Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: 0005-1144 (Print) 1848-3380 (Online) Journal homepage: https://tandfonline.com/loi/taut20

Parallel mining of uncertain data using segmentation of data set area and Voronoi diagrams

Ivica Lukić, Željko Hocenski, Mirko Köhler & Tomislav Galba

To cite this article: Ivica Lukić, Željko Hocenski, Mirko Köhler & Tomislav Galba (2018) Parallel mining of uncertain data using segmentation of data set area and Voronoi diagrams, Automatika, 59:3-4, 349-356, DOI: 10.1080/00051144.2018.1541645

To link to this article: https://doi.org/10.1080/00051144.2018.1541645

Published online: 25 Nov 2018.

AUTOMATIKA
2018, VOL. 59, NOS. 3–4, 349–356
https://doi.org/10.1080/00051144.2018.1541645

ORIGINAL SCIENTIFIC PAPER

OPEN ACCESS

Parallel mining of uncertain data using segmentation of data set area and Voronoi diagrams

Ivica Lukić, Željko Hocenski, Mirko Köhler and Tomislav Galba

Department of Computer Engineering and Automation, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Josip Juraj Strossmayer University of Osijek, Osijek, Croatia

ABSTRACT

Clustering of uncertain objects in large uncertain databases and problem of mining uncertain data has been well studied. In this paper, clustering of uncertain objects with location uncertainty is studied. Moving objects, like mobile devices, report their locations periodically, thus their locations are uncertain and best described by a probability density function. The number of objects in a database can be large which makes the process of mining accurate data, a challenging and time consuming task. Authors will give an overview of existing clustering methods and present a new approach for data mining and parallel computing of clustering problems. All existing methods use pruning to avoid expected distance calculations. It is required to calculate the expected distance numerical integration, which is time-consuming. Therefore, a new method, called Segmentation of Data Set Area-Parallel, is proposed. In this method, a data set area is divided into many small segments. Only clusters and objects in that segment are observed. The number of segments is calculated using the number and location of clusters. The use of segments gives the possibility of parallel computing, because segments are mutually independent. Thus, each segment can be computed on multiple cores.

Paralelno klasteriranje nesigurnih podatka koristeći se segmentacijom područja podataka i Voronojevim dijagramima. Klasteriranje podataka s nesigurnošću je vrlo proučavano područje u velikim bazama nesigurnih podataka. U takvim bazama podataka teško je pronaći korisne podatke u mnoštvu podataka s nesigurnošću. U ovom radu proučavano je klasteriranje objekata koji imaju nesigurnost položaja. Većina pokretnih objekata, kao što su mobilni uređaji, periodički izvještava svoj položaj, stoga je njihov položaj neprecizan te se mora opisati funkcijom gustoće vjerojatnosti. Broj objekata u bazi podataka može biti jako velik i doći do točnih podataka je izazovan zadatak i zahtijeva puno vremena. Sve metode za klasteriranje nesigurnih podataka koriste slične principe. Ovim radom predložen je nov pristup. Prvo je dan pregled postojećih metoda, a nakon toga predložena je nova metoda za paralelno klasteriranje nesigurnih podataka. Sve postojeće metode koriste se različitim postupcima pročišćavanja kako bi se izbjeglo računanje očekivane udaljenosti jer ono uključuje numeričke integracije i zahtijeva puno vremena. Predložili smo metodu nazvanu *paralelna segmentacija područja podataka*. U toj metodi, klastersko područje podijeljeno je u mnogo malih segmenata te se promatraju samo klasteri i objekti u tim malim segmentima. Broj segmenata izračunava se pomoću broja i položaja klastera u prostoru. To nam daje mogućnost za paralelno računanje jer segmenti su međusobno neovisni te se tako svaki segment može računati na više procesorskih jezgri.

KEYWORDS
Clustering algorithms; data mining; data uncertainty; Euclidean distance; parallel algorithms

KLJUčNE RIJEčI
Algoritmi za klasteriranje; Euklidska udaljenost; nesigurni podaci; rudarenje podataka; paralelno procesiranje

1. Introduction

In many databases, data contain uncertainty, so mining useful data from such uncertain databases is not a simple task. Different factors, such as a measurement error, sampling discrepancy and outdated data source, contribute to data uncertainty. To cluster data with location uncertainty, various methods are used, such as UK-means, MinMax pruning and Voronoi pruning. Clustered objects are mutually similar and near to the cluster centre, thus forming similar groups. The location of moving objects is uncertain because object locations are reported periodically. Therefore, the object's exact location must be estimated using the last known location and uncertainty value. Uncertainty depends on the location measurement error, speed of moving objects, last reported direction, elapsed time etc. Clustering methods can be used for many purposes: where tracking of moving objects is needed, such as mobile devices, traffic services, etc. An uncertain object is not represented by the exact location, but by the uncertainty region which is represented by a probability-density function (PDF). In this paper, object's locations are presented in 2D space and two-dimensional uncertainty. In real life applications, PDF can be specified using Gaussian distributions with means and variances [1].

For Gaussian distributions, a density function is exponentially dropped, which means that the probability density outside a certain region is zero. Thus, each

CONTACT Ivica Lukić ivica.lukic@ferit.hr

object can be bounded by a finite bounding region. The uncertainty region of a moving object is limited by the maximum speed and elapsed time. The main issue in the clustering process is the execution time, because efficiency is very important in real-time applications. In the clustering process, calculating the expected distances is the most demanding process, because numerical integration is involved using a large number of sample points for each PDF.

In [2], some pruning methods are introduced. In these pruning methods, bounding regions for each object are used to prune clusters. Using pruning methods, some clusters are eliminated as the candidate clusters when a closer cluster for the observed object is found. Thus, the computation of the expected distance from the object to pruned clusters is avoided, and the computational cost is saved. In [3], pruning methods based on Voronoi diagrams are introduced. The spatial relationship among cluster representatives in these pruning methods is taken into consideration. Voronoi pruning is an improvement of the basic bounding region method. In this paper, a new clustering method is presented, and some issues that are not presented in the previous methods are discussed. This new method is called Segmentation of Data Set Area – Parallel (SDSA-P). A data set area is divided into small segments and only clusters and objects in those segments are observed, thus the number of object-cluster observations is decreased. SDSA-P creates segments in such a way that all segments are mutually independent and can therefore be executed as parallel processes. New investments are not needed because parallel methods can be executed on a multi-core processor on one computer, it is not necessary to by new hardware like graphic cards or more computers to get parallel processing.

2. Overview of existing methods

An object location can be uncertain in two ways: existential uncertainty and value uncertainty. An object is existentially uncertain if it is uncertain whether that object exists. In a relational database, an object is associated with a probability value that indicates the confidence of its presence [4]. Efficient query evaluation on probabilistic databases is well explained in [5]. In the second case, the object is known to exist, but the object's value is uncertain, and the location is not precise.

The object is modelled as a minimum bounding rectangle (MBR), which bounds all possible location values. In [1,4,6], MBR is described by a probability density function. In this paper, clustering objects with value uncertainty, such as location uncertainty, are studied. MBR can be presented as rectangle, square or circle. In this paper, MBR is presented as a rectangle, because a rectangle shape is used in all cited references. Cluster analysis is used to identify the most probable values of model parameters [7] (such as means of Gaussian mixtures), high-density connected regions [8] (such as areas with high population density), or minimize the total squared distance to cluster centres [9]. In this paper, the latter case is studied with the aim of minimizing the total squared distance from objects to cluster centres. Distance can be measured, for example, a city-block distance, Minkowski distance [10], Euclidian distance etc. Data uncertainty is represented by a probability density function, which is represented by sets of sample values, and a large number of samples are needed to improve the accuracy. The distance must be calculated between all samples and the computational cost is higher compared to the simple distance calculation [11]. In the basic UK-means clustering algorithm, the expected distance (ED) is calculated from all objects to all clusters, which makes the algorithm ineffective [12]. In [2], the MinMax pruning method, which is significantly more effective than UK-means, is presented. Clustering uncertain data using Voronoi diagrams and R-trees is presented in [13]. The mentioned method is combined with the SDSA method presented in [14], and a new and improved SDSA-P method is presented in this paper. The SDSA method is used for segmentation of the data set area and Voronoi diagrams for cluster pruning. By synthesizing these two methods, the new method acquires the best pruning qualities taken from Voronoi diagrams, and by using the SDSA method, a data set area is divided into segments which are processed in parallel.

3. MinMax pruning

MinMax pruning method is presented in [2] as an improvement of Uk-means. Before the method is used, definitions are explained.

Definition 3.1 (Uncertain objects): Uncertain objects are a collection of data $O=\{o_1, \ldots, o_n\}$ in an m dimensional space R^m, where the distance between two objects is $d(o_i,o_j) \geq 0$.

Definition 3.2 (Probability density function): The probability density function of an object at point x inside R^m is $f_i(x) > 0$ (for points outside MBR $f_i(x) = 0$) and for all points $\int_{x\in R^m} f_i(x)\mathrm{d}x = 1$.

Definition 3.3 (expected distance): Expected distance from object o_i, to any point y is calculated using next formula:

$$ED(o_i, y) = \int_{x\in A_i} d(x,y)f_i(x)\mathrm{d}x \qquad (1)$$

where A_i is finite region and $f_i(x) = 0$ outside region A_i.

Definition 3.4 (clustering): The goal of clustering is to find a set of cluster points $C=\{c_1, \ldots, c_k\}$ and relation between objects and clusters $h{:}\{1, \ldots, n\} \rightarrow$

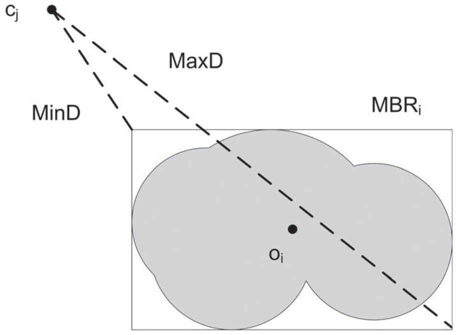

Figure 1. MBR of object, MinD and MaxD distance from an object to a cluster.

{*1*, ... ,*k*}, which minimize the total expected distance among $TED = \sum_{i=1}^{n} ED(o_i, c_{h(i)})$ is minimized.

In MinMax pruning method minimum bounding rectangle MBR_i is used to avoid unnecessary expected distance calculation. MBR is the smallest rectangle that is equal to finite region A_i as shown in Figure 1.

Using MBR and inexpensive Euclidian distance calculations, some clusters are pruned as candidates for an object. Thus, expected distances from those clusters to the object are not computed. For each object minimum distance to cluster is defined:

$$\text{Min}D(o_i, c_j) = \min_{x \in MBR_i} d(x, c_j) \tag{2}$$

Maximum distance to cluster:

$$\text{Max}D(o_i, c_j) = \max_{x \in MBR_i} d(x, c_j) \tag{3}$$

Smallest distance among all maximum distances:

$$\text{MinMax}D(o_i, c_j) = \min_{c_j \in C}\{\text{Max}D({}_i, c_j)\} \tag{4}$$

It is obvious that the minimum distance from an object to a cluster is smaller and the maximum distance is larger than the expected distance from an object to a cluster, as shown in the following formula:

$$\text{Min}D(o_i, c_j) \leq ED(o_i, c_j) \leq \text{Max}D(o_i, c_j) \tag{5}$$

Then, if it is satisfied the next condition:

$$\text{Min}D(o_i, c_p) \geq \text{Max}D(o_i, c_j)$$

Without computing the expected distances, cluster c_p is pruned from object $o_{i,}$ the expected distance is not calculated and the execution time is shortened. MinMax pruning is described by the following algorithm:

```
**for all** $c_j \in C$ **and objects** $o_i$ **do**
Compute Min$D(o_i, c_j)$ and Max$D(o_i, c_j)$
Compute MinMax$D(o_i)$
**for all** $c_j \in C$ **do**
**if** Min$D(o_i, c_j) >$ MinMax$D(o_i)$**then**
Remove $c_j$ from $CC_i$/*candidate clusters*/
**for all** remaining clusters calculate ED
```

4. Voronoi pruning

In contrast to MinMax pruning, in the Voronoi pruning method, the geometric structure of R^m is observed, which means that spatial relationships between clusters are considered. In [3], it has been proved that Voronoi pruning is theoretically strictly stronger than MinMax pruning. The same authors presented hybrid methods for even more efficient pruning. With a set of clusters $C = \{c_1, \ldots, c_k\}$, space R^m is divided into k cells with the following property:

$$d(x, c_p) \leq d(x, c_q) \quad \forall x \in V(c_p), c_p \neq c_q \tag{6}$$

After constructing Voronoi diagrams, the next step in the iteration is Voronoi cell pruning, in which it is checked whether MBR_i of object o_i is completely inside Voronoi cell $V(c_j)$. If so, object o_i is assigned to cluster c_j and there is no need for ED computation because all other clusters are pruned. Figure 2 shows that MBR_2 of object o_2 is completely inside Voronoi cell V_3, thus object o_2 is assigned to cluster c_3. However, MBR_1 is particularly inside Voronoi cell V_3 and object o_1 cannot be assigned to cluster c_3. For all objects which are not assigned to any clusters, the expected distance must be calculated. Voronoi pruning can be combined to develop hybrid methods, such as the combination of Voronoi and bisector pruning, cluster shift method, etc

Voronoi pruning in combination with bisector pruning method is described by the following algorithm:

5. Segmentation of data set area – parallel

This new method for parallel mining of uncertain data is combining Voronoi diagrams and the SDSA method presented in [14]. In the new Segmentation of Data

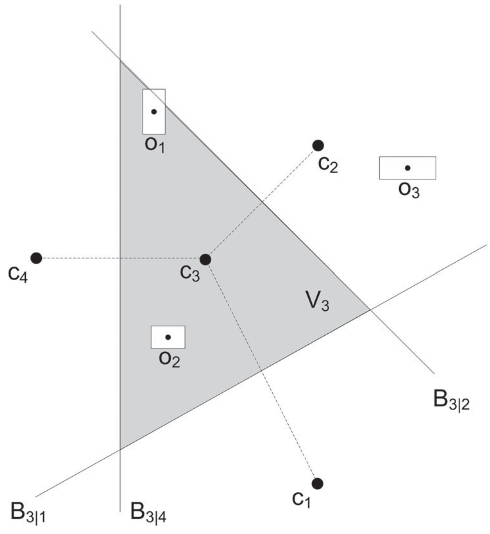

Figure 2. Voronoi cell for cluster c$_3$.

```
Compute the Voronoi diagram for $C = \{c_1, \ldots, c_m\}$
for all $c_j \in C$ and objects $o_i$ do
    if $MBR_i$ completely inside $V(c_j)$
        object $o_1$ is assigned to cluster $c_j$
for all unsigned objects and distinct clusters $c_p, c_q$
    if $MBR_i$ on same side of $B_{p/q}$ as cluster $c_p$
        Remove $c_q$ from $CC_i$/*candidate clusters*/
for all remaining candidate clusters calculate ED
```

Set Area-Parallel (SDSA-P) method, the process of segmentation is used, as shown in Figure 3. The SDSA-P method can be combined with many existing methods to significantly improve the execution times of original methods. It has been experimentally proved in [14].

A data set area is divided into many segments to reduce object-cluster observations and enable parallel. Only objects and clusters inside a segment are observed. According to this principle, many observations of object-cluster pairs are removed and the computational cost saved. The smaller the segments are, the more effective the clustering process is; however, segmentation is limited by the number of clusters and their position. It is important to note there is two type of segments, object and cluster segment. Clusters segment is larger than object segment, because object segment must be surrounded with clusters to ensure that clusters inside observed cluster segment are closer to objects inside object segment than any cluster outside observed cluster segment, such as clusters in other cluster segment. A cluster segment contains an object segment and all surrounding object segments. In Figure 3, the process of segmentation is shown. Starting from Figure 3(a) with a total data set area and all 16 object segments. In Figure 3(b), the outer object segment with four objects is shown. It is surrounded with three objects segments. These four segments create a cluster segment, and ensure that any outside cluster is closer to object segment. In Figure 3(c), one of the inner segments surrounded with object segments is shown. Inside that segment, six objects and associated MBRs are shown. For example, to observe each object-cluster pair there are 640000 observations in a data set with 10000 objects and 64 clusters. If the SDSA-P method is used, objects O can be divided into 16 segments O_{seg}, and clusters C are divided into four segments C_{seg}. The average number of objects in one segment is 625, and the average number of clusters is 16. The total number of calculations for all 16 segments is 160,000 (16 segments × 16 clusters × 625 objects), as opposed to 640,000 calculations. A decrease in the total number of calculations is proportional to the number of cluster segments. However, segmentation has size limits, which is dependent on the number of clusters and their position. An object segment must be surrounded by clusters. If the number of clusters is high, then segments are very small and can significantly speed up the clustering process. In this paper and in the experiments, objects are divided into 16, and clusters into four segments. These numbers are used because experiments are conducted on processor with four cores, and each core process four cluster segment. Further parallelization increasing number of segments would have no effect. SDSA-P method has no restriction for number of cluster or object segments, they are limited by hardware configuration, cluster location

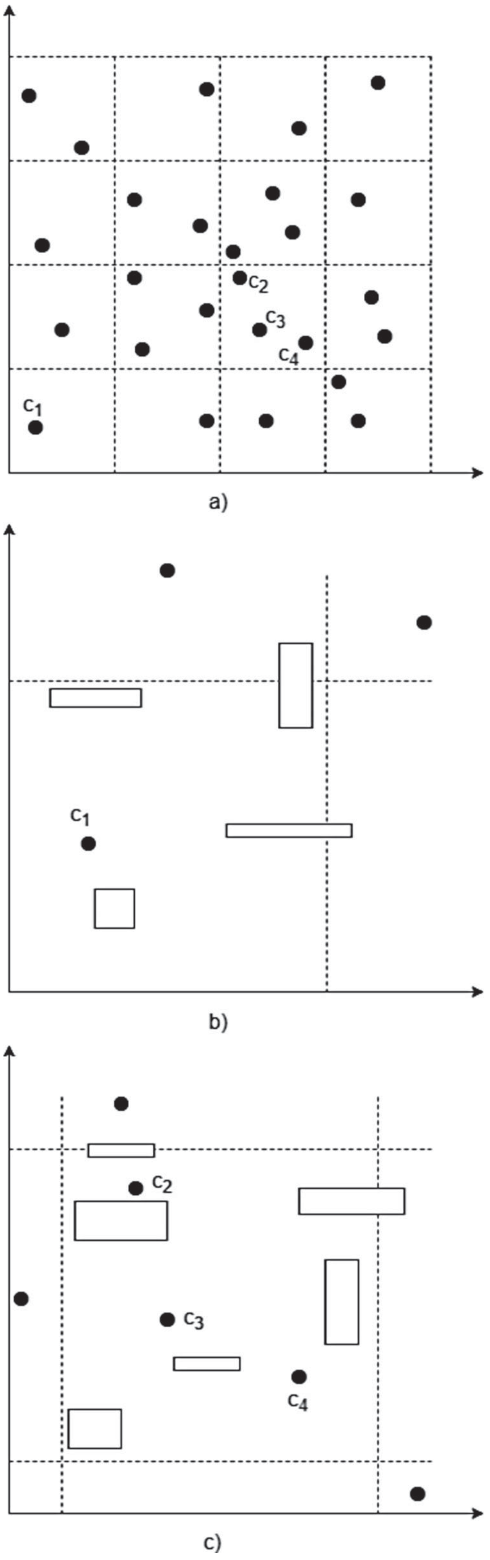

Figure 3. Process of segmentation.

and their number to ensure proper segments. Due to the fact that all segments are mutually independent, pruning methods (MinMax, Voronoi diagrams) can be executed for each segment separately (as parallel processes). The SDSA-P method is described using the following pseudo-algorithm:

```
for all segments do
  Compute the Voronoi diagram for $C_{seg} \in C$
  for all $c_j \in C_{seg}$ and objects $o_i \in O_{seg}$ do
    if $MBR_i$ completely inside $V(c_j)$
        object $o_1$ is assigned to cluster $c_j$
  for all unsigned objects and distinct clusters $c_p, c_q$
    if $MBR_i$ on same side of $B_{p/q}$ as cluster $c_p$
      Remove $C_q$ from $CC_i$/*candidate clusters*/
    for all remaining candidate clusters calculate ED
```

There are different types of parallel computing systems, like clusters, grids, distributed systems, multi-core, many-core processors and cloud computing systems [15,16]. Parallelization is also possible using CUDA [17,18] which is another type of parallelization and has possibility to be used for SDSA-P method. In this paper, the multi-core processor system is used, and according to Amdahl's law [19] speedup is proportional to number of parallel processes. Uncertain data can be clustered in distributed peer-to-peer networks [20]. In the following section, the execution time of a serial process, parallel processes on two cores and on four-core processor is measured. Experiments are conducted for SDSA-P 2 core and SDSA-P 4 cores. The pseudo-algorithm can be executed serially and in parallel. The serial execution is described in [14], and the execution time is measured. It is stated that calculations for each segment are interdependent and can be calculated in parallel. Thus, with minor implementation changes, the algorithm is transformed for parallel execution using SDSA-P pseudo-algorithm and Matlab Parallel Computing Toolbox for parallel processing. The algorithm is searching for a free process. If it is available, it runs this segment in a new process. More algorithms for clustering uncertain data can be found in [21,22].

6. Experimental set-up

For a data set of n, corresponding uncertainties described by MBRs are generated. All objects are located in [0,100] x [0,100] 2D space. MBRs are generated to have random side length for each object, but are bounded by the maximum length d of 10. For each object, MBR is divided into a sqrt(s) × sqrt(s) grid, where s is the number of samples per object's probability density function. Probability for each cell is randomly generated and the sum of all probabilities must be equal to 1. All objects are randomly positioned in space. For this data set, the initial cluster centres are chosen uniformly from the 2D space mentioned above.

Table 1. Basic data set.

Parameter	Description	Value
n	Number of uncertain objects	10000
k	Number of clusters	49
d	Maximum side length of MBR	10
s	Number of sample point per object	196

Table 2. Basic data set experimental results

Method	Execution time (s)
SDSA	93.75
SDSA-P 2 cores	62.25
SDSA-P 4 cores	44.63

The basic data set is represented in Table 1. It is the most widely used data set in literature.

Each experiment is repeated 20 times to obtain a more accurate average result. The experiment results are compared to ensure that each method has the same clustering results. All methods are implemented in MATLAB 7.0 and carried out on PC with a processor Intel Core i7-870, 2.93 GHz, with four physical cores, and 4GB of main memory.

7. Basic parameters experiments

First experiment is conducted with parameters shown in Table 1. Results are shown in Table 2.

In Table 2, the execution times are given in seconds, and the serial process is compared to parallel processes on two cores and four cores. The SDSA-P 2 cores method is 33.39% faster than the SDSA method, while SDSA-P 4 cores is 53.4% faster.

It is important to note that the used processor has only two physical cores and two virtual cores, and the execution time of the SDSA-P 4 cores method would be better on a processor with four physical cores. Hence, the SDSA-P 4 cores method is recommended for the basic parameters. In Figure 4, segments calculation for the SDSA-P 2 cores method is shown. Grey segments marked with number one are processed by the first core and white segments with number two by the second core. In Figure 5, segments calculation for the SDSA-P 4 cores method is shown. Grey segments are calculated by the first and the fourth core, and white segments by the second and the third core. In each core, there are two inner and two outer segments for the purpose of better distribution of the execution time, because the inside segments are surrounded by more clusters and require longer execution time.

8. Experiments with various numbers of objects

In these experiments, various numbers of objects are used, but other parameters retained the basic values. Experiments started with 5000 objects and ended with 40000. The experimental results are shown in Figure 6.

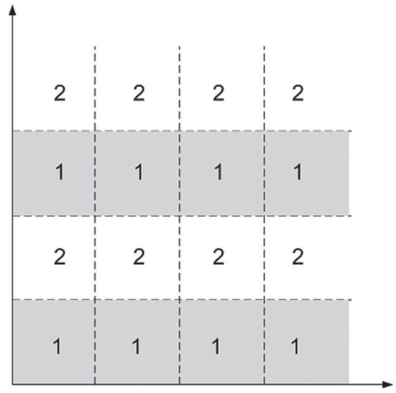

Figure 4. Segments calculation in two core process.

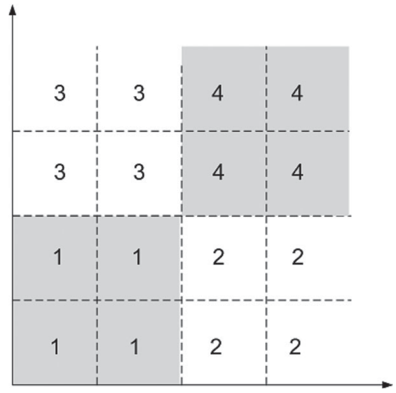

Figure 5. Segments calculation in four cores process.

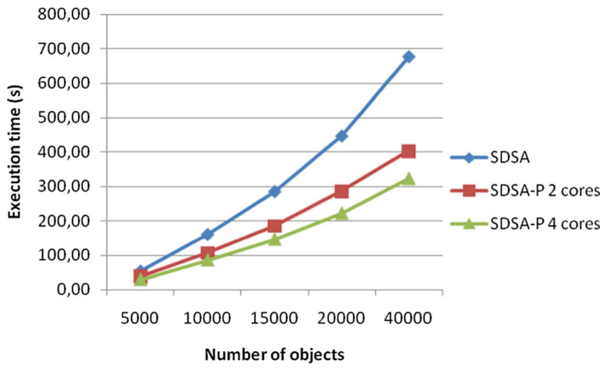

Figure 6. Experimental results for various numbers of objects.

For a small number of objects (less than 5000), parallelization has a minor execution time improvement compared to a serial method, because communication between processes contributes significantly to the total execution time. This communication occurs after each iteration when processes for each cluster share new position and objects assigned to that cluster. For a larger number of objects, communication between processes is negligible in total time, and benefits of parallelization are visible. The execution times ratio of SDSA and SDSA-P 2 cores is shown in Figure 7.

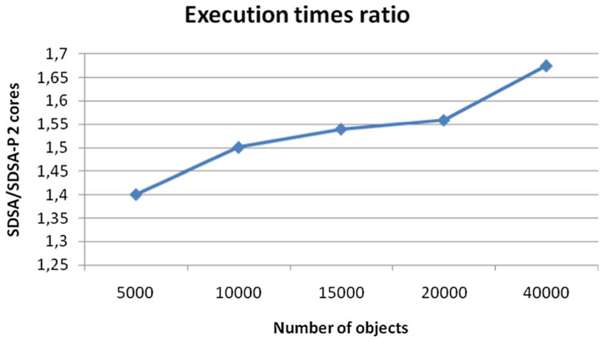

Figure 7. SDSA and SDSA-P 2 cores execution times ratio for various numbers of objects.

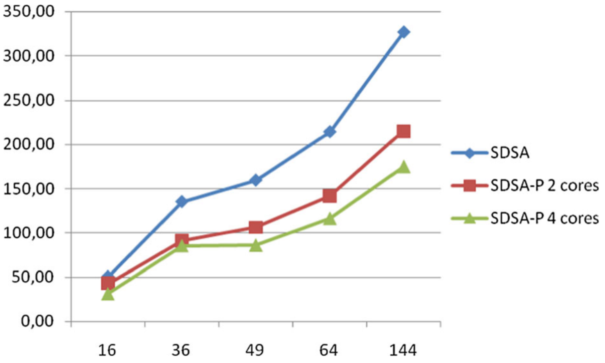

Figure 8. Experimental results for various numbers of clusters.

The ratio for the SDSA and SDSA-P 4 cores method is similar and there is no need to present it in a figure. For 10000 objects, the execution times of parallel methods are significantly better. For 40000 objects, the execution time of the SDSA method is 675.25 seconds, SDSA-P 2 cores 455.01 seconds, and SDSA-P 4 cores 376.55 seconds. In this case, execution time improvements are 32.61% and 44.35%. In Figure 7, the execution time ratio rises with the number of objects. Therefore, when more computational power is needed, a parallel method for a larger number of objects is recommended, while for small number of objects improvements is not significant because of communication after each iteration.

9. Experiments with various numbers of clusters

In these experiments, the number of clusters k varies from 16 to 144 with the basic values for other parameters, as shown in Figure 8. In Figure 9, the execution times ratio of a serial and two cores parallel process for different number of clusters is shown. As the number of clusters increases, cluster centres are closer and there is less probability for successful cluster pruning. Consequently, more ED calculations must be performed to assign an object to the cluster. ED calculations have a significant contribution to the total execution time.

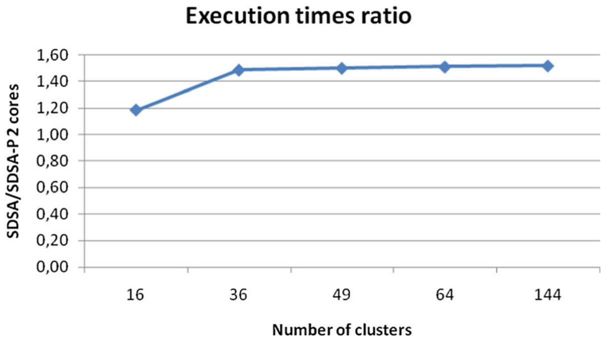

Figure 9. SDSA and SDSA-P 2 cores execution times ratio for various numbers of clusters.

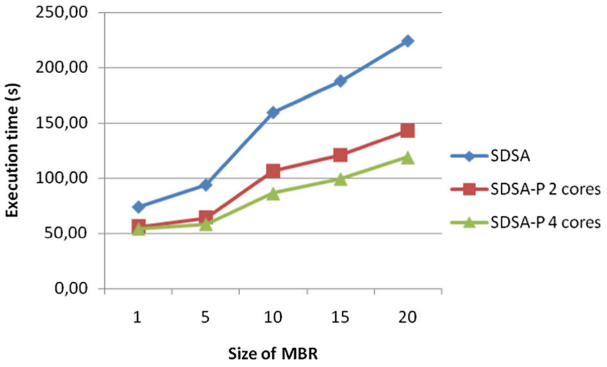

Figure 10. Experimental results for various sizes of MBR.

Thus, with a larger number of clusters, parallel processes perform better than a serial process. ED calculations are distributed to more processes and each process calculates only one part of ED calculations.

Based on the Figure 9, we can conclude that the advantages of parallelization become apparent as the number of clusters is higher, because the execution times ratio is higher as the number of clusters increases.

10. Experiments with various size of MBR

In these experiments, the size of MBR varies from 1 to 20 with the basic values for other parameters. The results are shown in Figure 10. It is visible that the execution time increases with the size of MBR. With a larger size of MBR, it is more probable that the MBR of an object will overlap with the borders of Voronoi cells causing unsuccessful pruning and more ED calculations.

Again, parallel processes are more effective because each process executes only a part of ED calculations, whereas a serial process calculates all ED calculations. Thus, in this case, parallel methods are more effective. In Figure 11 the execution time ratio for various sizes of MBR is shown. Ratio ranges from 1.32 for MBR $= 1$ to 1.57 for MBR $= 20$.

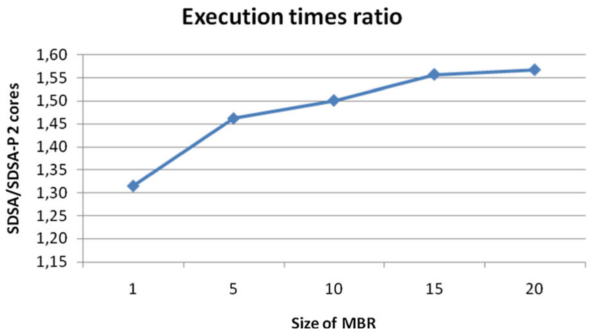

Figure 11. SDSA and SDSA-P 2 cores execution times ratio for various sizes of MBR.

11. Conclusion

In this paper, methods for parallel clustering of uncertain data are studied, and it has been experimentally confirmed that those methods yield better results than the serial method. Too much of the execution time is spent on ED calculations, because it involves numerical integration. The goal of all methods is to avoid ED calculations. However, ED calculations cannot always be avoided. In this paper, it has been found out which parts of the clustering method can be parallelized. SDSA-P methods were the best choice for parallelization. Calculation using segmentation can be easily parallelized with some changes in the serial method. The effectiveness of parallel methods has been experimentally proved, because the execution time has improved as the number of objects and clusters, and MBR has increased. This is due to the influence of communication between processes which is less important in the total computational costs. In our experiments, four parallel processes on four cores are used because of the hardware limitations. However, in practice, the number of parallel processes, which are employed, can be equal to the number of segments. The number of segments was 16 and, theoretically, 16 parallel processes could have been used. The experiment results showed that the parallel method outperformed the existing serial methods. In future work, the segmentation algorithm will be improved for optimal calculation sizes and number of segments, according to the number of clusters and their position. Using this algorithm, the maximum number of segments will be used for each cluster analysis. Thus, the number of parallel processes would be maximized and the execution time more reduced.

Disclosure statement

No potential conflict of interest was reported by the author.

References

[1] Wolfson O, Sistla P, Chamberlain S, et al. Updating and querying databases that track mobile units. Distrib Parallel Databases. 1999;7(3):257–387.

[2] Ngai WK, Kao B, Chui CK, et al. Efficient clustering of uncertain data. In ICDM, 2006. p. 436–445.

[3] Kao B, Lee SD, Cheung DW, et al. Clustering uncertain data using Voronoi diagrams. Data Mining, 2008. ICDM '08. Eighth IEEE International Conference; 2008 Dec 15–19. p. 333–342.

[4] Cheng R, Kalashnikov D, Prabhakar S. Querying imprecise data in moving object environments. IEEE TKDE. 2004;16(9):1112–1127.

[5] Nilesh D, Suciu D. Efficient query evaluation on probabilistic databases. In Proc. of VLDB Conference; 2004. p. 864–875.

[6] Cheng R, Xia X, Prabhakar S, et al. Efficient indexing methods for probabilistic threshold queries over uncertain data. In Proc. of VLDB Conference; 2004.

[7] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EMalgorithm. J R Stat Soc. 1977;B39:1–38.

[8] Zhang X X, Liu H, Zhang X, et al. Novel density-based clustering algorithms for uncertain data. Twenty-Eighth AAAI Conference on Artificial Intelligence, Hilton Québec; 2014. p. 2191–2197.

[9] MacQueen J. Some methods for classification and analysis of multivariate observations. In Proc. 5th Berkeley Symposium on Math. Stat. and Prob.; 1967. p. 281–297.

[10] Ichino M, Yaguchi H. Generalized Minkowski metrics for mixed feature type data analysis. IEEE TSMC. 1994;24(4):698V–708. doi:10.1109/21.286391

[11] Xiao L, Hung E. An efficient distance calculation method for uncertain objects. Computational Intelligence and Data Mining, CIDM 2007; 2007. p. 10–17.

[12] Chau M, Cheng R, Kao B, et al. Uncertain data mining: an example in clustering location data. In PAKDD, Singapore; 2006 Apr 9–12. p. 199–204. doi:10.1007/11731139_24

[13] Kao B, Lee SD, Lee FKF, et al. Clustering uncertain data using Voronoi diagrams and R-tree index. Knowl Data Eng IEEE Trans. 2010:1219–1233.

[14] Lukić I, Slavek N, Köhler M. The segmentation of data set area method in the clustering of uncertain data. Proceedings of the Jubilee 35th International ICT Convention – MIPRO; 2012. p. 420–425.

[15] Martinović G, Krpić Z, Rimac-Drlje S. Parallelization programming techniques: benefits and drawbacks. Cloud Computing 2010: The First International Conference on Cloud Computing, and Virtualization, 2010. doi:10.1.1.681.9034

[16] Bondhugula U, Baskaran M, Hartono A, et al. Towards effective automatic parallelization for multi-core systems. Proc. 22nd IEEE Int. Symp. Parallel and Distributed Processing, USA; 2008. p. 1–5.

[17] Hocenski Ž, Matić T. Acceleration of ceramic tiles machine vision quality control algorithm using CUDA. Proceedings of SICE Annual Conference, Taipei, SICE; 2010. p. 2170–2174.

[18] Matić T, Hocenski Ž. Parallel processing with CUDA in ceramic tiles classification. Knowledge-Based and Intelligent Information and Engineering Systems 14th International Conference, KES 2010, Cardiff, UK; 2010. p. 300–310.

[19] Sun X-H, Chen Y. Reevaluating Amdahl's law in the multi-core era. J Parallel Distrib Comput 2010;70: 183–188. doi:10.1016/j.jpdc.2009.05.002

[20] Zhou JJ, Chen L, Chen CLP, et al. Uncertain data clustering in distributed peer-to-peer networks. IEEE Trans Neural Netw Learn Syst. 2017. doi:10.1109/TNNLS.2017.2677093

[21] Zufle A, Emrich T, Schmid KA, et al. Representative clustering of uncertain data. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2014. p. 243–252.

[22] Jin P, Qu S, Zong Y, et al. CUDAP: a novel clustering algorithm for uncertain data based on approximate backbone. J Softw. 2014;9(3):732–737.