Peter Benjamin Volk, Frank Rosenthal, Martin Hahmann, Dirk Habich, Wolfgang Lehner

Clustering Uncertain Data with Possible Worlds

Erstveröffentlichung in / First published in:

IEEE 25th International Conference on Data Engineering. Shanghai, 29.03.-02.04.2009. IEEE, S. 1625-1632. ISBN 978-1-4244-3422-0

DOI: https://doi.org/10.1109/ICDE.2009.174

Final edited form was published in "IEEE 25th International Conference on Data Engineering. Shanghai, 2009". S. 1625-1632. ISBN 978-1-4244-3422-0
https://doi.org/10.1109/ICDE.2009.174

Clustering Uncertain Data With Possible Worlds

Peter Benjamin Volk, Frank Rosenthal, Martin Hahmann, Dirk Habich, Wolfgang Lehner
Technische Universität Dresden
Database Technology Group
01062 Dresden, Germany
dbinfo@mail.inf.tu-dresden.de

Abstract

The topic of managing uncertain data has been explored in many ways. Different methodologies for data storage and query processing have been proposed. As the availability of management systems grows, the research on analytics of uncertain data is gaining in importance. Similar to the challenges faced in the field of data management, algorithms for uncertain data mining also have a high performance degradation compared to their certain algorithms. To overcome the problem of performance degradation, the MCDB approach was developed for uncertain data management based on the possible world scenario. As this methodology shows significant performance and scalability enhancement, we adopt this method for the field of mining on uncertain data. In this paper, we introduce a clustering methodology for uncertain data and illustrate current issues with this approach within the field of clustering uncertain data.

1 Introduction

Database systems offer many possibilities to optimize data mining algorithms [5, 20], such as indexes for nearest neighborhood queries [19] or specialized mining operators [12, 18]. These functionalities are increasingly explored by several data mining vendors by implementing numerous algorithms. With this increasingly tight integration of database systems and data mining algorithms, the efficiency and system leverage of both algorithms and database systems increase.

In the field of database systems, current research work is being performed in the field of uncertain data. While traditional database systems only consider data as exact values, these newly created systems utilize a data model much closer to the real world. In general, they focus on the integration of the uncertainty into their data model. In almost every scenario where database systems are used, uncertainty definitions can be derived for the data. Integration systems, for example, create uncertain values once the actual values cannot be determined or an exact result must be approximated; e.g. the integration of addresses from heterogeneous systems [7]. Differently written addresses might refer to the same location. This difference could be expressed by a similarity probability. A second example: If values are rounded, the values become uncertain. This comes from the fact that the exact value is, e.g. 2.45 and the rounded value is 2. If the rounded value is used for further computation, the actual range must also be taken into consideration. In this example, the unrounded value of 2 is in the range between 1.5 and 2.49.

Fundamentally, uncertain data has been explored in many applications. A traditional field where uncertainty is evident, such as sensor data, is joined by many topics from computer science and data analytics. For example, [4, 10] explore data mining for continuous uncertainty. Continuous uncertainty is a property of the uncertainty region. Here, a continuous probability density function is set around a data point defining the appearance probability of moving objects or sensor data. Besides continuous uncertainty, the field of uncertain discrete values has been attracting further research interest in many ways. Agrawal et al. [2] work only on discrete uncertainty; meaning that tuple instances are assigned an appearance probability. With a high manifold of various application areas for uncertainty, different uncertainty models have been defined.

Most recent research in the field of data mining is performed with a focus on clustering uncertain points. Here, the main goal is to include the uncertainty information of the data into the clustering algorithms. The common assumption is: if the uncertainty information is explicitly used in the clustering, then the resulting partitioning has a higher quality than without this additional information. The validity of this assumption has been proven in various papers, such as [10, 15].

As research emerges on mining uncertain data, the database community develops more efficient models to handle uncertain data. In this field, a high diversity of system

Final edited form was published in "IEEE 25th International Conference on Data Engineering. Shanghai, 2009". S. 1625-1632. ISBN 978-1-4244-3422-0
https://doi.org/10.1109/ICDE.2009.174

capabilities can be found, such as the ability to store specific uncertainty definition types and its flexibility to include new models of uncertainty. In those systems the most effort is spent on optimizing the performance of answering queries on uncertain data [2, 3]. In contrast to those systems, the approach presented in [13] introduces a greater level of flexibility, scalability and performance. The approach is based on the monte-carlo principle and the system is called monte-carlo database system (MCDB).

While different approaches for uncertain data management have been proposed and much work has been done on uncertain data mining, no literature can be found in combining these two principles. To tackle this problem we propose a new method to cluster uncertain data on the base of the previously motioned MCDB approach in this paper. By having the same computational principle in the mining algorithm and the database system, the already established tight integration of data mining algorithms and database system is continued for the uncertain data field.

Contribution and Structure of This Paper:

This paper is organized as follows: First—in Section 2—we give an overview of current uncertain clustering algorithms and review them specifically under the consideration of the used distance functions. Then, we investigate the algorithms in regard to their ability to work with different distance functions. Furthermore, we review work done in the field of uncertain data management. Second—Section 3—, we propose our new clustering approach based on the MCDB approach [13]. Fundamentally, the major advantage of our clustering approach is that we can utilize any common clustering algorithm in combination with any arbitrary distance function. Third—Section 4—, we illustrate (1) optimization possibilities, (2) test our proposed clustering approach and (3) present future work aspects. Section 5 presents our evaluation, based on performance and quality. Finally Section 6 will draw a conclusion of all chapters conclusion.

2 Related Work

In this section, we illustrate related work in two fields: we review (1) the field of uncertain data stores as well as (2) the field of clustering uncertain data. Both fields are relevant since our newly proposed method aims to be integrated as closely as possible into an existing database system.

2.1 Probabilistic Storage

Database systems for uncertain data have the focus of storing, managing and querying data annotated with uncertainty. Many different methods have been proposed to optimize query processing and storage. The main concepts of the different systems are proposed in [2, 3, 21]. While the common base is the storage of uncertain data and query methods on this data, the detailed concepts differ very greatly. Fundamentally, the main difference between the approaches lies in the way the uncertainty definition is stored and the capability of storing different types of uncertainty. For example, the Trio system [2] can only process nominal uncertainty in various forms, while the Mystiq [3] and the Orion system [21] use a more complete model since the uncertainty is stored more flexible.

Orion, introduced in [21] is an extension of the open source database system PostgreSQL. Each tuple in a table is annotated with its uncertainty definition by extending the relational table model by additional information for the uncertainty definition. With this extension, new methods were developed to answer queries on uncertain data. One of the main ideas is to store the uncertainty definition on a symbolic base. This means that a density function is represented by a string (e.g. GAUS) with the parameters needed for this one specific data point. By this means, the amount of storage needed for the uncertainty definition is reduced compared to an exact definition of the functional dependencies of the uncertainty. Furthermore, Orion contains specially developed index structures to speed up the query processing. To answer queries, Orion stores a histogram representation of the PDF of a tuple once it is used for query processing. Therefore, the amount of memory needed to answer complex queries can become very large. Furthermore, the uncertainty model can be extended by implementing stored procedures. These customized uncertainty definitions are handled differently than the built-in definitions. Therefore, the optimized query processing can be applied to only a small extent on the custom defined uncertainty definition.

A different approach is chosen by Trio [2]. The foundation of the query processing lies in tracking the lineage of the tuples during the query process. With the computation model in Trio it is possible to calculate the uncertainty of intermediate and final result with high precision. This results in a major drawback of high memory usage during processing time. Furthermore, the current published results show only the possibility of storing nominal values with its possible worlds. With the limitation of being able to store only nominal data, this system is not applicable to all use cases and has only a very small usage base.

Compared to the approaches highlighted above, the Monte Carlo DataBase (MCDB) approach from [13] proposes a completely different approach. The main concept is to create different possible worlds of the uncertain data, via value generator functions (VG-Functions), which are then processed independently of each other. In a last step, the data is aggregated to a final result. Compared to other approaches, the MCDB approach does not alter the traditional

Final edited form was published in "IEEE 25th International Conference on Data Engineering. Shanghai, 2009". S. 1625-1632. ISBN 978-1-4244-3422-0
https://doi.org/10.1109/ICDE.2009.174

table model but leverages different stages within the query execution plan to perform the probabilistic queries. Furthermore, with the help of the VG-Functions, the MCDB approach handles a very broad range of uncertainty classes. The MCDB approach is, to our knowledge, the most flexible and efficient way of managing and querying uncertain data. For this reason, we use the basic concept of MCDB and transfer it into a clustering approach for uncertain data.

2.2 Uncertain clustering

Related work in clustering uncertain data can be classified in four different groups. The classification is based on the underlying base clustering method. The first class, proposed in [10, 15, 24], is density-based algorithms. In this case, clusters consist of dense regions and the number of clusters is not predetermined. In [1, 4], uncertain clustering algorithms are introduced based on k-means [11] (second class). Here the goal is to minimize the total expected distance between the points within a cluster. Conceptual clustering is proposed in [22] (third class). While other algorithms use a more general approach within the density function, conceptual clustering limits the uncertainty to categorical data. With this side condition the categorical clustering performs better compared to other approaches. The last class is based on hierarchical clustering and can be found in [16]. Further publications, such as [6, 14, 17], introduce techniques to enhance the clustering performance but do not introduce a novel concept of clustering or distance functions.

While the various approaches differ in detailed aspects, the common base is the property of clustering on data that is annotated with uncertainty. This annotation is constructed by a probability density function (PDF). Within the algorithms, the PDF is integrated into well-known distance measures. For example, in [4] the expected distance E from a point x_i to a cluster C_j mean c_j is defined as:

$$E(c_j, x) = \sum_{j=1}^{k} \sum_{i \in C_j} \int \|c_j - x_i\|^2 \, PDF(x_i) dx_i \quad (1)$$

and the cluster mean c_j of cluster j defined as

$$c_j = \frac{1}{C_j} \sum_{i \in C_j} \int x_i PDF(x_i) dx_i \quad (2)$$

Equations 1 and 2 both contain integral equations that have to be approximated. These approximations consume the majority of the computation time of the distance computation as depicted in [1].

While k-means-based algorithms minimize the expected total distance within one cluster, the density-based approaches modify the distance function and the associated definitions of distance, reachability and the property of a core object as defined by $DBSCAN$ in [8]. First, the definition of a distance is transformed to a distance density function:

$$P(a \leq d(o, o') \leq b) = \int_a^b PDF_d(o, o')(x) dx \quad (3)$$

P denotes the probability that the distance between two objects, $d(o, o')$, lies between a and b and $PDF_d(o, o')$ is the probability distribution function of the distance function between the two points. Second, on this base the reachability and core object definitions are redefined as reachability probability and core object probability. To be able to cluster data the properties of each point, within the dataset D, have to be checked at least against the property of density reachability from $DBSCAN$. $DBSCAN$ also introduces two parameters for the density property, with ϵ defining the range around a point and μ defining the number of points within the ϵ range needed to classify the point as a cluster. The reachability property of two points p and o from a database D with respect to the density parameters μ and ϵ is defined as:

$$P^{reach}_{\epsilon,\mu,d,D}(p, o) = P^{core}_{\epsilon,\mu-1,d,D\setminus\{p\}}(o) \cdot P_d(p, o)(\epsilon) \quad (4)$$

The core object probability is defined as:

$$P^{core}_{\epsilon,\mu,d,D}(o) = \sum_{\substack{A \subseteq D \\ |A| \geq \mu}} \prod_{p \in A} P_d(p, o)(\epsilon) \prod_{p' \in D \setminus A} (1 - P_d(p', o)(\epsilon)) \quad (5)$$

Clearly eqations 3, 4 and 5 are of complex nature because of the operations on the PDF and the high number of data subsets in equation 5. Furthermore, $FDBSCAN$ proposes to solve the integrals by approximation methods on the function. Therefore, $FDBSCAN$ cannot scale appropriately in terms of number of data points within the data set D.

The approach proposed in [10], named $DBSCAN^{EA}$, introduces means for clustering the data on the base of objects with a specific bound. As the previous examples show, the PDF itself is abstractly defined as a function. Since the PDF does not necessarily need to have a zero-point, the method cannot be applied to all PDF functions. For example, a commonly used PDF is the normal distribution function. The normal distribution does not have a zero-point. Therefore, a determination of a zero point is necessary. Furthermore, $DBSCAN^{EA}$ requires calculating the minimum and maximum distances between two arbitrary shapes. This is solved by sampling the objects and calculating the distance between the sampling points. From this calculation the minimum and maximum is determined. Because the mesh of the sampling points on the objects needs

Final edited form was published in "IEEE 25th International Conference on Data Engineering. Shanghai, 2009". S. 1625-1632. ISBN 978-1-4244-3422-0
https://doi.org/10.1109/ICDE.2009.174

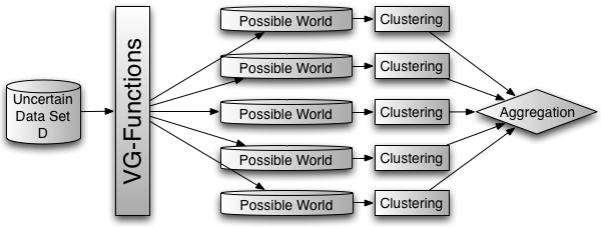

Figure 1. Possible Worlds Mining

to be very dense to approximate the minimum and maximum distances accordingly, the distance computation time increases accordingly.

The approaches illustrated in [1, 22, 24] work on similar complex distance functions and therefore have the similar shortcomings in scalability towards large datasets.

3 Possible World Clustering

In this section we introduce our new approach based on possible world scenario from the MCDB database. The complete process is illustrated in Figure 1. From a database D with N_u tuples annotated with uncertainty definitions, we generate M possible worlds, with each world containing N_u tuples. To generate the possible worlds we leverage the value generator functions (VG-Function) from [13]. A VG-Function uses the uncertainty definition from D to generate random values derived from the tuple. Since the VG-Functions generate the new tuples on the base of the PDF, the union of all M can be seen as a representation of the PDFs of the original points from D.

After M possible worlds are generated, the next step is to perform the data analysis on each possible world. As a result each possible world has its own cluster model. Each cluster model represents only a local model to its possible world. Hence, we generate M possible cluster models. As the VG-Functions generate independent possible worlds, each world can be processed parallel to other worlds. Hence this method achieves a high parallelism. With multi-core and multi-CPU systems, this method can achieve a high speedup for the complete process compared to the classical uncertain clustering approaches. Since the goal is to generate a global cluster model for the database D, the local cluster models have to be merged. The MCDB approach, which we use as a design reference, uses the single values of the possible worlds and generates histograms from the results.

The final clustering result must be derived from the local cluster models in some way. For this purpose we use the method of clustering aggregation and pair-wise similarity, as introduced in [9]. At first we build M local similarity graphs. A local similarity graph is constructed as follows: Each point P_i is a node. All points within the same cluster in a local cluster model are connected with an edge weighted with 1. As a result we have derived M similarity graphs from the M local cluster models. In the next step the local similarity graphs are aggregated to one global similarity graph using the techniques introduced in [9]. The resulting global similarity graph can now be used to derive a global cluster model.

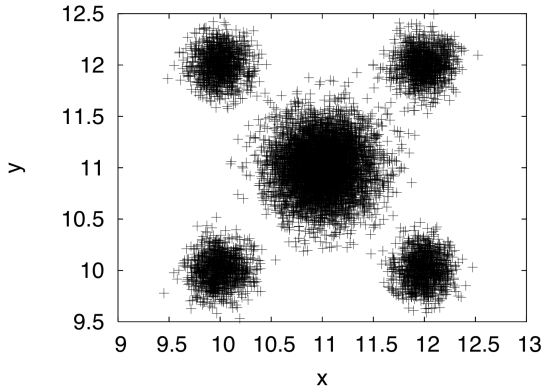

Figure 2. Dataset used for evaluation

Although this approach is a more extensive algorithm than current approaches for uncertain clustering it offers a more scalable approach as the following section shows.

4 Possible Optimizations

As the proposed new algorithm is of complex nature wen now illustrate an optimization strategy for the algorithm and its impact on the clustering result. Furthermore, we will present further research interest within this topic. For all evaluations in this section we use a data set as illustrated in Figure 2. It consists of 5 Gaussian clusters. We vary the number of data points from 10'000 to 50'000. As an uncertainty definition we use a rectangular function with a maximum error of 5% of the coordinate in the 3σ area of the PDF. Furthermore, we use DBSCAN from [8] to generate the local cluster models.

The usage of the previously introduced algorithm is connected with the usage of parallel components. Since it is necessary to produce many possible worlds to approximate the PDF properly, the number of parallel resources required can exceed current CPU technology for parallel processing. Therefore, multiple possible worlds have to be executed in a serial fashion thus reducing the speed of the algorithm tremendously. To tackle this problem we illustrate an op-

Final edited form was published in "IEEE 25th International Conference on Data Engineering. Shanghai, 2009". S. 1625-1632. ISBN 978-1-4244-3422-0
https://doi.org/10.1109/ICDE.2009.174

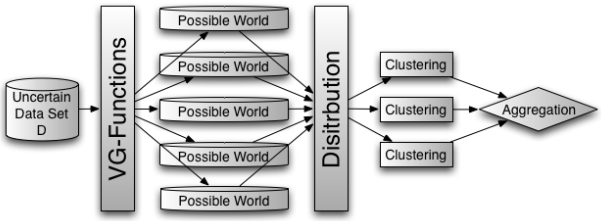

Figure 3. Reduction of Possible Worlds

timization approach in this section. The optimization analyses the effect of the distribution of the possible worlds to a reduced number of threads. With a very low number of threads the total runtime of the algorithm can further be improved since they can be executed fully in parallel.

4.1 Possible World Distribution

As illustrated in the previous section, the VG-Functions generate M possible worlds. Since the possible worlds are independent of each other, we can perform the clustering of the worlds in parallel. Therefore, we distribute each possible world on one thread. This produces the need of $P = M$ threads. Parallelism on modern architecture is limited in many ways. If P exceeds the number of available parallel processing units then not all M possible worlds can be processed in parallel. Therefore, parallel processes are executed sequentially, reducing the clustering speed.

To ensure that all possible worlds are executed in parallel, we propose an extension to the possible world clustering called possible world distribution. After the VG-Functions generate M possible worlds we distribute the worlds on P processes with $P < M$. To perform a reduction from M possible worlds to P independent datasets, as illustrated in Figure 3, a few side conditions must be considered and also their effect on the resulting clustering.

At first, we generate M possible worlds for one data point U from D. Then, we create a histogram with N_b number of bins from the values. Figure 4 illustrates a histogram of a normal distributed function around the value μ. Each bin contains the same number of data points. Therefore, the width of the bin is of interest rather than the occurrence frequency. This property of the bins is used later in the process to generate the data sets for the processes. For the number of generated possible worlds we consider the side condition that the number M is an integer multiple of P. This guarantees that each process becomes the same number of data points. Furthermore we use this side condition to consider that each process P_i has the same number of possible worlds from one data point U. This is important for the aggregation process, as we present in more detail later on.

To compare the actual distribution methods with each

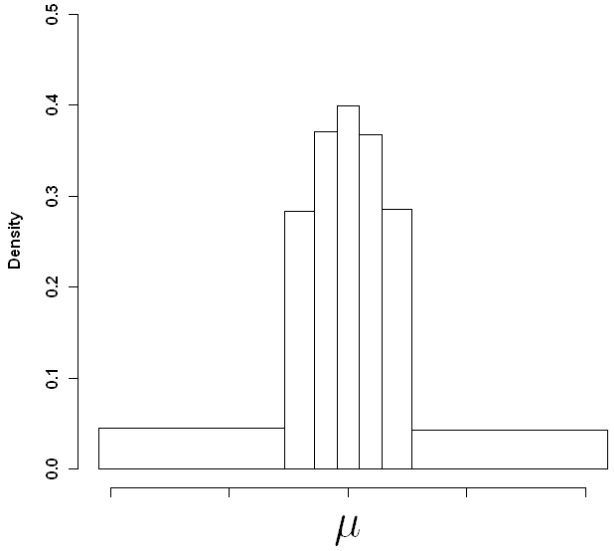

Figure 4. Histogram of the generated possible worlds

other we measure the similarity of the cluster models. As a similarity measure we use the pair-wise distance measure from [9]. At first, the dissimilarity of two points (u and v in D) in two cluster models(C_1 and C_2) is calculated with the dissimilarity measure:

$$d_{v,u}(C_1, C_2) = \begin{cases} 1 & \text{if } C_1(u) = C_2(v) \text{and} C_1(u) \neq C_2(v) \\ & \text{or } C_1(u) \neq C_2(v) \text{and} C_1(u) = C_2(v) \\ 0 & \text{else} \end{cases} \tag{6}$$

The total distance S of two cluster models is then calculated by the sum of all dissimilarities of all tuple pairs in D:

$$S_D(C_1, C_2) = \sum_{(u,v) \in D \times D} d_{v,u}(C_1, C_2) \tag{7}$$

The maximum distance of two cluster models is equal to the number of data points within the cluster model. Furthermore, we generate 30 possible worlds and distribute them to 5 processes.

For the actual distribution we consider two different methods. The first method is the most obvious: The random distribution of the possible worlds on the processes. Each process is assigned N_u number of possible worlds with

$$N_u = \frac{M}{P} \tag{8}$$

With this random distribution of possible worlds on the processes the resulting cluster models are very similar to each other since they are a good mix of generated values. Figure 5 illustrates the differences of the local cluster models

Final edited form was published in "IEEE 25th International Conference on Data Engineering. Shanghai, 2009". S. 1625-1632. ISBN 978-1-4244-3422-0
https://doi.org/10.1109/ICDE.2009.174

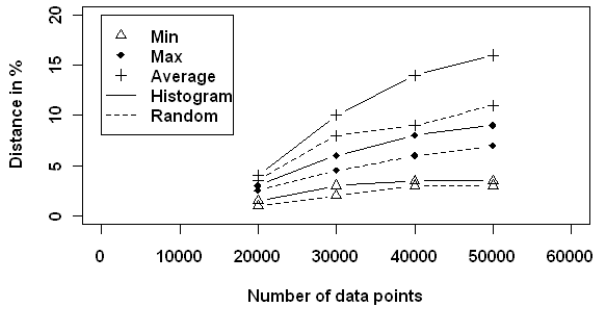

Figure 5. Normalised local cluster model similarities

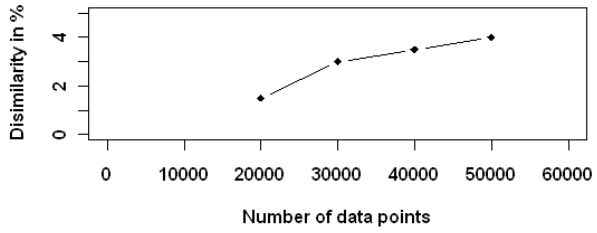

Figure 6. Disimilarity of final cluster models

for a random mix of worlds. Clearly the differences of the models are only very limited.

The second method is to leverage the histogram of the generated possible worlds. We generate $N_b = N_u$ bins and assign each bin to exactly one process. Figure 5 shows the normalized distance of the local cluster models with different dataset sizes. Furthermore, we normalized the distances by the maximum distance to create a more comprehensible evaluation. This way the local cluster models have a high diversity of points compared with each other, which is also reflected in the differences in the local cluster model, as Figure 5 shows.

Next, we compare the two final clustering results generated with the two distribution methods. Figure 6 shows the measured results for different dataset sizes. Clearly Figure 6 shows that the more data points within the dataset are, the higher the difference between the two methods is. Therefore, it shows that the selection of the distribution is an important decision to ensure high cluster quality.

Finally, we compare the clustering performance between the distributed and the undistributed method. Table 1 shows that for different dataset sizes the distribution of the possible worlds to lower number of processes increases the clustering performance. Therefore, the distribution is a good method to increase the clustering performance. Furthermore, an increase of possible worlds increases the quality of taking the uncertainty into consideration, since the PDF is approximated with a higher accuracy.

Distribution Method	10'000	20'000	30'000
None	1911s	7652s	24898s
Random	321s	1264s	4898s
Histogramm	345s	1302s	4965s

Table 1. Clustering time with different distribution models

4.2 Further Issues

As the above optimization reduces the number of sequential operations still, further research interest exists on this topic. At first, the question arises for the optimal M. With an increasing number of generated possible worlds, the precision of the system increases. This is possible since the base PDF is approximated with a higher precision than with a lower M. Surely the optimal M strongly depends on the PDF itself. The number of possible worlds needed to approximate a normal distribution is obviously higher than to approximate an equally distributed PDF. Therefore, an extensive analysis of the PDF under consideration with the selected algorithm is necessary. Furthermore, if we allow different number of possible worlds per PDF then it is not possible to evaluate the possible worlds separately from each other since not every process contains a representation of U from D.

By recalling the distribution models, as illustrated above, there are also possible additional optimizations for the aggregation method. Since the distribution is performed on histograms from the PDF, each bin of the histogram can be assigned a weight according to its position within the histogram. With a modification of the construction of the similarity graph it is possible to integrate this weight into the similarity graph. In this way the graph also is created with respect to the PDF.

5 Evaluation

In this section we compare our approach with existing approaches. Currently no clustering performance measurements for uncertain data are known. Therefore, we compare our approach only against existing methods. At first, we compare the time needed to cluster data (performance) with the other approaches and show that our new approach scales with an increasing number of data points as well as number of possible worlds. Furthermore, we compare the results with existing algorithms such as $FDBSCAN$ from [15] and $DBSCAN^{EA}$. For all experiments we use the dataset illustrated in Section 4. All experiments are executed on a quad core xeon IBM blade providing a physical parallelization of a factor 4.

Final edited form was published in "IEEE 25th International Conference on Data Engineering. Shanghai, 2009". S. 1625-1632. ISBN 978-1-4244-3422-0
https://doi.org/10.1109/ICDE.2009.174

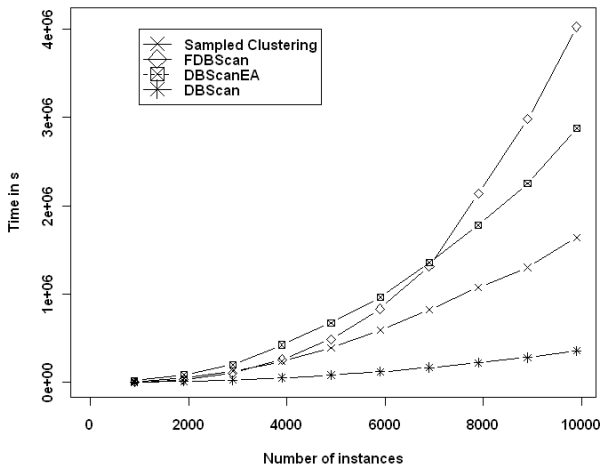

Figure 7. Comparison of clustering times

5.1 Performance

Our new approach offers the possibility of creating the local clustering models in parallel. With this parallelism we can create the final clustering result faster compared to other uncertain clustering algorithms. At first we performed performance tests with $DBSCAN$, $FDBSCAN$, $DBSCAN^{EA}$ and our new approach. Figure 7 shows the results of our experiments. $DBSCAN$ clearly is the algorithm with the shortest time needed to create the final cluster models, since the uncertain algorithms $FDBSCAN$, $DBSCAN^{EA}$ and possible world clustering all contain $DBSCAN$ as a base. All modifications of the algorithm to include the uncertainty into $DBSCAN$ produce the specific algorithm overhead, compared to $DBSCAN$.

Figure 7 shows that $FDBSCAN$ has a short run time for a small number of data points. This comes from the small overhead added by the distance function. The distance function is performed n^2 times to create each clustering. Since $FDBSCAN$ adds a very costly operation to the distance function, the costs for this modification are exponentially added. Therefore, $FDBSCAN$ does not scale very well for larger number of data points compared to $DBSCAN$.

$DBSCAN^{EA}$ also adds a specific overhead to the distance function. Since we choose the PDF function as a rectangular function, the min-/max- distance calculation adds only very limited overhead. Therefore, the increase of time needed for the clustering also increases. Since the overhead is not as large as with $FDBSCAN$, the total time for $DBSCAN^{EA}$ does not increase as fast as for $FDBSCAN$.

Finally, our Possible World clustering performs best from the uncertain clustering algorithms. The overhead in the Possible World clustering is derived from two points: (i) possible world generation and distribution on the threads, (ii) cluster aggregation. Figure 7 shows that for a small number of tuples the management overhead is larger than the overhead produced by $FDBSCAN$. Since the overhead is not produced during the distance computation, it does not grow exponentially as the overhead from $FDBSCAN$ does. Therefore, the Possible World clustering performs better for larger data sets than $FDBSCAN$.

5.2 Quality

Clustering quality is an important measure in comparing new algorithms with existing methods. Typically, a prelabled dataset is used to determine how good the clustering algorithms can determine the labels. A second approach is to use quality metrics such as the Xi-Beni index introduced in [23]. Since no clustering metrics and prelabled datasets are known for uncertain data, we compare our results only with the results from existing algorithms. The goal is to have similar results as $FDBSCAN$ produces since our focus lies on creating a method to cluster data faster compared to other approaches.

Similarity	$Sampling$	$DBSCAN^{EA}$	$FDBSCAN$
$Sampling$	0	90%	69%
$DBSCAN^{EA}$	90%	0%	80%
$FDBSCAN$	69%	80%	0%

Table 2. Clustering Difference.

Table 2 shows the normalized similarity of the final cluster models to each other. The parameters for all clustering algorithms were kept constant. Also in this experiment we generated 4 possible worlds. Hence, we choose a very low approximation of the PDF. Clearly, the error introduced by this small number of possible worlds is only minimal compared to the other algorithms.

6 Conclusion

In this paper we presented a novel clustering technique for uncertain data. As more algorithms are included into database systems, we propose a method similar to the Monte Carlo Database for uncertain data to be able to include the algorithm into the database management system. The method is divided into three steps. In the first step, multiple possible worlds are generated from a dataset according to their uncertainty definition. In the second step, a cluster model is built for each world. For a final clustering the local clustering, results are aggregated into one clustering.

Final edited form was published in "IEEE 25th International Conference on Data Engineering. Shanghai, 2009". S. 1625-1632. ISBN 978-1-4244-3422-0
https://doi.org/10.1109/ICDE.2009.174

Furthermore, we illustrate possible extensions to the method. With these extensions the algorithm may achieve a higher clustering quality than without the extensions. Also we can show that with the extension we can leverage modern multicore CPU architectures more efficiently than with the naïve method.

References

[1] C. C. Aggarwal and P. S. Yu. A framework for clustering uncertain data streams. In *ICDE*, pages 150–159. IEEE, 2008.

[2] P. Agrawal, O. Benjelloun, A. D. Sarma, C. Hayworth, S. U. Nabar, T. Sugihara, and J. Widom. Trio: A system for data, uncertainty, and lineage. In U. Dayal, K.-Y. Whang, D. B. Lomet, G. Alonso, G. M. Lohman, M. L. Kersten, S. K. Cha, and Y.-K. Kim, editors, *VLDB*, pages 1151–1154. ACM, 2006.

[3] J. Boulos, N. Dalvi, B. Mandhani, S. Mathur, C. Re, and D. Suciu. Mystiq: a system for finding more answers by using probabilities. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 891–893, New York, NY, USA, 2005. ACM.

[4] M. Chau, R. Cheng, B. Kao, and J. Ng. Uncertain data mining: An example in clustering location data. In W. K. Ng, M. Kitsuregawa, J. Li, and K. Chang, editors, *PAKDD*, volume 3918 of *Lecture Notes in Computer Science*, pages 199–204. Springer, 2006.

[5] M.-S. Chen, J. Hun, and P. S. Yu. Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, 8:866–883, 1996.

[6] G. Cormode and A. McGregor. Approximation algorithms for clustering uncertain data. In *PODS '08: Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 191–200, New York, NY, USA, 2008. ACM.

[7] X. L. Dong, A. Y. Halevy, and C. Yu. Data integration with uncertainty. In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB 2007, Vienna, Austria, September 23-27)*, pages 687–698, 2007.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, Oregon, 1996. AAAI Press.

[9] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *ACM Trans. Knowl. Discov. Data*, 1(1):4, 2007.

[10] D. Habich, P. B. Volk, W. Lehner, R. Dittmann, and C. Utzny. Error-aware density-based clustering of imprecise measurement values. In *ICDMW '07: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, pages 471–476, Washington, DC, USA, 2007. IEEE Computer Society.

[11] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[12] A. Hinneburg, W. Lehner, and D. Habich. Combi-operator: Database support for data mining applications. In *Proceedings of 29th International Conference on Very Large Data Bases (VLDB 2003, September 9-12, Berlin, Germany)*, pages 429–439, 2003.

[13] R. Jampani, F. Xu, M. Wu, L. L. Perez, C. M. Jermaine, and P. J. Haas. Mcdb: a monte carlo approach to managing uncertain data. In J. T.-L. Wang, editor, *SIGMOD Conference*, pages 687–700. ACM, 2008.

[14] B. Kao, R. Cheng, and S. Lee. Reducing uk-means to k-means. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, pages 483–488. Dune, 2007.

[15] H.-P. Kriegel and M. Pfeifle. Density-based clustering of uncertain data. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 672–677, New York, NY, USA, 2005. ACM.

[16] H.-P. Kriegel and M. Pfeifle. Hierarchical density-based clustering of uncertain data. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 689–692, Washington, DC, USA, 2005. IEEE Computer Society.

[17] W. K. Ngai, B. Kao, C. K. Chui, R. Cheng, M. Chau, and K. Y. Yip. Efficient clustering of uncertain data. In *Proceedings of the Sixth International Conference on Data Mining (ICDM'06)*, 2006.

[18] C. Ordonez. Building statistical models and scoring with udfs. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2007, Beijing, China, June 12-14)*, pages 1005–1016, 2007.

[19] C. Ordonez and P. Cereghini. Sqlem: Fast clustering in sql using the em algorithm. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD 2000, May 16-18, Dallas, Texas, USA)*, pages 559–570, 2000.

[20] S. Sarawagi, S. Thomas, and R. Agrawal. Integrating mining with relational database systems: Alternatives and implications. In *Proceedings ACM SIGMOD International Conference on Management of Data (SIGMOD 1998, June 2-4, Seattle, Washington, USA)*, pages 343–354, 1998.

[21] S. Singh, C. Mayfield, S. Mittal, S. Prabhakar, S. Hambrusch, and R. Shah. Orion 2.0: native support for uncertain data. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1239–1242, New York, NY, USA, 2008. ACM.

[22] Y. Xia and B. Xi. Conceptual clustering categorical data with uncertainty. In *ICTAI (1)*, pages 329–336. IEEE Computer Society, 2007.

[23] X. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):841–847, 1991.

[24] Z. Yu and H.-S. Wong. Mining uncertain data in low-dimensional subspace. In *ICPR (2)*, pages 748–751. IEEE Computer Society, 2006.