based on data mining.

2. Related works

There is a large number of works concerning data clustering which can be adapted for the purposes of spatial data grouping. The most notable ones are described in this section, along with other technological and algorithmic aspects related to the subject of the paper.

2.1. Data clustering algorithms

The most popular of all data clustering algorithms, *k*-means clustering, is a method of vector grouping originated from signal processing. *k*-means divides *N* observations into *k* clusters so that each observation is assigned to the cluster with the nearest mean (an average/median, medoid and so on). This results in a partitioning of the data space into Voronoi cells. The problem is computationally difficult (NP-hard), however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. One of the possible drawbacks of *k*-means clustering is that it tends to find clusters of comparable spatial extent.

Various extensions of the *k*-means clustering algorithm exist, with one of the most commonly used and cited in scientific literature being DBSCAN (first described in [2]). One of the most important changes implemented in DBSCAN is that it marks as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN is widely used for spatial data mining, for example in [3], where a special grid-based algorithm is used.

Another extension to *k*-means is the OPTICS (ordering points to identify the clustering structure) algorithm, first described in [4]. The basic idea of OPTICS is similar to DBSCAN but it addresses one of DBSCAN's major weaknesses: the problem of detecting meaningful clusters in data of varying density. In order to do so, the observations are (linearly) ordered so that points which are spatially closest become neighbors in the ordering. Additionally, a special distance is stored for each point that represents the density around the node. By that distance, using a reachability-plot, a hierarchical structure of the clusters can be obtained.

Often mistaken with *k*-means, *k*-nearest neighbors (kNN) is another machine learning technique, first described in [5]. Both for classification and regression, it can be useful to assign a weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of $1/d$, where d is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as a training set for the algorithm, though no explicit training step is required. A shortcoming of the k-NN algorithm is that it is sensitive to the local structure of the data.

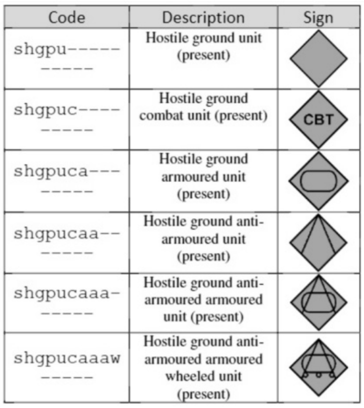
2.2. Data mining

Various other approaches to analysis of spatial data can be found – for example [6] utilizes methods of user guided knowledge discovery. Those often do not apply in the case of grouping data into clusters or finding duplicates, however other data mining techniques like classification and regression can be mixed with data clustering for results improvement. A clear explanation of those methods can be found in [7] and will not be discussed in this paper.

2.3. APP-6A codes

NATO Military Symbols for Land Based Systems was the NATO standard for military map marking symbols. It was published as Allied Procedural Publication 6A (APP-6A). The symbols are designed to enhance NATO’s joint interoperability by providing a standard set of common symbols. APP-6A constitutes a single system of joint military symbology for land based formations and units, which can be displayed for either automated map display systems or for manual map marking. It covers all of the joint services and can be used by them.

Perspectives to use the codes for interoperability and finding distances were briefly described in [8]. Example codes are presented in Fig. 1.



Code	Description	Sign
shgpu----- -----	Hostile ground unit (present)	
shgpuc---- -----	Hostile ground combat unit (present)	CBT
shgpuca--- -----	Hostile ground armoured unit (present)	
shgpucaa-- -----	Hostile ground anti-armoured unit (present)	
shgpucaaa- -----	Hostile ground anti-armoured armoured unit (present)	
shgpucaaaw -----	Hostile ground anti-armoured armoured wheeled unit (present)	

Fig. 1. Example of APP-6A codes and symbology

2.4. Lexical distance

Many works concerning the calculation of lexical distance can be found. One of the most notable ones, the Levenshtein distance [9] is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other. The Levenshtein distance was implemented for example in the Wagner–Fischer algorithm [10]

Due to the format of APP-6A codes, calculating a lexical distance between them by those algorithms might not be adequate. This is mostly because the codes are highly schematic, so many aspects of calculating a lexical distance can and should be simplified (like finding a prefix problem for example).

2.5. mCOP

Following the official description [11], mCOP in an android application was developed by WAT for decision support. It is integrated into a tactical communication network and a command server application. The application deeply integrates with any android handheld (smartphone and tablet) to deliver CTP (Common Tactical Picture) and COP (Common Operational Picture) products for lower level commanders and group leaders. The application delivers military and civilian operational pictures with standardized GIS data sources and operational symbology standards (STD2525, APP6A, crisis management).

mCOP supports an implementation of detailed battlespace information: units, equipment and installations, their affiliation, status. The commander is able to gather specific tactical information on equipment, warfare as well as unit supplies. Furthermore, the detailed combat information contains not only static data, but also a description of units behavior in time (the current tasks and history).

The application has a possibility to work independently, however it does not utilize an option to establish direct P2P connections with other devices.

2.6. Mobile peer-to-peer systems

A special usage case of the desired algorithm includes a situation where the network nodes with high computational power (preferably servers) are not available. Situations like that are possible when the peer-to-peer mobile networks must be established. Possible software and hardware solution that utilizes WiFi interfaces for creating mobile P2P networks are described e.g. in [12].

3. The proposed solution

Basing on the level of knowledge and analysis of existing algorithms, it was decided that the OPTICS algorithm provided the most suitable effects of grouping. However, some careful parameterization and modification must be done.

The basic idea of the proposed solution is to use the Euclidean distance of two spatial objects (observations) but also to use the factor of lexical distance between codes of those objects.

3.1. Number of groups selection

As the OPTICS algorithm requires a number of desired groups, which in this case might vary, it is assumed that the number of groups will be specified by the user. The number might vary from 2 to $N - 1$ for N observations (as there is no need for using the algorithm for 1 group, and no sense for finding N groups).

At a large number of groups (larger than $N/2$) pairs of possible duplicates will be proposed. Then, the user can decide if it is a true or false duplicate proposal. Those decisions can further be used to improve the grouping algorithm by another data mining process.

Another approach is to depend the number of groups to the map zoom level. Then, whenever the user zooms out, a number of groups is reduced down to 3. It is assumed that the user would not want the data to be grouped for the largest possible. This implies that the grouping has to be redone for each change for zoom level, however the distances do not have to be recalculated with each zoom change, but only for additional data.

An important output of the grouping algorithm is the number of grouped observations, which have to be given back to the user.

3.2. Distance calculation

Because of the APP-A6 codes feature of being more specific in every next letter, a special lexical distance must be calculated for each compared pair of the observations. The distance is calculated by the formula:

$$D(a,b) = De(a,b) * (1 + L(a,b))$$

where $D(a,b)$ is the distance between observation of A and B used by the density algorithm, $De(a,b)$ is the Euclidean distance between two objects calculated from geographical coordinates, and $L(a,b)$ is the lexical distance between them. $L(a,b)$ is calculated by the formula:

$$L(a,b) = \sum_{i=0}^{n} v_i * dl(a[i], b[i])$$

where again n is the length of the code, $a[i]$ is a character at position i, and dl is a cost function of the difference between the characters. This function value largely depends on the character number (i) and interpretation. Then, factor v_i is used to balance the cost of each character.

An example of visualization of the desired grouping algorithm results is shown in Fig. 2.

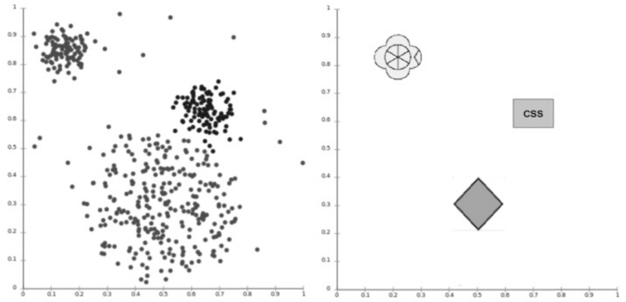


Fig. 2. Desired output of the grouping algorithm

4. Tests

The proposed solution has to be put under tests and careful calibration before it can be used in practice. For example, the parameters of the distance calculation function should be optimized.

The proposed solution was implemented into a branch of mCOP application, using the ELKI's OPTICS implementation as a base.

As mCOP is a mobile application based on Android devices, the screen space is a huge limitation. For the test purposes, it was assumed that the readable number of units that can be displayed is 12, therefore only 12 biggest clusters are going to be chosen.

The number of units to group is variable. Each time the units were randomly deleted out of a group of a 1000 in a generated scenario, to match the test group quantity.

The average of 10 algorithm execution times for each number of units are presented in Table 1. The example output of the grouping algorithm is shown in Fig. 3.

Tab. 1. Execution times

Number of units:	50	100	200	500	1000
Execution time:	0.16 s	0.24 s	0.33 s	0.51 s	0.79 s

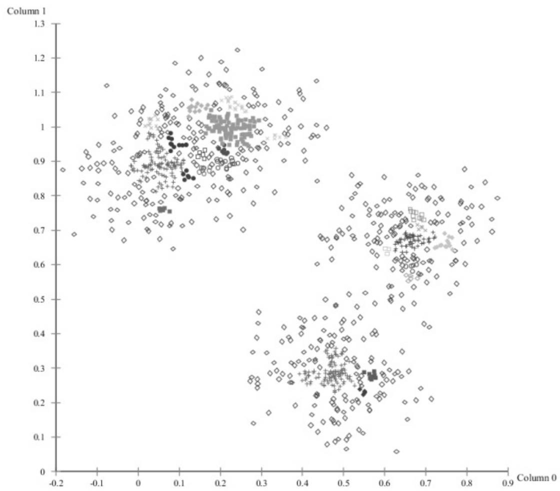


Fig. 2. Test data grouped in 12 clusters. Remaining points are ignored

5. Future Work

Although the efficiency of the solution is acceptable, the output should be confronted with real user experience. Then, basing on the users opinions, the solution parameters can be optimized preferably by another data mining procedure.

Possible limitations of the solution, connected with relatively small acceptable data sets, can be resolved by using a grid approach similar to the one described in [3]. With that, also a larger scale real time clustering of over 1000 elements can be achieved.

Moreover, the solution is planned to be used as a base for a more complicated duplicate detection system.

6. Conclusions

As a result of this work, a solution was proposed for a problem of spatial data grouping in mobile environment by using a dedicated variation of OPTICS algorithm combined with lexical distance.

The proposed solution can be implemented in mobile systems due to a relatively small range of operations (approximately 2×2 km), thus implementing acceptable small amounts of data to be clustered in the acceptable computation time, shorter that 1 s.

The proposed system and solution can have adaptations in a few different civilian fields. It is possible to adapt it to find possible data duplicates or correlations of reports in emergency management systems. Finding a larger clusters of events can automatically lead to rising the alert levels and starting the procedures of crisis management.

Alternative fields can include the variety of mobile applications connected with reporting and preliminary classification of observations, like e.g. Yanosik application, birds, ships and planes spotting, or even games like geohashing.

7. References

[1] Pence Harry E.: Smartphones, smart objects, and augmented reality. The Reference Librarian 52.1-2 (2010), pp. 136-145.

[2] Ester Martin, et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd. Vol. 96. No. 34. 1996.

[3] Wang Wei, Jiong Yang, and Muntz Richard: STING: A statistical information grid approach to spatial data mining. VLDB. Vol. 97. 1997.

[4] Ankerst Mihael, et al.: OPTICS: ordering points to identify the clustering structure. ACM Sigmod Record. Vol. 28. No. 2. ACM, 1999.

[5] Altman N. S.: An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46 (3): 175–185, 1992.

[6] Koperski K., and Jiawei Han: Data mining methods for the analysis of large geographic databases. Proc. of 10th Annual Conf. on GIS, Vancouver, BC. 1996.

[7] Cichosz P.: Data Mining Algorithms: Explained Using R. John Wiley & Sons, 2014.

[8] Chmielewski M., Gałka A.: Automated mapping JC3IEDM data in tactical symbology standards for Common Operational Picture services. Proceedings of the Military Communications and Information Systems Conference MCC. 2009.

[9] Levenshtein V.I.: Binary codes capable of correcting deletions, insertions, and reversals." Soviet physics doklady. Vol. 10. No. 8. 1966.

[10] Navarro G.: A guided tour to approximate string matching. ACM computing surveys (CSUR) 33.1 (2001): 31-88.

[11] mCOP application homepage at http://uranus.wat.edu.pl:8808/wiki/index.php/MCOP

[12] Gajewski B.K., and Martyn T.: Smart mobile P2P communication optimization for close range by an automatic interface switch. Measurement Automation and Monitoring, vol. 61, no 07, 2015, pp. 317–319.

[13] ELKI: Environment for Developing KDD-Applications Supported by Index-Structures Open Source project. http://elki.dbs.ifi.lmu.de/

Received: 07.02.2016 *Paper reviewed* *Accepted: 04.04.2016*

Bartlomiej GAJEWSKI, MSc

PhDstudent at Warsaw University of Technology. Specializes in mobile communication technologies, geolocalization and analysis of geographical data. Generation of location-based data and content is his main area of research.

e-mail: b.gajewski@ii.pw.edu.pl

Tomasz MARTYN, PhD, DSc

Assistant Professor at Institute of Computer Science, Warsaw University of Technology. Author and coauthor of 4 books as well as many research papers and articles published in various peer-reviewed international journals and conferences. His research interests include fractal geometry, scientific visualization and real-time rendering techniques.

e-mail: martyn@ii.pw.edu.pl