The Hong Kong Polytechnic University

Department of Computing

Clustering and Classification on Uncertain Data

Lei Xu

A thesis submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

April 2013

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

____________________(Signed)

Lei Xu (Name of student)

Abstract

We study the problem of mining on uncertain objects whose locations are uncertain and described by probability density functions (pdf). Clustering and classification are two important tasks in data mining.

Clustering on uncertain objects is different from traditional case on certain objects. UK-means is proposed based on K-means but it is time consuming. Pruning techniques are proposed to improve the efficiency of UK-means. First we analyze existing pruning algorithms and experimentally show that there exists a new bottleneck in the performance due to the overhead of pruning candidate clusters for assignment of each uncertain object in each iteration. In this thesis, we will show that by considering squared Euclidean distance, UK-means (without pruning techniques) is reduced to K-means and performs much faster than pruning algorithms, however, with some discrepancies in the clustering results due to different distance functions used. Thus, we propose Approximate UK-means to heuristically identify objects of boundary cases and re-assign them to better clusters. In addition, we propose three models for the representation of cluster representative (certain model, uncertain model and heuristic model) to calculate expected squared Euclidean distance between objects and cluster representatives. The experimental results show that our approach (Approximate UK-means) reduces the discrepancies of K-means' clustering results by taking more time than K-means.

In the case of classification on uncertain objects, some existing algorithms are hundreds or thousands times more complex than traditional ones, because an uncertain object is represented by hundreds or thousands of samples. Due to the complex representation of uncertain objects and existing algorithms, it is time consuming to classify uncertain objects. In this thesis, we propose a novel supervised UK-means algorithm to classify uncertain objects more efficiently. In supervised UK-means, we consider to select features that can capture the relevant properties of uncertain data similarly to feature selection on certain objects. We also extend supervised UK-means to ensemble learning. We experimentally demonstrate that our proposed approaches are more efficient than existing algorithms and can attain comparatively accurate results on non-overlapping data sets.

In supervised UK-means, the classes are assumed to be well separated. But the real data are usually distributed arbitrarily and the classes cannot be separated by simple linear boundaries. We propose Supervised UK-means with Multiple Subclasses (SUMS) which considers that the objects in the same class can be further divided into several groups (subclasses) within the class and tries to learn the subclass representatives to classify objects more accurately. Moreover, we propose a Bounded Supervised UK-means with Multiple Subclasses (BSUMS) to avoid overfitting. From our experiments, Supervised UK-means with Multiple Subclasses (SUMS) and BSUMS perform better than supervised UK-means on synthetic data sets and real data sets.

Keywords: UK-means, uncertain objects, clustering, classification, expected distance

List of Publications

1. Edward Hung, Lei Xu and Chi-Cheong Szeto, "A Heuristic on Effective and Efficient Clustering on Uncertain Objects", In AI 2010: Advances in Artificial Intelligence - 23rd Australasian Joint Conference, Proceedings, pages 92C101, 2010.

2. Lei Xu and Edward Hung, "Distance-based Feature Selection on Classification of Uncertain Objects", In AI 2011: Advances in Artificial Intelligence - 24th Australasian Joint Conference, Proceedings, pages 172C181, 2011.

3. Lei Xu and Edward Hung, "Improving Classification Accuracy on Uncertain Data by Considering Multiple Subclasses". In AI 2012: Advances in Artificial Intelligence - 25th Australasian Joint Conference, Proceedings, pages 743C754, 2012.

4. Lei Xu and Edward Hung and Chi-Cheong Szeto, "A Heuristic on Effective and Efficient Clustering on Uncertain Objects". In Computational Intelligence. (under review)

5. Lei Xu and Edward Hung, "Ensemble UK-means model on Classification of Uncertain Objects". In Information Sciences. (under review)

6. Lei Xu and Edward Hung, "Improving Classification Accuracy on Uncertain Data by Considering Multiple Subclasses". In Expert System with Applications. (under review)

Acknowledgement

I am deeply grateful to my chief supervisor Dr. Edward Hung. His constant and gentle guidance has allowed me to enjoy the research I have done. He is always there whenever I needed help, and with his support and encouragement, I overcame the difficulties in this work. I am thankful for all his insightful comments and for the very thorough revision he has made of my articles and this thesis. I consider myself very fortunate to have the chance to learn from him.

I would like to thank my co-supervisor Prof. Korris Chung for his valuable and helpful suggestions to my research. His enthusiasm on work and quest for excellent research for perfection will always inspire me to devote myself fully to the work.

Many thanks to the faculty and support staff in Department of Computing at The Hong Kong Polytechnic University for their help over the years. I would also like to thank the help and kindness of many friends during my Ph.D. study. I have greatly enjoyed working with my colleagues: Dr. Yan Li, Chi-Cheong Szeto, Takazumi Matsumoto, Dongmin Guo, Xingzheng Wang, Feng Liu, Xiaofeng Qu, Jin Xie, Jingjing Li, Zhizhao Feng, and Bo Peng.

Finally, I cannot end without thanking my parents and my husband, on whose constant encouragement and love I have relied throughout the time of my study. To them I dedicate this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

While a great number of research have been focused on mining and queries on relational databases [91], the focus has been on databases that data represented by exact values. In many real-life applications, however, the raw data (for example, in the case of sensor data) are not precise or accurate when they were collected or produced due to the limitation of the underlying equipment or other reasons. There are many sources of uncertain data, for example, sensor readings, information extracted from input sources by using probabilistic parsing, results obtained by predictive softwares in stock market, and so on.

For example, some temperature sensors are installed to monitor the temperature of a location. The temperature values observed by the sensor are not accurate due to the limitation of equipments. The real value of temperature at a given time cannot be exactly known and is uncertain. Another example for uncertain data, in an online shopping system (hotel booking system), different customers may give different scores to a hotel. The rating of a hotel is uncertain.

Uncertain data can be represented by an exact value with margins of error, with or without a (density) probability distribution function (pdf). The result can also be in the form of a set of values or an interval, one of which maybe the real value. However, only exact values can be stored in traditional databases, so that uncertain data have to be transferred into exact value by using value with the highest frequency or calculating mean value or weighted average (for numerical attributes). The storage and mining in databases can become simpler when existing mining techniques and database systems are used. It is obvious that in the databases, the intermediate or final results from the mining tasks and queries will also be approximate or maybe wrong by approximating the uncertain source data values. For example, in clustering application, the locations of centroids of clusters may be deviated from the real ones, which can make data be assigned to wrong clusters. In the case of classification, the learned classifier may be different from the real ones, or some testing objects will be predicted by a wrong label.

In this thesis, we consider the problem of data mining applications (clustering and classification) on uncertain objects with multidimensional uncertainty where an object is represented by an uncertain region over which a discrete probability distribution function (PDF) or a probability density function (pdf) is defined. Formally, we consider a set of n objects o_i, $1 \leq i \leq n$ in an m-dimensional space. An object o_i is represented by a pdf f_i: $IR^m \rightarrow IR$ (IR represents real number space) that specifies the probability density of each possible location of object o_i. In this thesis, the applications discussed just require that for each object o_i, the uncertain region A_i of each object o_i is finite, (i.e. $\forall x \notin A_i, f_i(x) = 0$) without relying on any special forms of the pdf (f_i). Thus, a bounding box can be used to bound each object. In practice, the probability density of an object is just concentrate in a very small region, so that the assumption is convincing.

1.1 Overview of Uncertain Data Mining

There have been some researches in uncertain database [57]. Data uncertainty are mainly classified into two types. One is existential uncertainty caused by being uncertain about the existence of an object or a data tuple [12, 23, 30, 84]; The other is value uncertainty caused by not knowing the value precisely. Recently relationship uncertainty is proposed by Bin Jiang et al. in [56]. The task of summarizing the relationship uncertainty in [56] between objects is learning the order of the values on a dimension of the domain. For example, a traveler gives a higher score to a hotel whose location is close to the central, and a lower score to a hotel that is far from the central, which likely means that the user prefers the hotel near central. Learning the order of values can infer more knowledge of domain.

In this thesis, we focus on value uncertainty. In value uncertainty, the value of an object is not unique, and the samples are used to represent an object. Sometimes, an object is represented by hundreds or thousands samples, which makes the problem of mining uncertain data different from that of mining certain data. The uncertain data is value uncertain data in this thesis. The mining techniques handling uncertain data are different from those on certain data. Because the representation of uncertain data are more complex than certain data, traditional methods on certain data cannot be used directly.

Some applications on mining (i.e. clustering and classification) were proposed for value uncertain data. When clustering uncertain data, for example in UK-means, the similarity calculation (i.e. distance) between uncertain objects are more complex than that between exact objects in K-means compared with traditional techniques. UK-means [22] can be considered as a generalization of K-means. The only difference between UK-mean and K-mean is that expected distance is used for distance calculation in UK-means. In [62, 63], fuzzy distance is used to measure the distance between objects which is different from that used in traditional certain case.

A second application is classification. The aim of classification on certain objects is to predict the labels of objects with the minimum error loss. The uncertainty of objects may affect the results of classifier. In [17], the model of data is defined as a bounded geometric region. The key idea of [17] is to get an optimized probabilistic separation that are on the two sides of the boundary between the two classes. The classification algorithms on uncertain objects try to provide classifiers that can predict the labels of objects by the minimum probability error loss.

Other applications were also developed to handle uncertain objects. The methods are different and more complex than those on certain data, because uncertain information are taken into account. Thus, a large amount of challenges are faced on the field of uncertain data mining.

1.2 Challenges of Uncertain Data Mining

There are a number of special challenges on some fronts posed in the field of uncertain data mining. The challenges are mainly include two broad issues: model of uncertain data and the applications on uncertain data. Data management and mining application working on uncertain data are the examples of uncertain data applications. Some challenges are described in detail as followings:

The model of uncertain data

A number of models have been discussed in [2] and [86]. The proper model can capture the characteristic of data. The complexity of model directly affects the efficiency of database management and data mining algorithms. It is a challenge to estimate an appropriate model for uncertain data that can catch real uncertain information of data or close enough to its real model.

The metrics between uncertain data

A large amount of work are proposed to measure the distance between certain objects [18, 100, 101], but few can be directly used in uncertain data. The metrics for uncertain objects should consider the uncertain information of objects. The metrics can affect the efficiency of the applications on uncertain data. It is still a problem to measure the similarity between objects with considering uncertain information as well as keeping the calculation methods efficient.

The mining algorithms on uncertain data

The model of uncertain data and the operation on them are the two keys of uncertain data mining algorithms. It is certainly that developing mining algorithms on uncertain data is a challenging work. The algorithms should perform efficiently when mining uncertain objects by using complex uncertain model.

1.3 Contributions of the Thesis

There are a few challenges and open problems in the task of clustering and classification on uncertain objects.

Clustering on uncertain objects

Before clustering uncertain objects, we propose expected squared Euclidean distance to calculate the distance between objects efficiently. Different from traditional work on certain objects, we propose three models for the representation of cluster representative (certain model, uncertain model and heuristic model) to calculate expected squared Euclidean distance between objects and cluster representatives. Then we propose Approximate UK-means to heuristically identify objects of boundary cases and re-assign them to better clusters.

Classification on uncertain objects

We build supervised UK-means based on UK-means to classify uncertain objects. Considering the relevant properties of uncertain data, we extend supervised UK-means to feature selection and Adaboost, respectively.

On the other hand, in real applications, the data are usually distributed arbitrarily and the classes cannot be separated by simple linear boundaries. Thus, we consider the objects in the same class can be further divided into several groups (subclasses) within the class. We use PG-means (projected Gaussian) [45] to estimate the number of subclasses in a class and train the subclass representatives by UK-means.

1.4 Outline of the Thesis

The reminder of this thesis is organized as follows. In Chapter 2, we briefly discuss previous work that have been applied on uncertain data mining. In Chapter 3, Approximate UK-means is proposed to heuristically clustering uncertain objects efficiently. In Chapter 4, we present supervised UK-means for classifying uncertain objects, and extend supervised UK-means to feature selection and Adaboost. We overcome the limitation of supervised UK-means by further dividing classes into multiple subclasses in Chapter 5. Finally, we present our conclusions and discuss future work in Chapter 6.

Chapter 2

Literature Review

A large number of challenges are proposed to the field of uncertain data. The challenges can be divided into two broad issues: modeling the uncertain data and a variety of applications working with them, i.e. clustering, classification and other data mining tasks on uncertain data. In this chapter, we first introduce the models used for uncertain data in Section 2.1, then we review clustering on uncertain data in Section 2.2, classification on uncertain data in Section 2.3, and other data mining tasks on uncertain data in Section 2.4. Finally, we summarize this chapter in Section 2.5.

2.1 Modeling Uncertain Data

The problem of uncertain data model has been deeply studied in the literature [1, 43, 44, 53, 85]. The uncertain data model is represented in two way: probabilistic database [14, 49, 86] and uncertain data [24, 49, 92]. Both of the two models are the formalism of the "possible worlds model" [1, 31, 53, 54, 86].

A probabilistic database [14, 49, 86] contains a number of probabilistic tables. T is a probabilistic table that is represented by a set of uncertain tuples. In T, $P(t)$ is used to describe the appearing probability of t in T, t is the tuple in T. All the possible tuples are consistent with a given schema. It is important to note that the probabilistic table is represented by an exponential number of tuples.

A probability density function (pdf) is used to describe an uncertain object [24, 49, 92] in a domain **U**. Actually, we have no idea of the probability function. A number of samples or instances are collected or generated with assuming an approximating probability density function. The sum of the probabilities of an object is 1. Additionally, uncertain objects and probabilistic databases are equivalent and can be converted to each other without considering the dependency between objects in the discrete case [76]. On the other side, there are also uncertain models for semistructured and structured XML data [50, 51, 74, 96, 110].

2.2 Clustering

Clustering is a classic problem in real life [47, 48, 68, 69]. The task of clustering is that the objects assigned into the same group are more similar than objects from other classes [9, 68]. A large amount of clustering algorithms have been proposed. Clustering algorithms are summarized based on their models. Several typical cluster models have been proposed, i.e. connectivity models (hierarchical clustering [88]), centroid models (k-means) [47, 48, 69], distribution models [72], density models (DBSCAN and OPTICS [10, 36]), subspace model [6] and so on.

K-means and UK-means

K-means [69] is used to cluster certain objects. In K-means, if an object o is assigned to a cluster c, it means that c's representative is the closest one among all clusters to o based on Euclidean distance. UK-means [22] is a generalization of the traditional K-means algorithm to handle uncertain objects whose locations are represented by pdfs. The only difference between UK-means and K-means is that, between an object and a cluster representative, expected Euclidean distance is calculated instead of Euclidean distance in UK-means. For arbitrary pdfs, the bottleneck of UK-means is the calculation of expected distance, which are computationally expensive. Thus, pruning techniques were proposed to remove the candidate clusters from consideration, which are certainly not closest to an object, reducing a large amount of expected distance calculation.

2.2.1 Pruning Techniques

The basic idea of pruning techniques is using simple distance calculation to identify cluster representatives which could not be the closest one to a given uncertain object. Hence, the expected distance calculation between the object and those representatives can be skipped.

MinMax-BB

Each object o_i is bounded by a minimum bounding rectangle (MBR)[1] in MinMax-BB [73], outside which the object has zero (or negligible) probability of occurrence. The minimum distance ($MinDist_{i,j}$) and the maximum distance ($MaxDist_{i,j}$) are calculated to prune unnecessary expected distance calculations. Among all the

[1] The pruning techniques require that for each object o_i, the uncertain region A_i of each object o_i is finite, i.e. $\forall x \notin A_i, f_i(x) = 0$. Thus, each object can be bounded by a finite bounding box.

maximum distances, the minmax distance which is used to prune unnecessary expected distance calculations is the smallest one. The overhead of MinMax-BB includes the time of $MinDist_{i,j}$ and $MaxDist_{i,j}$ calculation.

VDBi

VDBi [58] is another pruning method using Voronoi diagrams [33] to consider the spatial relationships among cluster representatives. VDBi is more efficient than MinMax-BB by using Voronoi-cell pruning and bisector pruning. For Voronoi-cell pruning, given K cluster representatives, the Voronoi diagrams divide the space IR^m into K cells called $V(p_{c_1}), V(p_{c_2}), ..., V(p_{c_j}), ..., V(p_{c_k})$ with the properties of $d(x, p_{c_j}) < d(x, p_{c_k}) \forall x \in V(p_{c_j}), p_{c_j} \neq p_{c_k}$. Therefore, object o_i can be assigned to cluster c_j directly without any expected distance computation with the MBR of object o_i completely falling into any Voronoi cell (i.e. $V(p_{c_j})$). Bisector pruning considers the case of distinct cluster representative pair (i.e. p_{c_j} and p_{c_k} from a set K of cluster representatives). The points in the perpendicular bisector lying on the boundary of a cell $V(p_{c_j})$ and its adjacent cell $V(p_{c_k})$ are denoted by $p_{c_j}|p_{c_k}$. The hyperplane which is perpendicular to the line segment joining p_{c_j} and p_{c_k} and passes through the mid-point of the line segment is called the bisector. The space IR^m is divided into two halves. $H_{j/k}$ denotes the half containing p_{c_j} (excluding the hyperplane). Thus, the following properties are obtained: (i) $\forall p_{c_j}, p_{c_k} \in K, d(y, p_{c_j}) < d(y, p_{c_k}) \forall y \in H_{j/k}$; (ii) $d(y, p_{c_j}) = d(y, p_{c_k}) \forall y \in p_{c_j}|p_{c_k}$. If MBR_i lies completely in $H_{j/k}$, p_{c_k} can be pruned from K. Thus, VDBi can be more efficient than MinMax-BB. If a candidate cluster c_j cannot be pruned by VDBi, neither does MinMax-BB. However, if VDBi may prune a candidate cluster c_j that cannot be pruned by MinMax-BB [58]. The overhead of VDBi includes the time of

Voronoi diagrams construction, Voronoi-cell pruning and bisector pruning.

SHIFT

The pruning methods can be more efficient with the use of cluster-shift technique. Because it is likely that the cluster representatives shift by small distance in the next iteration, the tighter bound can be made to prune candidate clusters more efficiently. Cluster-shift (SHIFT) technique can be applied in MinMax-BB and VDBi. The additional overhead of SHIFT technique includes the time of cluster representative shift calculation between two consecutive iterations. Although the pruning techniques have reduced most of expected distance calculations, it is still expensive to use these pruning techniques for each object in each iteration. Thus, the pruning process becomes a new bottleneck.

2.2.2 Density-based Clustering

Recently, there have been studies on density-based clustering on uncertain data. FDBSCAN [62] and FOPTICS [63] are based on DBSCAN [36] and OPTICS [10] respectively to handle density-based clustering on uncertain objects. In DBSCAN, clusters are formed based on the definitions of reachability and core objects. In FDBSCAN, the definitions of core objects and reachability are re-defined by integrating the information of fuzzy distance functions to handle uncertain objects. If the probability that the number of other objects that are close to o exceeds a certain probability threshold, o is a core object. Whether o is "reachable" from another object x depends on both the probability that x is a core object and the probability of o being close to x. OPTICS is modified in a similar way for FOPTICS to handle uncertain data.

2.2.3 Fuzzy Clustering

Fuzzy logic [84] has been studied for a long time. Fuzzy clustering is another research area related to uncertain data clustering. Among most widely used fuzzy clustering methods [16, 35], for example, in fuzzy c-means, the object is associated with a degree of belongingness for each cluster. Normal or fuzzy data have used fuzzy clustering methods to produce fuzzy clusters [87, 89]. The difference between our work and fuzzy clustering is that we developed the model for clustering on uncertain objects. In our work, each object can only belong to one cluster while in fuzzy clustering each object can belong to more than one cluster with different degrees. The other important difference is that we handle uncertain objects while fuzzy clustering does not.

Additionally, compared with previous work, the cluster representative is considered as a certain point. However, in our work the uncertainty of cluster representative is taken into account.

2.3 Classification

Classification is another classic task in real applications. More research has focused on the problem of classification on certain data [42, 70, 81]. Some work has been extended to handle uncertain data.

SVM

In [17], support vector machine is used to classify uncertain data. In the method, the uncertain object is assumed as a simple bounded geometric model. Support vector machine creates margins by using uncertain objects which overlap the boundary.

In the model, the size of uncertain area of objects are different. If more part of the uncertain area of an object overlaps the boundary, the margin will be influenced and adjusted by the classifier. The difference between traditional SVM and uncertain SVM is that uncertain SVM computes the degree of separation between the two classes [5] by using the probability of a given data point lying on either side of the boundary. In uncertain SVM, the size of uncertainty area is estimated in the algorithm, but the maximum boundary of objects is given in our model.

uRules

In [80], uRule is proposed based on Rule-based algorithm to classify uncertain information. The difference between Rule-based and uRule is that the instances are partly covered by the rule in uRule. The key idea in uRule is that the algorithm computes which proportion of the instances is covered by a rule based on the uncertain attribute interval and probabilistic function. uRule considers to classify uncertain numerical and categorical data.

Uncertain Decision Tree and Naive Bayes

In [94, 95], an uncertain object is associated with a probability density function (pdf) and a finite region. The decision tree classifier is extended to handle uncertain data by using averaging or distribution-based approach. To improve the efficiency, some pruning techniques were proposed without affecting the results of decision tree. In [82], the uncertain model is the same as that in [94, 95]. In [82], Naive Bayes is extended to classify uncertain data. Three approaches are proposed (averaging, formula-based, and sample-based) to calculate the probability of object label and assign it to the class with highest probability.

The uncertain data are represented by hundreds or thousands samples, so, the classification methods are more complex when handling uncertain data. The computational cost on uncertain data classification are expensive. Thus, in this thesis we propose supervised UK-means to classify uncertain data which is simpler and more time-saving than existing methods.

2.4 Other Mining Techniques on Uncertain Data

There are also studies on other data mining tasks on uncertain data, such as outlier detection, frequent pattern mining, and domain orders learning.

In [4, 55, 71, 97], outlier detection is discussed. In [55] and [4], an uncertain object is represented by a probability density function (pdf). In [4], an uncertain point o is a (α,β)-outlier, if the probability of Y falling into a region of some subspace is less than β whose density is at least α. Wang et al. [97] proposed outlier detection on an uncertain table based on distance method. The table is consist of a set of tuples, and each tuple is represented by an appearing probability. Possible world semantics in [1, 31, 53, 54, 86] are the basis of outlier definition. Matsumoto et al. [71] proposed a new implementation for outlier detection by using parallelized cross-platform OpenCL framework on uncertain data. Different from other work, Jiang et al. [55] also considers outlier instances while others only focus on the outlier objects.

The problem of frequent pattern mining on uncertain data are proposed in the literature [3, 15, 28, 29, 67, 108]. Expected support [3, 28, 29, 67, 108] and frequentness probability [15] are proposed to handle uncertainty, and both follow the possible world semantics. Frequent pattern mining was first extended to handling uncertain data by Chui et al. [29]. Chui et al. [29] modified the classic Apriori algorithm to U-Apriori by using expected support. Chui et al. [28] also developed a pruning technique to speed up the U-Apriori algorithm. Aggarwal et al. [3] extended non-candidate generation algorithms to uncertain data (i.e. H-mine algorithm [75] and FP-growth algorithm [46]). Frequentness probability is used in [15] to measure the support of an itemset in an uncertain transaction database.

Learning domain orders on certain data is widely studied in the application of mining user preferences, for example, preference queries and recommendation systems [65] on large databases [25, 26, 41, 60, 64]. The framework of preference learning in multidimensional space for numerical and categorical domains is proposed in [59]. In [7], a framework of combining and expressing the preferences is proposed. On the other side, the model of domain order is related to the notion of dominance relationship in skyline query processing [19, 27, 40, 61, 90]. A large number of skyline variations are also proposed in the literature [21, 34, 77, 93, 109]. However, few work focuses on handling uncertain data. Jiang et al. [56] first brought domain learning into uncertain data. They learn domain orders on uncertain data by using greedy method.

2.5 Summary

In this chapter, we briefly introduce the model of uncertain data, and review the methods that have been used in clustering and classification on uncertain data. In this chapter, we also give a glance at other mining techniques on uncertain data.

In Chapter 3, we analyze existing algorithms for clustering uncertain data. Existing algorithms are time consuming because of the calculation of expected distance. We reduce UK-means to K-means by using expected squared distance. Then we propose Approximate UK-means to heuristically identify objects of boundary cases and re-assign them to better clusters. Additionally, we consider the uncertainty of cluster representative and propose three models for cluster representative. The major result of Chapter 3 is published in *Proceedings of the 23rd Australasian Joint Conference on Artificial Intelligence (AI 2010)* [52].

In Chapter 4, we build an efficient classifier based on UK-means to handle uncertain data which is much faster than existing methods. To capture related properties of data, we extend feature selection to supervised UK-means. Moreover, we also extend ensemble model to supervised UK-means based on Adaboost [38]. The major result of Chapter 4 is published in *Proceedings of the 24th Australasian Joint Conference on Artificial Intelligence (AI 2011)* [105].

In Chapter 5, we overcome the limitation of supervised UK-means when a class is divided by other classes. We propose supervised UK-means with multiple subclasses (SUMS). In SUMS, we estimate the number of subclasses first, and then learn the subclass representatives. To avoid overfitting, we add a *condition* on SUMS. The major result of Chapter 5 is published in *Proceedings of the 25th Australasian Joint Conference on Artificial Intelligence (AI 2012)* [106].

Chapter 3

Clustering on Uncertain Objects

In this chapter, we develop a clustering framework to summarize uncertain objects efficiently and effectively.

In previous work, UK-means is based on K-means for clustering on uncertain objects. UK-means is time consuming due to expensive distance computational cost. Ngai et al. [58, 73] improve the efficiency of UK-means by pruning candidate clusters. However, existing algorithms are still undesirably slow due to the overhead of pruning candidate clusters for each object in each iteration.

In this chapter, we reduce UK-means to K-means by using squared Euclidean Distance and heuristically reduce the discrepancy generated by different distance metrics. Compared with previous work, we consider the uncertainty of cluster representative. Three models are developed for representing cluster representative (certain model, uncertain model and heuristic model) to calculate expected squared Euclidean distance between objects and cluster representatives. We briefly introduce uncertain objects clustering in Section 3.1. In Section 3.2, we illustrate some clustering techniques on uncertain objects. In Section 3.3, we discuss three models for representing cluster representatives. In Section 3.4, we introduce "Approximate UK-means" which heuristically identifies objects on the boundary cases and re-assigns them to better clusters in order to reduce the discrepancies in clustering results. Section 3.5 demonstrates the efficiency and effectiveness of Approximate UK-means by extensive experiments. We summarize this chapter in Section 3.6.

3.1 Introduction

Clustering of such "uncertain" data can be illustrated by the following simple realistic example. Consider sensors on wild animals that update their locations periodically. The sample locations of an animal over a period generate a (discrete) probability distribution function (PDF) which describes the possible location of the animal. Clustering results on those animals may reveal the possible groups and interactions between them. In our work, we consider the problem of clustering objects with multidimensional uncertainty where an object is represented by an uncertain region over which a discrete probability distribution function (PDF) or a probability density function (pdf) is defined.

Formally, we consider a set of n objects o_i, $1 \leq i \leq n$ in an m-dimensional space. An object o_i is represented by a pdf f_i: $IR^m \rightarrow IR$ (IR represents real number space) that specifies the probability density of each possible location of object o_i. The methods to be discussed in this chapter just require that for each object o_i, the uncertain region A_i of each object o_i is finite, (i.e. $\forall x \notin A_i, f_i(x) = 0$) without relying on any special forms of the pdf (f_i). Thus, a bounding box can be used to bound each object. In practice, the probability density of an object is just concentrate in a very small region, so that the assumption is convincing.

The goal of clustering is to group n these objects into K clusters so that the sum of *expected Euclidean distances* (EED) [22] between the uncertain objects and their cluster centers is minimized. Thus, suppose $C(o_i) = c_j$ represents that object o_i is assigned to cluster c_j, and $p_{C(o_i)}$ is the cluster's representative point, we want to find the K cluster representatives such that the objective function $\sum_{i=1}^{n} EED(o_i, p_{C(o_i)}) = \sum_{i=1}^{n}(\int f_i(x)ED(x, p_{C(o_i)})dx)$ is minimized where ED is the Euclidean distance function based on a metric d (i.e. Euclidean distance in UK-means and pruning algorithms, squared Euclidean distance in our methods).

3.2 Clustering Techniques on Uncertain Objects

Efficiency is important in real time application. The bottlenecks of uncertain object clustering are expected distance calculation and pruning of candidate clusters. However, by considering squared Euclidean distance (instead of Euclidean distance as in UK-means), UK-means can be reduced to K-means (so, no pruning of clusters is necessary) [66], which is running much faster with some discrepancies in the clustering results as shown in the experimental section.

3.2.1 Expected Distance Calculation between Uncertain Objects

If each attribute is seen as a dimension, a certain object is represented as a point in a multidimensional space produced by the domains of all attributes. Because of the system limitation and uncertain nature during data collection, the imperfect data quality causes uncertain attribute values of an object. A set of points are used to represent an uncertain object, and each point is a possible location of the object. A probability distribution function (PDF) is defined for representing the distribution of the probabilities of possible location. A finite or infinite region can also be used to represent an uncertain object by covering all the possible locations of the object (especially for infinite number of possible locations). $UD(o_i)$ is noted as the uncertain domain (region) of object o_i. A probability density function (pdf), f_i, is used to indicate the probability density of each possible location within the region ($\int_{UD(o_i)} f_i(x)\,dx = 1$).

There are two uncertain objects o_i, o_j, whose pdfs are $f_i(x_i)$, $f_j(x_j)$, where x_i and x_j are possible locations of them and o_j. The uncertain domains of them are $UD(o_i)$ and $UD(o_j)$. The distance between possible locations (x_i and x_j) of uncertain objects is denoted as $D(x_i, x_j)$. The expected distance and the pdf of the expected distance between o_i and o_j are given as following.

$$E(D(o_i, o_j)) = \int_{UD(o_i)} \int_{UD(o_j)} D(x_i, x_j) f_i(x_i) f_j(x_j)\, dx_i\, dx_j\ . \tag{3.1}$$

The pdf $D_{i,j}$ is defined to return the probability of a distance value as following:

$$D_{i,j}(s) = \int_{UD(o_i)} \int_{UD(o_j)} F(D(x_i, x_j), s) f_i(x_i) f_j(x_j)\, dx_i\, dx_j\ . \tag{3.2}$$

where s is a non-negative real number; $F(x, y) = 1$ if $x = y$; $F(x, y) = 0$ otherwise. In other words, $D_{i,j}(s)$ returns the probability that the distance between object o_i, o_j is actually s. The expected distance between o_i and o_j can be represented in terms of $D_{i,j}(s)$ as Equation (3.3). When PDFs (e.g. F_i) are used instead of pdfs(e.g. f_i), $\int_{UD(o_i)} \int_{UD(o_j)} f_i(x_i) f_j(x_j)\, dx_i\, dx_j$ is changed to $\sum_{UD(o_i)} \sum_{UD(o_j)} F_i(x_i) F_j(x_j)$.

$$E(D(o_i, o_j)) = \int_0^{\infty} D_{i,j}(s) s\, ds\ . \tag{3.3}$$

The distance function D can be Euclidean distance, squared Euclidean distance and Manhattan distance, etc. The following section presents the analytical solutions of expected distance calculation of uncertain objects. In the case of clustering on uncertain objects, both the objects and cluster representatives are uncertain objects like o_i and o_j.

3.2.2 Reduce UK-means to K-means

An uncertain object can also be represented as a set of points, each of which is a possible location of object o_i [58, 73, 104]. As Figure 3.1 shows, the uncertain domain is divided into a number of grid cells. Each grid cell represents a possible location of o_i. The expected Euclidean distance (EED) from object o_i (represented by a pdf f_i) to the cluster representative p_{c_j} is the weighted average of the distances between the samples in o_i and p_{c_j}, i.e. $EED(o_i, p_{c_j}) = \sum_{t=1}^{T} \sum_{z=1}^{|c_j|\times T} F_i(s_{i,t}) F_{p_{c_j}}(p_{c_j,z}) ED(s_{i,t}, p_{c_j,z})$, where T is the number of samples in o_i, $|c_j|$ is the number of objects assigned to cluster c_j, $s_{i,t}$ is the location (vector) of the tth sample of o_i, $p_{c_j,z}$ is the location (vector) of the zth sample of cluster representative p_{c_j}, $F_i(s_{i,t}) = \int_{x\in cell_t} f_i(x)dx$ (F_i is a discrete probability distribution function over T grid cells, $cell_t$ is the grid cell that sample $s_{i,t}$ represents, x is the possible location of sample s_t in $cell_t$), which is similar to $F_{p_{c_j}}(p_{c_j,z})$, and the metric ED is Euclidean distance used in [22, 58, 73], squared Euclidean distance in [104] and our method.

By using expected squared Euclidean distance, Lee. et al. [66] shows that UK-means algorithm can be reduced to K-means. In the following, we are going to show another derivation by applying the analytic solution in [104]. We will first define the mean vector and the trace of covariance matrix of an uncertain object

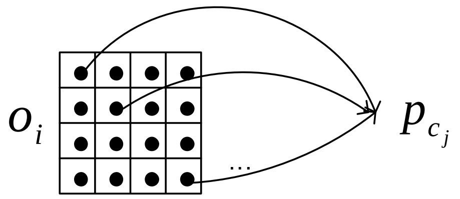

Figure 3.1 Expected distance calculation from o_i to p_{c_j} in [22, 58, 73]

given its samples as follows. Suppose $\overline{o_i}$ is a $m \times 1$ (m is the number of dimensions) mean vector of an uncertain object o_i, which is the weighted mean of all T samples (or possible locations) in the object as Formula (3.4).

$$\overline{o_i} = \sum_{t=1}^{T} s_{i,t} \times F_i(s_{i,t}). \tag{3.4}$$

Where $s_{i,t}$ represents the tth sample of object o_i. Suppose Σ_{o_i} is a $m \times m$ covariance matrix of samples of o_i. $trace(\Sigma_{o_i})$ is the sum of all diagonal elements in Σ_{o_i}. On the other hand, $trace(\Sigma_{o_i})$ can also be expressed as Equation (3.5):

$$trace(\Sigma_{o_i}) = \sum_{t=1}^{T} ||s_{i,t} - \overline{o_i}||^2 \times F_i(s_{i,t}). \tag{3.5}$$

In [104], the expected squared Euclidean distance (ESED) between two uncertain objects o_i and o_j can be obtained by

$$ESED(o_i, o_j) = ||\overline{o_i} - \overline{o_j}||^2 + trace(\Sigma_{o_i}) + trace(\Sigma_{o_j}). \tag{3.6}$$

It is obvious that we can preprocess the uncertain objects and obtain their $\overline{o_i}$ and $trace(\Sigma_{o_i})$ in the beginning so that Expected Squared Euclidean Distance (ESED) between any object o_i and any cluster representative p_{c_j} can be easily obtained. Given an uncertain object o_i, to find the closer one out of two cluster representatives p_{c_j} and p_{c_k}, we could calculate the difference between their ESED from o_i:

$ESED(o_i, p_{c_k}) - ESED(o_i, p_{c_j}) = (||\overline{o_i} - \overline{p_{c_k}}||^2 + trace(\Sigma_{o_i}) + trace(\Sigma_{p_{c_k}})) - (||\overline{o_i} - \overline{p_{c_j}}||^2 + trace(\Sigma_{o_i}) + trace(\Sigma_{p_{c_j}})) = ||\overline{o_i} - \overline{p_{c_k}}||^2 - ||\overline{o_i} - \overline{p_{c_j}}||^2 + trace(\Sigma_{p_{c_k}}) - trace(\Sigma_{p_{c_j}})$, where $\overline{p_{c_k}}$ and $\overline{p_{c_j}}$ are the mean vectors of p_{c_k} and p_{c_j} respectively. As a result, it is no longer necessary to add $trace(\Sigma_{o_i})$ in ESED. Instead of calculating the whole ESED, we only need to calculate a part of the Expected Squared Euclidean distance (PESED) between uncertain object o_i and cluster representative p_{c_j} as follows:

$$PESED(o_i, p_{c_j}) = ||\overline{o_i} - \overline{p_{c_j}}||^2 + trace(\Sigma_{p_{c_j}}). \tag{3.7}$$

In this chapter, the mean vector of cluster representatives $\overline{p_{c_j}}$ are obtained by Equation (3.8), where $|c_j|$ is the number of objects assigned to cluster c_j.

$$\overline{p_{c_j}} = \frac{1}{|c_j|} \sum_{i=1}^{|c_j|} \overline{o_i}. \tag{3.8}$$

Compared with the traditional expected distance calculation method in [58, 73], the execution time of expected squared Euclidean distance calculation is not related to the number of samples. The overhead time in our method is the precomputation of mean vector and $trace(\Sigma_{o_i})$ ($trace(\Sigma_{p_{c_j}})$ is obtained by $trace(\Sigma_{o_i})$ and $|c_j|$, to be discussed later). The consideration of uncertainty of clustering representative will be introduced in Section 3.3.

3.3 Models of Cluster Representative

3.3.1 Certain Model of Cluster Representative

In most related work, p_{c_j} is assumed to be a certain point, so $\overline{p_{c_j}}$ is obtained by Equation (3.8) and $trace(\Sigma_{p_{c_j}}) = 0$. From Equation (3.7), instead of calculating

PESED, we only need to calculate Means' Expected Squared Euclidean distance (MSED) between uncertain object o_i and cluster representative p_{c_j} as follows:

$$MSED(o_i, p_{c_j}) = ||\overline{o_i} - \overline{p_{c_j}}||^2. \tag{3.9}$$

In this model, we can preprocess the uncertain objects and obtain their $\overline{o_i}$ in the beginning so that Mean Squared Euclidean Distance (MSED) between any object o_i and any cluster representative p_{c_j} can be readily obtained.

3.3.2 Uncertain Model of Cluster Representative

From Equation (3.7), if the difference between $trace(\Sigma_{p_{c_j}})$ and $trace(\Sigma_{p_{c_k}})$ is quite large, the variance of cluster representative is likely to affect clustering results. In this model, we consider to represent a cluster representative as an uncertain object. In Equation (3.7), p_{c_j} is assumed to be an uncertain object, $\overline{p_{c_j}}$ is obtained by Equation (3.8) and $trace(\Sigma_{p_{c_j}})$ is represented as Equation (3.10).

$$\begin{aligned} trace(\Sigma_{p_{c_j}}) &= trace(\sum_{z=1}^{|c_j|} \frac{\Sigma_{o_z}}{|c_j|}) \\ &= \frac{\sum_{z=1}^{|c_j|} trace(\Sigma_{o_z})}{|c_j|^2} \ . \end{aligned} \tag{3.10}$$

where o_z is the object assigned to cluster c_j, and $|c_j|$ is the number of objects assigned to cluster c_j. Furthermore, the part of Expected Squared Euclidean distance (PESED) between object o_i and cluster representative p_{c_j} can be obtained from Equation (3.11).

$$PESED(o_i, p_{c_j}) = ||\overline{s_i} - \overline{p_{c_j}}||^2 + \frac{\sum_{z=1}^{|c_j|} trace(\Sigma_{o_z})}{|c_j|^2}. \tag{3.11}$$

In this model, we can preprocess the uncertain objects and obtain their $\overline{o_i}$ and $trace(\Sigma_{o_i})$ in the beginning. The uncertainty of each cluster c_j ($trace(\Sigma_{p_{c_j}})$) should be calculated during each iteration. Part of Expected Squared Euclidean Distance (PESED) between any object o_i and any cluster representative p_{c_j} can then be obtained. The uncertain model of cluster representative is slower than certain model because of the calculation of $trace(\Sigma_{o_i})$ (once in the beginning) and $trace(\Sigma_{p_{c_j}})$. $trace(\Sigma_{p_{c_j}})$ is related to the sample variance of objects assigned to the cluster c_j and $|c_j|$ (the number of objects assigned to c_j).

3.3.3 Heuristic Model of Cluster Representative

The heuristic model considers the cluster representative as a certain object at first to cluster objects, and then heuristically considers the uncertainty of cluster representatives. In this model, we first calculate Means' Squared Euclidean distance by Equation (3.9) using certain model (K-means). When the algorithm using certain model converges, we use the uncertain cluster representative model to re-cluster objects based on the clustering results obtained by certain model. The experimental results of clustering algorithms are shown in Section 3.5.

3.4 Approximate UK-means

In Section 3.2, UK-means is reduced to K-means by using squared Euclidean distance while UK-means originally uses Euclidean distance. It is not surprising that the clustering results of K-means will deviate from those of UK-means.

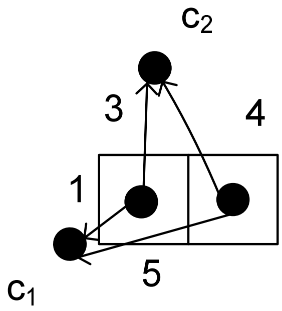

Figure 3.2 An example of the cause of discrepancy.

3.4.1 Discrepancy

The order of cluster representatives sorted by their expected distances to a given object may be different from that sorted by expected squared distances. Figure 3.2 gives an example of discrepancy between these two distance functions. Assume an uncertain object has two samples in two grid cells and the probabilities of these two samples are 0.5. The distances from one sample of the uncertain object to cluster representatives c_1 and c_2 are 1 and 3 respectively. The distances from the other sample of the uncertain object to c_1 and c_2 are 5 and 4 respectively. Thus, the uncertain object will be assigned to cluster c_1 according to expected distance ($\frac{1+5}{2} < \frac{3+4}{2}$). However, the uncertain object belongs to cluster c_2 according to expected squared distance ($\frac{1^2+5^2}{2} < \frac{3^2+4^2}{2}$).

Figure 3.3 gives an example of the same clustering result no matter expected distance or expected squared distance is used. The assumption of this case is the same as that of Figure 3.2. The distances from one sample of the uncertain object to c_1 and c_2 are 1 and 3 respectively. The distances from the other sample of the uncertain object to c_1 and c_2 are 2 and 4. The uncertain object is assigned to cluster c_1 if the expected distance is used ($\frac{1+2}{2} < \frac{3+4}{2}$), and it is also assigned to cluster c_1

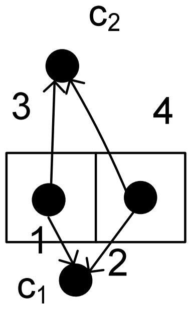

Figure 3.3 An example of the same clustering result.

if the expected squared distance is used instead ($\frac{1^2+2^2}{2} < \frac{3^2+4^2}{2}$). It means the order of cluster representatives sorted by their expected distances to a given object is the same as that sorted by expected squared distances.

Discrepancy is used to measure the difference of clustering results between two clustering algorithms based on purity. Purity is the maximum probability that a cluster in algorithm A contains objects of the same cluster from algorithm B. The purity of cluster c_i is defined as Equation(3.12).

$$prob_i = \max_j prob_{ij}. \tag{3.12}$$

where $prob_{ij}$ is the probability that a member of cluster c_i in algorithm A belongs to cluster c_j in algorithm B. Note that $prob_{ij} = \frac{n_{ij}}{n_i}$ where n_i is the number of objects in cluster c_i in algorithm A, and n_{ij} is the number of objects of cluster c_j (in algorithm B) within these n_i objects. The overall purity of clustering result of algorithm A is defined as Equation(3.13).

$$purity = \sum_{i=1}^{K} \frac{n_i}{n} prob_i. \tag{3.13}$$

where n is the number of uncertain objects, K is the number of clusters, and the range of purity is $[0, 1]$. Finally we define discrepancy as Equation(3.14).

$$discrepancy = 1 - purity. \tag{3.14}$$

3.4.2 Boundary Case

Definition of Boundary Case p_{c_m} and p_{c_q} are two closest cluster representatives of object o_i. $average_{MSED}$ is the average of $MSED(o_i, p_{c_m})$ and $MSED(o_i, p_{c_q})$ where MSED is Means' Squared Euclidean Distance. Assume $MSED(o_i, p_{c_m}) < MSED(o_i, p_{c_q})$, object o_i is defined to be a boundary case if $MSED(o_i, p_{c_m}) \geq \beta \times average_{MSED}(0 \leq \beta \leq 1.0)$, where β is an input parameter. β ranges from 0 to 1 and is fixed in the algorithm. If β is close to 1, the two MSEDs of the boundary case o_i from the two closest cluster representatives will be close and o_i is close to the boundary between clusters c_m and c_q. We notice that objects assigned to a cluster that is different from another algorithm are likely near the boundary between clusters. Therefore, we propose a heuristic called **Approximate UK-means**.

3.4.3 Algorithms

The basic idea of the heuristic is picking out objects near the boundary between two closest clusters and re-assigning them in the first clustering iteration of Approximate UK-means. In Approximate UK-means, if object o_i is a boundary case, we calculate the expected Euclidean distances from o_i to p_{c_m} and p_{c_q} and assign o_i to the closest cluster. **The above is only done in the first clustering iteration of Approximate UK-means because we observed that the assignment of objects in the first iteration is the most important, which will greatly affect the later**

Algorithm 1 Approximate UK-means of Certain Model

```
randomly initialize all cluster representatives (reps) mean vectors $\overline{p_{c_j}}$;
for i=0; $i < n$; i++ do
    precompute the mean vector $\overline{o_i}$ of object $o_i$;
end for
repeat
    if this is the first iteration then
        find and re-assign boundary cases by Algorithm 4;
    else
        for i=0; $i < n$; i++ do
            for j=0; $j < K$; j++ do
                compute Means' Squared Euclidean Distance $MSED(o_i, p_{c_j}) = ||\overline{o_i} - \overline{p_{c_j}}||^2$;
            end for
            assign $o_i$ to cluster $c_m$ where $p_{c_m}$ is the closest cluster rep by $MSED$;
        end for
    end if
    update all cluster reps $p_{c_j}$ by Equation(3.8);
until all cluster reps converge
```

iterations due to the shift of cluster representatives.

Algorithm 1 shows Approximate UK-means using certain cluster representative model, where n is the number of objects, and K is the number of clusters.

Algorithm 2 shows Approximate UK-means using uncertain cluster representative model. In Algorithm 2, the cluster representative is uncertain, and initialize

Algorithm 2 Approximate UK-means of Uncertain Model

```
randomly initialize all cluster representatives (reps) mean vectors $\overline{p_{c_j}}$ and $trace(\Sigma_{p_{c_j}}) = 0$;
for i=0; $i < n$; i++ do
    precompute the mean vector $\overline{o_i}$ and $trace(\Sigma_i)$ of object $o_i$;
end for
repeat
    if this is the first iteration then
        find and re-assign boundary cases by Algrithm 4;
    else
        for i=0; $i < n$; i++ do
            for j=0; $j < K$; j++ do
                compute Part of Squared Euclidean Distance $PESED(o_i, p_{c_j}) = ||\overline{o_i} - \overline{p_{c_j}}||^2 + trace(\Sigma_{p_{c_j}})$;
            end for
            assign $o_i$ to cluster $c_m$ where $p_{c_m}$ is the closest cluster rep by $PESED$;
        end for
    end if
    update all cluster reps $\overline{p_{c_j}}$ by Equation(3.8) and $trace(\Sigma_{p_{c_j}})$ by Equation(3.10);
until all cluster reps converge
```

$trace(\Sigma_{p_{c_j}}) = 0$. In clustering process, the expected squared Euclidean distance includes the term $trace(\Sigma_{p_{c_j}})$. When update $\overline{p_{c_j}}$, $trace(\Sigma_{p_{c_j}})$ is calculated as well.

Algorithm 3 shows Approximate UK-means using heuristic cluster representative model. In Algorithm 3, the initial cluster representatives are the same as those of Algorithm 2. The clustering process is a combination of certain model and uncertain model. The cluster representative is assumed to be certain at first. Thus, we calculate the MSED between uncertain object o_i and cluster representative p_{c_j} and update the mean vector of cluster representative ($\overline{p_{c_j}}$) during the certain clustering process. When the shift of cluster representative converges, then we use uncertain model to calculate the PESED between uncertain object o_i and cluster representative p_{c_j}, and update $\overline{p_{c_j}}$ and $trace(\Sigma_{p_{c_j}})$ during the uncertain cluster representative update process.

In all the Approximate UK-means, heuristic (Algorithm 4) is used to find and reassign boundary cases. The time complexity of Approximate UK-means is $O(nKT_1)$, where n is the number of objects, K is the number of clusters, and T_1 iteration times. In the experiments, we set β from 0.7 to 1. If $\beta = 1$, the inequality of boundary case becomes $MSED(o_i, p_{c_m}) \geq average_{MSED}$. In other words, the inequality of boundary case is $MSED(o_i, p_{c_m}) \geq MSED(o_i, p_{c_q})(\beta = 1.0)$. However, p_{c_m} is the closest cluster representative, so $MSED(o_i, p_{c_m}) \leq MSED(o_i, p_{c_q})$, by definition the algorithm cannot find boundary cases. If $MSED(o_i, p_{c_m})$ is equal to $MSED(o_i, p_{c_q})$, then the Euclidean distance between o_i and p_{c_m} is the same as Euclidean distance between o_i and p_{c_q}. The re-assignment of cluster representative will become unnecessary and redundant. Thus, Approximate UK-means is reduced to K-means if $\beta = 1.0$. In the definition, $MSED(o_i, p_{c_m}) \geq \beta \times average_{MSED}(0 \leq \beta \leq 1.0)$, then o_i is considered as boundary case. In other words, if $\frac{MSED(o_i,p_{c_q})}{MSED(o_i,p_{c_m})} \leq \frac{2}{\beta} - 1$, o_i is boundary case. When β decreases, $\frac{2}{\beta} - 1$ become larger. Thus, more objects are considered as boundary cases because more objects satisfy the inequality and will be re-checked

Algorithm 3 Approximate UK-means of Heuristic Model

```
randomly initialize all cluster representatives (reps) mean vectors $\overline{p_{c_j}}$ and $trace(\Sigma_{p_{c_j}}) = 0$;
Line 2-17 same as Line 2-17 in Algorithm 1;
repeat
    for i=0; $i < n$; i++ do
        for j=0; $j < K$; j++ do
            compute Part of Squared Euclidean Distance $PESED(o_i, p_{c_j}) = \|\overline{o_i} - \overline{p_{c_j}}\|^2 + trace(\Sigma_{p_{c_j}})$;
        end for
        assign object $o_i$ to cluster $c_m$ where $p_{c_m}$ is the closest cluster rep by $PESED$;
    end for
    update all cluster reps $p_{c_j}$ by Equation(3.8) and $trace(\Sigma_{p_{c_j}})$ by Equation(3.10);
until all cluster reps converge
```

again. In fact, experimental results in the next section show that this heuristic can significantly reduce the discrepancies of clustering result by up to 70% compared with K-means. The tradeoff is only 25% more execution time, which is still at least 10 times faster than existing pruning algorithms.

Algorithm 4 Heuristic for Finding Boundary Case

```
for i=0; $i < n$; i++ do
    for j=0; $j < K$; j++ do
        compute Means' Squared Euclidean Distance $MSED(o_i, p_{c_j}) = ||\overline{o_i} - \overline{p_{c_j}}||^2$;
    end for
    let $p_{c_m}$ and $p_{c_q}$ be the 1st and 2nd closest cluster reps by $MSED$;
    $average_{MSED} = (MSED(o_i, p_{c_m}) + MSED(o_i, p_{c_q}))/2$;
    if $MSED(o_i, p_{c_m}) \geq \beta \times average_{MSED}$ then
        compute $o_i$'s expected Euclidean distances (EED) from $p_{c_m}$ and $p_{c_q}$, and
        assign $o_i$ to cluster with smaller EED;
    else
        assign object $o_i$ to cluster $c_m$;
    end if
end for
```

3.5 Experimental Evaluation

In this section, we evaluate Approximate UK-means using three different models experimentally by comparing it with K-means and pruning UK-means (MinMax-SHIFT and VDBi-SHIFT) [58, 73]. Section 3.5.2 compares their execution time and Section 3.5.3 compares their clustering results. All algorithms were written in Java and were run on a Linux machine with an Intel 2.5GHz Pentium(R) Dual-Core processor and 8GB of main memory.

Table 3.1 Parameters for experiments using data sets.

Parameter	Description	Baseline Value
n	number of uncertain objects	20000
K	number of clusters	50
T	number of samples per object	196
S	maximum size of MBR, $S \times S$	5
$mindis$	minimum distance between two clusters	2
D	number of dimensions	2
σ	standard deviation of Gaussian distribution	16
β	the factor of picking out boundary cases	0.8

3.5.1 Data Sets

For ease of comparison with previous work like [58, 73, 94] which used synthetic data sets only, we generated 125 random data sets for the experiments. For each data set, a set of n uncertain objects represented by MBRs with size $S \times S$ was randomly generated in 2D space $[0, 100] \times [0, 100]$. An MBR is divided into $\sqrt{T} \times \sqrt{T}$ grid cells. Each grid cell corresponds to a sample. Each sample is associated with a randomly generated probability value. All probabilities in an MBR are normalized to have their sum equal to 1. For each data set, a set of K cluster representatives was randomly initialized and was repeatedly used in all experiments on the same data set for more fair comparisons. This is to eliminate variations in the results due to the uses of different sets of initial cluster representatives.

To make the clustering results more reasonable, we also generated 125 data sets with Gaussian distribution. The n uncertain objects in a data set were equally grouped into K clusters. For each cluster, the centers of $\frac{n}{K}$ uncertain objects were generated from a Gaussian distribution, whose mean and standard deviation are equal to the cluster center and σ respectively. The cluster center was randomly generated and was restricted to have a minimum distance *mindis* with other cluster centers.

The parameters used for the experiments on random data sets and gaussian data sets are summarized in Table 3.1. For each set of parameters, a set of five experiments was run on five different randomly generated data sets. Each experiment was repeated on the six algorithms. The average value of five runs on each algorithm was taken and reported.

3.5.2 Execution Time

Varying Sample Number

In the experiments, we varied the sample number T of an object from 100 to 900. The other parameters were kept at baseline values. Figure 3.4(a) shows the execution time of the six algorithms on random data sets. However, all three Approximate UK-means run almost as fast as K-means and their execution time grows much slower than MinMax-SHIFT and VDBi-SHIFT. The significant improvement in the performance of Approximate UK-means is due to two reasons: (i) the distance calculation is done much faster (Figure 3.4(b)), and (ii) the overhead is much reduced as no pruning is necessary (Figure 3.4(c)). Figure 3.4(b) also shows that the (expected) distance calculation time of Approximate UK-means (in all three models) does not change a lot with sample number T, because PESED calculation

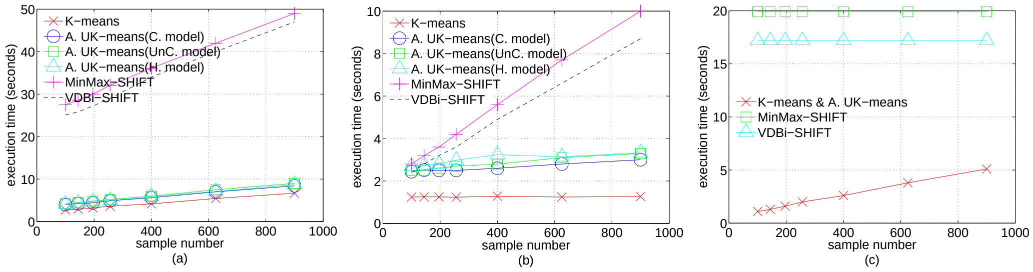

Figure 3.4 (a) Total clustering time with varying T on random data sets (RDS) (b) (Expected) distance calculation time with varying T on RDS (c) Overhead time with varying T on RDS.

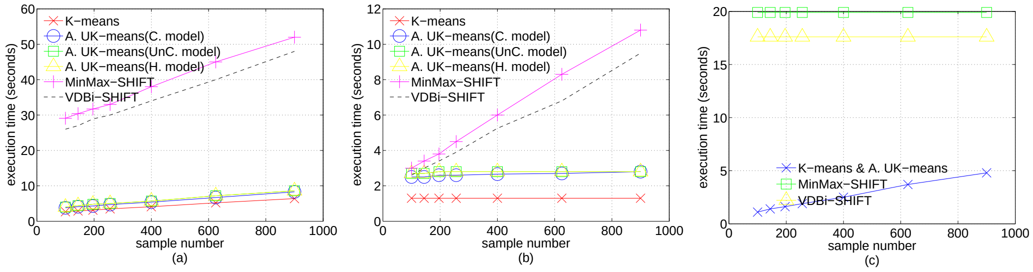

Figure 3.5 (a) Total clustering time with varying T on Gaussian data sets (GDS) (b) (Expected) distance calculation time with varying T on GDS (c) Overhead time with varying T on GDS.

does not depend on sample number. And the expected distance calculation in the first clustering iteration of Approximate UK-means is only a minor cost. In Figure 3.4(a), the total execution time of the three Approximate UK-means models is similar. The (expected) distance calculation in Approximate UK-means using Uncertain model (UnC. model) and Approximate UK-means using Heuristic model (H. model) add the variance of cluster representative. Thus, the uncertain model and heuristic model cost a little more time than certain model (Figure 3.4(b)). The execution time comparison of six algorithms on Gaussian data sets with varying T (Figure 3.5) is similar to that on random data sets (Figure 3.4).

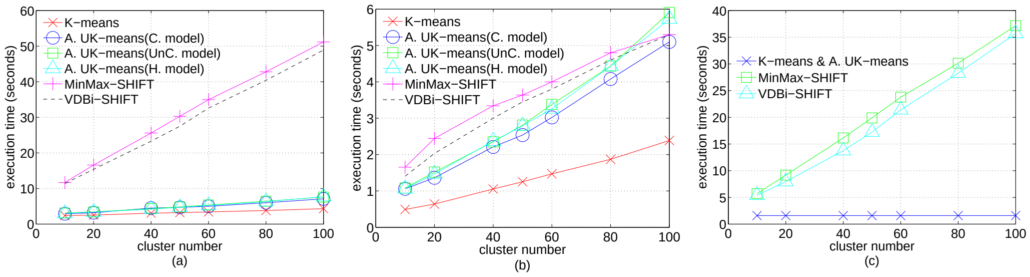

Figure 3.6 (a) Total clustering time with varying K on RDS (b) (Expected) distance calculation time with varying K on RDS (c) Overhead time with varying K on RDS.

Varying Cluster Number

In the experiments, we varied the cluster number K from 10 to 100. The other parameters were kept at baseline values. Figure 3.6 shows the execution time of the six algorithms on random data sets. Figure 3.6(a) shows that the total execution time of all six algorithms grows as K increases. However, all Approximate UK-means almost spend the same time as K-means and its execution time grows much slower than MinMax-SHIFT and VDBi-SHIFT. From Figure 3.6(b), (expected) distance calculation of K-means is more efficient than the other five algorithms. (Expected) distance calculation of Approximate UK-means (all three models) is faster than MinMax-SHIFT and VDBi-SHIFT if K is smaller than 100. The overhead time of Approximate UK-means is not related to K while the overhead time of pruning techniques grows linearly with K (Figure 3.6(c)). Similar to the situation of varying sample number T, in Figure 3.6(a), the total execution time of all the three Approximate UK-means models is similar. Moreover, the uncertain model and heuristic model cost a little more time than certain model (Figure 3.6(b)). The execution time comparison of six algorithms on Gaussian data sets with varying K (Figure 3.7) is similar to that on random data sets (Figure 3.6).

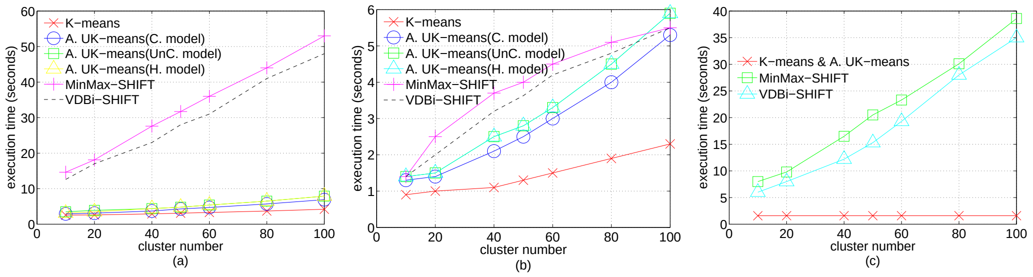

Figure 3.7 (a) Total clustering time with varying K on GDS (b) (Expected) distance calculation time with varying K on GDS (c) Overhead time with varying K on GDS.

Varying Maximum Size of MBR

In the experiments, we varied the maximum MBR size ($S \times S$) by varying S from 5 to 25. The other parameters were kept at their baseline values. Figure 3.8(a) and Figure 3.9(a) show the execution time of the six algorithms on random data sets and Gaussian data sets respectively. Figure 3.8(a) and Figure 3.9(a) show that the total execution time of pruning algorithms increases as S increases while K-means and Approximate UK-means (all three models) do not increase with S. However, all Approximate UK-means run almost as fast as K-means and its execution time grows much slower than MinMax-SHIFT and VDBi-SHIFT. The significant improvement in the performance of all Approximate UK-means is due to two reasons: (i) the distance calculation is done much faster (Figure 3.8(b) and Figure 3.9(b)), and (ii) the overhead is much reduced as no pruning is necessary (Figure 3.8(c) and Figure 3.9(c)). In Figure 3.8(a) and (b), the total execution time of all three Approximate UK-means models is similar, and the (expected) distance calculation time of all three Approximate UK-means models is also similar.

Varying Object Number

In the experiments, we varied the object number n from 5000 to 60000. The other

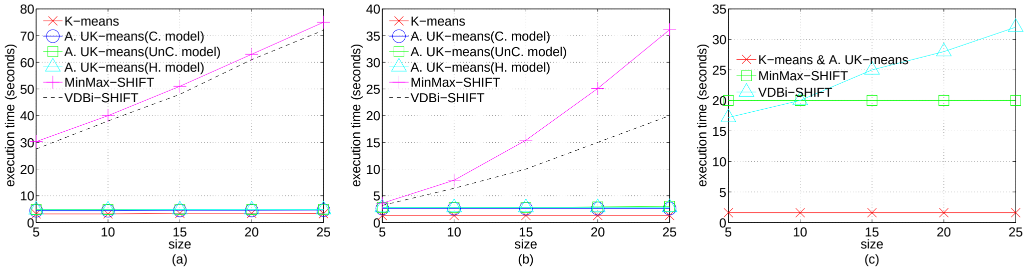

Figure 3.8 (a) Total clustering time with varying S on RDS (b) (Expected) distance calculation time with varying S on RDS (c) Overhead time with varying S on RDS.

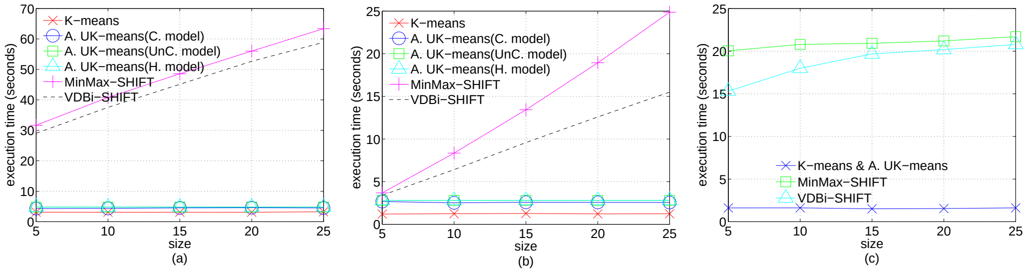

Figure 3.9 (a) Total clustering time with varying S on GDS (b) (Expected) distance calculation time with varying S on GDS (c) Overhead time with varying S on GDS.

parameters were kept at their baseline values. Figure 3.10(a) and Figure 3.11(a) show that the total execution time of all six algorithms increases as object number n increases. However, all Approximate UK-means run almost as fast as K-means, and their execution time grows much slower than MinMax-SHIFT and VDBi-SHIFT which is similar to other cases. In Figure 3.10(a) and Figure 3.11(a), the execution time of the three Approximate UK-means models is similar to each other. In Figure 3.10(b) and Figure 3.11(b) the (expected) distance calculation time of Approximate UK-means using Uncertain model (UnC. model) is similar to that of using Heuristic model (H. model), but it is longer than Certain model (C. model).

Varying Dimension Number

In the experiments, we varied the dimension number D from 2 to 6 on random data

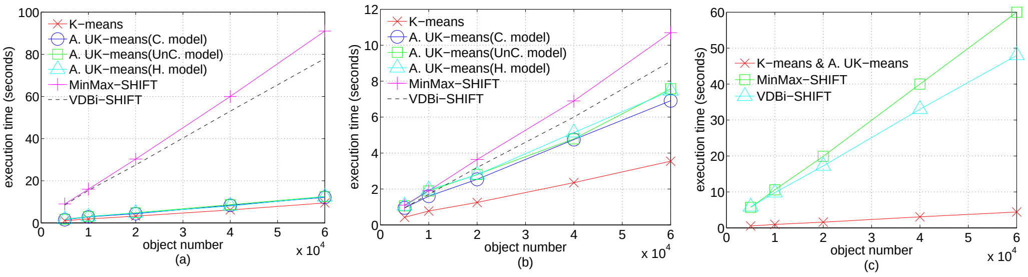

Figure 3.10 (a) Total clustering time with varying n on RDS (b) (Expected) distance calculation time with varying n on RDS (c) Overhead time with varying n on RDS.

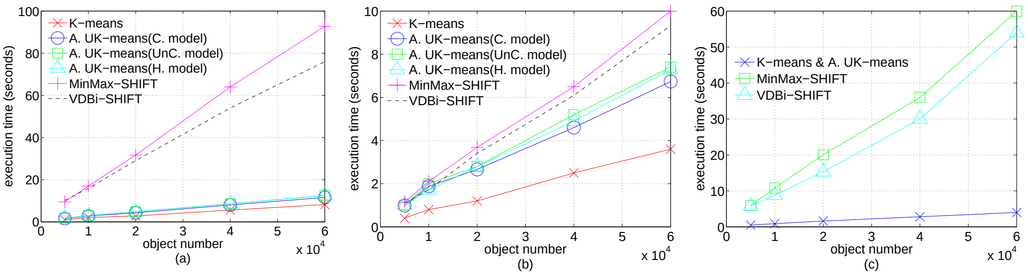

Figure 3.11 (a) Total clustering time with varying n on GDS (b) (Expected) distance calculation time with varying n on GDS (c) Overhead time with varying n on GDS.

sets. The other parameters were kept at baseline values. Figure 3.12(a) shows the total execution time of the six algorithms grows as D increases on random data sets. All three Approximate UK-means models run almost as fast as K-means and their execution time grows much slower than MinMax-SHIFT and VDBi-SHIFT. The significant improvement in the performance of Approximate UK-means is due to two reasons: (i) the distance calculations are done much faster in low dimension space (Figure 3.12(b)), and (ii) the overhead is much reduced as no pruning is necessary (Figure 3.12(c)).

Varying σ

In this experiment, we varied the standard deviation σ per cluster from 8 to 40

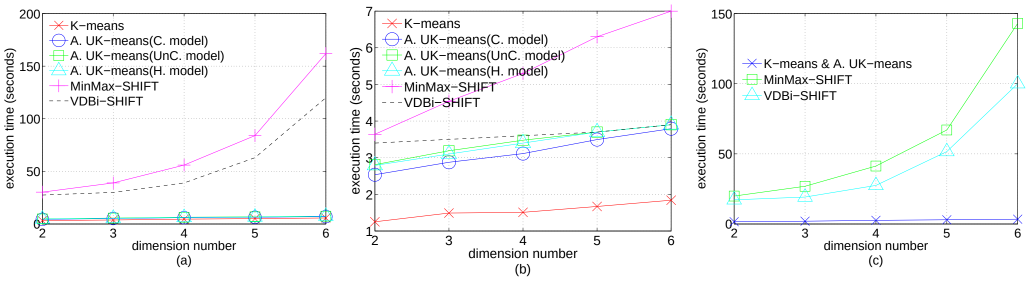

Figure 3.12 (a) Total clustering time with varying D on RDS (b) (Expected) distance calculation time with varying D on RDS (c) Overhead time with varying D on RDS.

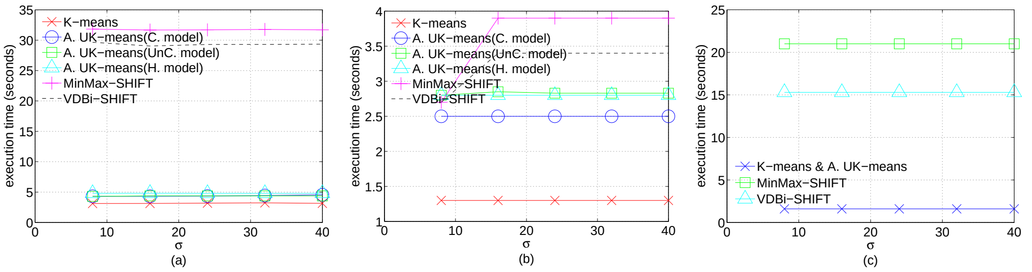

Figure 3.13 (a) Total clustering time with varying σ on GDS (b) (Expected) distance calculation time with varying σ on GDS (b) Overhead time with varying σ on GDS.

on Gaussian data sets. The other parameters were kept at baseline values. Figure 3.13(a) shows that the total execution time of all the algorithms does not increase as σ increases, and the execution time in all Approximate UK-means and K-means is much faster than that of MinMax-SHIFT and VDBi-SHIFT. Figure 3.13(b) and (c) show that the time of distance calculation and overhead in Approximate UK-means and K-means is much faster than that of pruning techniques.

From Figure 3.4(a) to 3.13(a), the execution time in all Approximate UK-means and K-means increases much slower than MinMax-SHIFT and VDBi-SHIFT, and the total execution time of all the three Approximate UK-means models is similar to each other. In Figures 3.4(b)-3.13(b), we can see that the (expected) distance calculation time of Approximate UK-means using Uncertain model (UnC. model) is similar to that of using Heuristic model (H. model), but costs a bit more time than certain model (C. model). In Figures 3.4(c)-3.13(c), the overhead time used for pruning in MinMax-SHIFT and VDBi-SHIFT occupies a large portion of total execution time, which is the new bottleneck of pruning UK-means, and the overhead time in all Approximate UK-means are the same for different β values. The overhead of Approximate UK-means is the mean vector calculation of uncertain objects. The three Approximate UK-means have the same overhead time.

3.5.3 Discrepancy of Clustering Results

This section compares the clustering results of Approximate UK-means using three different models with UK-means (pruning algorithms) by *discrepancy*. **It is emphasized that there is no "true" or "correct" clustering result because even the K-means running on traditional certain objects may give different clustering results by using different distance metrics or different seeds. In fact, the discrepancy between the results of UK-means and K-means is due to the different distance metrics used. It does not mean that the clustering result by K-means is *wrong*. Therefore, here we would like to only point out the changes that K-means may bring to UK-means (given the same seeds) and also how much Approximate UK-means may reduce these changes.**

For each comparison, we ran Approximate UK-means using three different models of cluster representative with different β values ranging from 0.7 to 1 to study the effect of β on the discrepancy. Note that when β equals 1, all Approximate UK-means is reduced to K-means (the variance of cluster representatives was initiated to be zero). Figure 3.14 and Figure 3.15 show five data sets where the

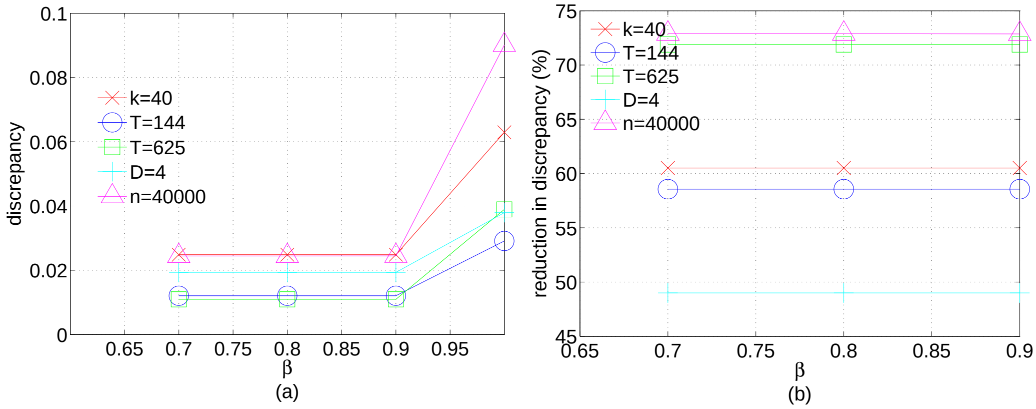

Figure 3.14 (a) Discrepancy of Approximate UK-means using certain cluster representative and K-means with respect to pruning UK-means (b) Reduction in discrepancy of clustering results of Approximate UK-means using certain cluster representative with respect to K-means on RDS as β varies.

discrepancy were reduced most by Approximate UK-means on random data sets and Gaussian data sets respectively. The pairs of Figure 3.16 and Figure 3.17, Figure 3.18 and Figure 3.19 are similar to Figure 3.14 and Figure 3.15 which also show five data sets with the discrepancy reduced most by uncertain and heuristic models. The settings of the 5 lines are on the baseline except the parameter specified in the figures. Figure 3.14(a)-3.19(a) show the discrepancy between the clustering results of Approximate UK-means and UK-means (also K-means and UK-means at $\beta = 1.0$). In Figures 3.14(b)-3.19(b), the algorithm is K-means and the reduction in discrepancy with respect to itself is zero at $\beta = 1.0$, so we eliminate the point $\beta = 1.0$. Figures 3.14(b)-3.19(b) show the reduction in discrepancy of Approximate UK-means with respect to K-means. The figures show that the reduction is stable as β decreases. Approximate UK-means tries to identify boundary cases and re-assign them to reduce discrepancy.

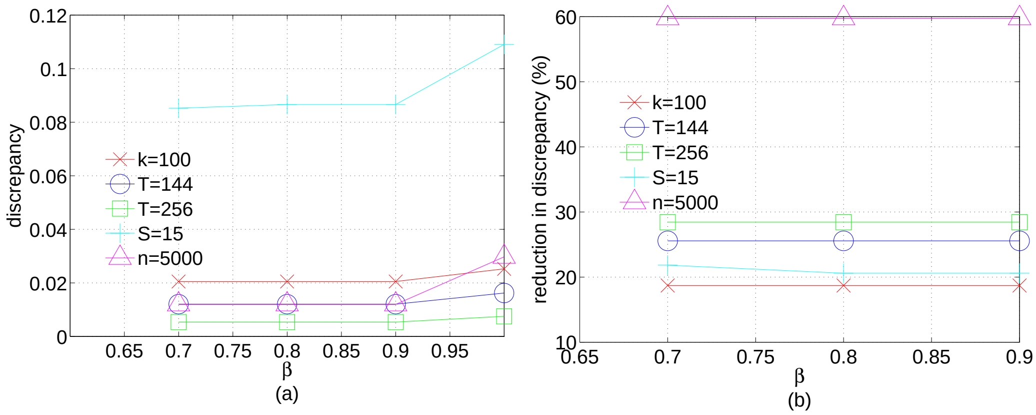

Figure 3.15 (a) Discrepancy of Approximate UK-means using certain cluster representative and K-means with respect to pruning UK-means (b) Reduction in discrepancy of clustering results of Approximate UK-means using certain cluster representative with respect to K-means on GDS as β varies.

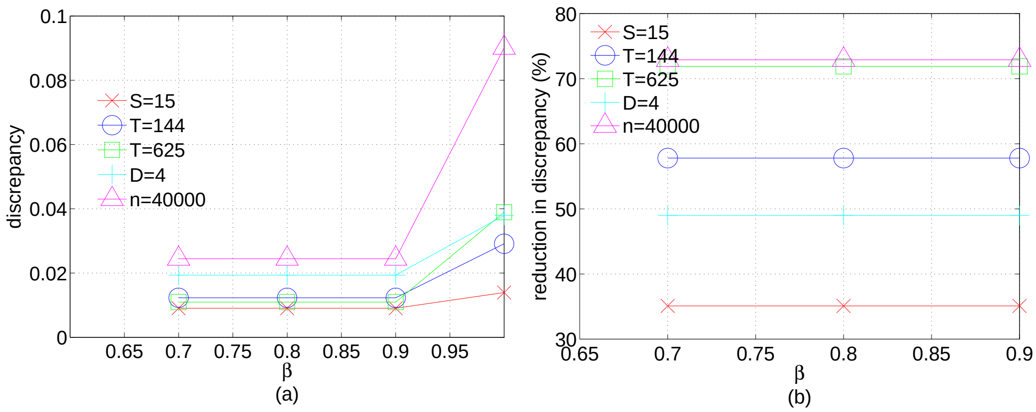

Figure 3.16 (a) Discrepancy of Approximate UK-means using heuristic cluster representative and K-means with respect to pruning UK-means (b) Reduction in discrepancy of clustering results of Approximate UK-means using heuristic cluster representative with respect to K-means on RDS as β varies.

Table 3.2 and Table 3.3 show the additional reduction in discrepancy of Approximate UK-means using uncertain model with respect to Approximate UK-means

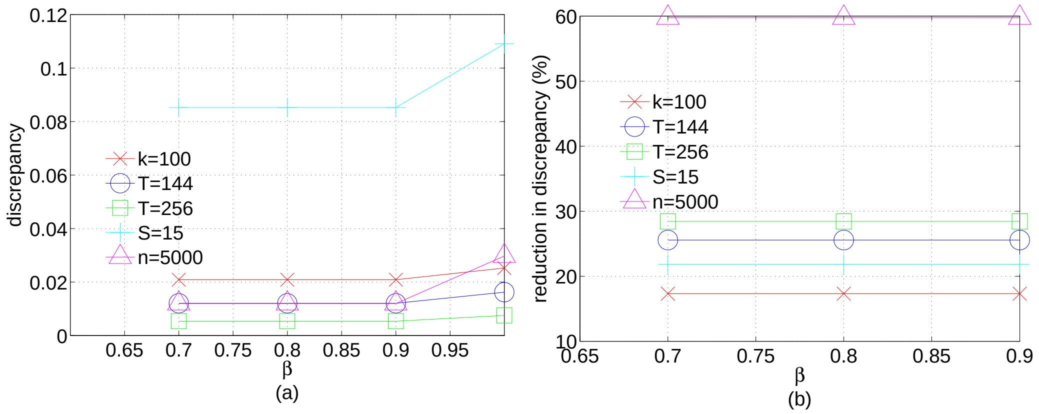

Figure 3.17 (a) Discrepancy of Approximate UK-means using heuristic cluster representative and K-means with respect to pruning UK-means (b) Reduction in discrepancy of clustering results of Approximate UK-means using heuristic cluster representative model with respect to K-means on GDS as β varies.

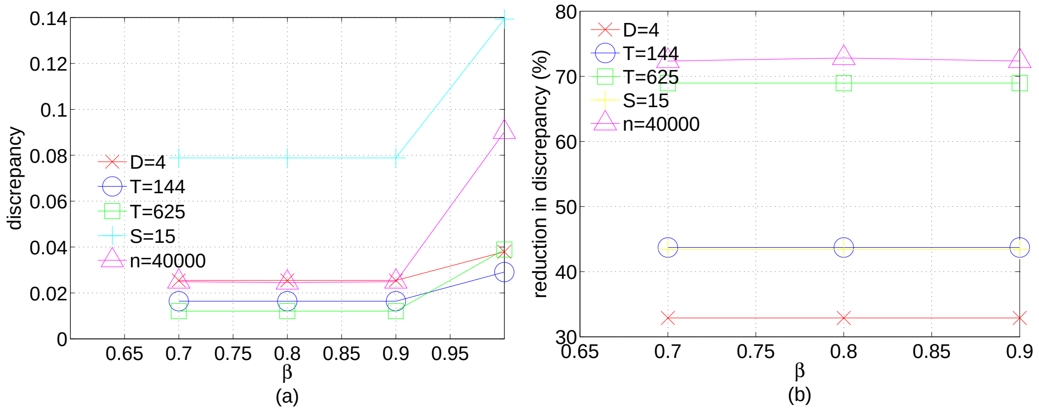

Figure 3.18 (a) Discrepancy of Approximate UK-means using uncertain cluster representative and K-means with respect to pruning UK-means (b) Reduction in discrepancy of clustering results of Approximate UK-means using uncertain cluster representative model with respect to K-means on RDS as β varies.

using certain model on some random data sets and Gaussian data sets which were improved most by uncertain model. The uncertain model of cluster representative

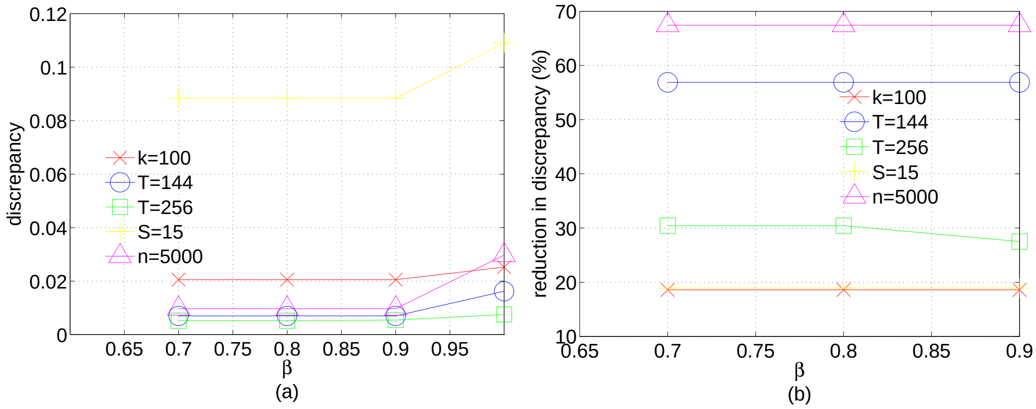

Figure 3.19 (a) Discrepancy of Approximate UK-means using uncertain cluster representative and K-means with respect to pruning UK-means (b) Reduction in discrepancy of clustering results of Approximate UK-means using uncertain cluster representative model with respect to K-means on GDS as β varies.

can reduce the discrepancy caused by uncertainty of cluster representatives. In Table 3.2 and Table 3.3, uncertain model can reduce the discrepancy with respect to certain model up to 21.9% and 77.0% on random and Gaussian data sets respectively. The MBR size of uncertain object is large, and the object is more uncertain which makes the cluster representative more uncertain.

From the experiments, (i) our experimental results show that on average the execution time of Approximate UK-means is only 25% more than K-means (while pruning algorithms are 300% more) and our approach reduces the discrepancies of K-means' clustering results up to 70%; (ii) Approximate UK-means using uncertain model can additionally reduce the discrepancy of Approximate UK-means using certain model up to 77% with only a little more execution time on some data sets.

Table 3.2 Reduction in discrepancy of clustering results of Approximate UK-means (Uncertain) with respect to Approximate UK-means (Certain) on RDS.

Parameter	discrepancy (certain model)	discrepancy (uncertain model)	Reduction rate
$D = 3$	0.04853	0.0379	21.9%
$S = 15$	0.08647	0.07855	9.2%
$k = 100$	0.04896	0.04507	7.8%

Table 3.3 Reduction in discrepancy of clustering results of Approximate UK-means (Uncertain) with respect to Approximate UK-means (Certain) on GDS.

Parameter	discrepancy (certain model)	discrepancy (uncertain model)	Reduction rate
$k = 100$	0.02051	0.00472	77.0%
$T = 144$	0.01205	0.00698	42.1%
$S = 15$	0.21812	0.17684	18.9%

3.6 Summary

This chapter proposed a heuristic on efficient and effective clustering on uncertain data.

After applying the analytic solution in [104] to reduce UK-means to K-means, we experimentally show that K-means performs much faster than existing pruning algorithms proposed in [58, 73] with some discrepancies in clustering results due to different distance functions used. We propose Approximate UK-means to heuristically identify objects of boundary cases and re-assign them to better clusters. In addition, we consider the uncertainty of cluster representative. The expected distance calculation considering the uncertainty of cluster representative is a bit slower than certain model. The best reduction of discrepancy of uncertain model is 21.9% and 77.0% with respect to certain model on random data sets and Gaussian data sets respectively.

Our experimental results show that Approximate UK-means using reduces the discrepancies of K-means' clustering results up to 70% with the execution time only 25% more than K-means (while pruning algorithms are 300% slower than K-means). The significant improvement in the speed is due to (i) the distance calculation is done much faster, (ii) the overhead is much reduced as no pruning is necessary, and (iii) it only needs to load samples in the beginning while pruning algorithms needs to load them for every expected distance calculation.

Chapter 4

Distance Based Classification on Uncertain Objects

In this chapter, we develop a classification framework to classify uncertain objects.

The aim of classification on uncertain data is to predict the label of a testing instance from a set of class labels. Though some algorithms have been extended to classify uncertain information, the problem of building classifiers on uncertain data is still a challenge. The algorithms take quite a long time to process uncertain data because of intensive computational bottleneck.

In this chapter, we built an efficient classifier based on UK-means to handle uncertain data which is much faster than other algorithms. Additionally, we capture related properties of data by extending supervised UK-means to feature selection. We also extend supervised UK-means to ensemble model based on Adaboost [38]. We first give an introduction of classification on uncertain objects in Section 4.1. Two approaches (*Averaging* and *Distribution-based*) to handle uncertain objects and feature selection on uncertain objects are described in detail in Section 4.2. The ensemble learning is introduced in Section 4.3. The experimental evaluation on the performance of the algorithms are shown in Section 4.4. Finally we summarize this chapter in Section 4.5.

4.1 Introduction

The classification of uncertain data can be illustrated by the following simple realistic example. We want to learn a model to classify the climate type of cities. The temperature is uncertain during a day, or a month even at the same place. We have to consider the uncertainty of temperature when we learn the model. A lot of problems have been proposed in classification, but most of them focus on certain data [42, 81]. Similar to clustering on uncertain objects, the problem of classifying objects also considers multidimensional uncertainty where an object is represented by an uncertain region over which a discrete probability distribution function (PDF), or a probability density function (pdf), is defined. Formally, we consider a training set of N labeled objects o_i, $1 \leq i \leq N$ in an m-dimensional space. o_i is associated with a class label $c_j(c_j \in L)$, where L is a class label set. Hence, our task is to construct a model that is able to predict the label of uncertain object correctly.

4.2 Supervised UK-means

Classification is an important task in machine learning. [98, 99] shows that the learning based on similarity metrics can perform better in multimedia data classification and retrieval. Similarly to [98, 99], we learn a model based on expected Euclidean distance in UK-means. In supervised model, there are a set of N training objects $o_1, o_2, ..., o_N$, and m numerical (real-valued) feature attributes $A_1, ..., A_m$. The domain of attribute $A_u(1 \leq u \leq m)$ is $dom(A_u)$. Each o_i is associated with a probability density function (pdf $f_i(x)$) and a class label c_j ($c_j \in L$, where L is the set of all class labels), where x is a possible location of o_i, and $UD(o_i)$ is uncertain domain of o_i. Each tuple x is associated with a feature vector $x = (\tilde{x_1}, \tilde{x_2}, ..., \tilde{x_m})$, where $\tilde{x_u} \in dom(A_u)(1 \leq u \leq m)$. The goal of supervised UK-means is to find K class representatives which can predict a testing object o_{test} to class c_k with the minimum expected Euclidean distance.

In supervised UK-means, the initial class representative is obtained by the mean vector of training objects associated with class labels. Then, we predict the labels of testing objects with the minimum expected Euclidean distance between objects and class centers. To obtain more accurate classification results, we repeat the testing process and update the class representatives until the algorithm converges. Algorithm 5 shows the generalized supervised UK-means, where N is the number of training objects, n is the number of testing objects, and K is the number of class labels.

In [82, 95], *Averaging* Approach and *Distribution-based* Approach are proposed to modify decision tree algorithm to handle uncertain data. Similarly, we also use these approaches to handle uncertain objects in supervised UK-means. In addition, we try to capture relevant properties on uncertain data by feature selection.

Algorithm 5 Supervised UK-means Algorithm

```
for i=0; $i < N$; i++ do
    compute $\overline{o_i}$ of training objects by Equation (3.4);
end for
for j=0; $j < K$; j++ do
    calculate all class representatives mean vectors $\overline{p_{c_j}}$ by Equation (3.8);
end for
for i=0; $i < n$; i++ do
    compute $\overline{o_i}$ of testing object by Equation (3.4);
end for
repeat
    for i=0; $i < n$; i++ do
        for j=0; $j < K$; j++ do
            compute expected distance between testing object and class representative $EED(o_i, p_{c_j}) = \int_{UD(o_i)} f_i(x) ED(o_i, p_{c_j})\, dx$;
        end for
        assign object $o_i$ to the nearest class $c_k$;
    end for
    update all cluster representatives by Equation (3.8);
until all cluster representatives converge
```

4.2.1 Averaging Approach

A straightforward method to deal with uncertain object is to replace each pdf with its expected value [82, 95]. Then the object is converted into exact value object, which reduces the problem back to that for certain data. Originally, object o_i is represented by T grid cells with pdf as Figure 3.1 shown. In Averaging approach, the expected distance between o_i and class representative p_{c_j} is the exact distance between the mean vector $\overline{o_i}$ of o_i and the mean vector $\overline{p}_{c_j}$ of p_{c_j}. $\overline{o_i}$ and $\overline{p}_{c_j}$ are calculated as Equation (3.4) and Equation (3.8). In Averaging approach, line 13 of Algorithm 5 is changed to $EED(o_i, p_{c_j}) = ED(\overline{o_i}, \overline{p}_{c_j})$ (where ED is Euclidean Distance). The time complexity of averaging approach is $O(1)$. Averaging approach is time saving, but loses some uncertain information.

4.2.2 Distribution-based Approach

In distribution-based approach, the training and testing object are uncertain, and are represented by samples based on their own distributions. The difference between Averaging and Distribution-based approach is the calculation method of expected distance between o_i and p_{c_j}. In this approach, line 13 of Algorithm 5 is replaced by $EED(o_i, p_{c_j}) = \sum_{t=1}^{T} F_i(s_{i,t}) ED(s_{i,t}, \overline{p}_{c_j})$, which is the same as that in UK-means. The time complexity of distribution-based approach is $O(T)$, where T is the number of samples of an object. Our distribution-based approach is different from that in [79], their work aims at summarizing the value distribution on identifier attributes (categorical attributes). The value distribution in [79] is based on all the values of the identifier attribute in the data set. As mentioned before, we focus on numerical attributes, and each value is described by T samples. The samples of value are distributed by a certain probability distribution (i.e. uniform distribution).

Weighted Expected Euclidean Distance

We select features in *Distribution-based* supervised UK-means. Here weighted expected distance ($||.||_w$) is calculated instead of expected Euclidean distance. $||.||_w$ is calculated sample by sample. The weighted distance between $s_{i,t}$ and p_{c_j} is calculated as Equation (4.1), where $s_{i,t,u}$ is the t-th sample of object o_i on u-th dimension, and w_u is the weight factor w on the u-th dimension. Furthermore, $||.||_w$ is calculated as Equation (4.2), which is the weighted average of weighted Euclidean distance between sample $s_{i,t}$ of object o_i and class center p_{c_j} ($\overline{p}_{c_j}$ is the mean vector of p_{c_j}).

$$||s_{i,t} - p_{c_j}||_w = \sqrt{\sum_{u=1}^{m} w_u^2 \times (s_{i,t,u} - \overline{p}_{c_{j,u}})^2}. \tag{4.1}$$

$$||o_i - p_{c_j}||_w = \sum_{t=1}^{T} F_i(s_{i,t})||s_{i,t} - \overline{p}_{c_j}||_w. \tag{4.2}$$

Feature Selection

In previous supervised UK-means, all features are considered to be equally important. To build a classifier with high accuracy, it is necessary to select features from feature set to capture the relevant properties on uncertain data. There have been quite a lot of feature selection techniques to make the classifier more compact and accurate. In [8], modified K-means is combined with Simulated Annealing to adapt the weight of each feature. Feature selection in [39] maximizes the margins between objects from different classes. [8] and [39] focus on exact value data, and averaging approach converts uncertain objects into deterministic point objects. So the existing algorithms can be readily used, but they cannot be directly used to distribution-based approach. Here we just extend distribution-based approach to feature selection based on [39].

Definition 1 *Let Q be a set of uncertain objects and $o_q \in Q$. Let w be a weight* *vector over the feature set A, then the distance function of o_q is*

$$\theta(o_{q,w(A)}) = (\sum_{j=1, j \neq C(o_q)}^{K} ||o_q - p_{c_j}||_w) - ||o_q - p_{C(o_q)}||_w. \tag{4.3}$$

where $||.||_w$ is weighted distance and has been described in detail before.

Definition 1 defines $w(A)$ to indicate the weight values on the feature set A. $\theta(o_{q,w(A)})$ can also be written as $\theta(o_{q,w})$. Similar to Simba [39], to make $\theta(o_{q,\lambda w}) = |\lambda|\theta(o_{q,w})$ for any scalar λ, w is normalized in the way that max $w_u^2 = 1(1 \leq u \leq m)$ (where w_u is the u-th value of $w(A)$, m is the number of attributes) to guarantee that $||.||_w^2 \leq ||.||^2$, where $||.||$ is Euclidean distance when $w = (1, ..., 1)$[1].

Definition 2 *Given a training set N ($Q \subseteq N$) and a weight vector w, the distance based evaluation function is*

$$e(w) = \sum_{o_i \in N} \theta(o_i, w). \tag{4.4}$$

Definition 2 gives the evaluation function of feature selection. In the function, we aim to make all the objects in the training set N nearest to the class that they are labeled and farthest to other classes. The task of feature selection is to find $w(A)$ that can maximize the evaluation function $e(w)$.

Distance Based Feature Selection Algorithm (DBFS)

To find the feature weight $w(A)$ that can maximize the distance evaluation function $e(w)$. We use gradient ascent to maximize Formula (4.4) [39], since the evaluation function $e(w)$ can be seen smooth almost everywhere ($lim_{\Delta \to 0} e(w + \Delta) = e(w)$). The gradient of $e(w)$ is shown as follows (Formula (4.5)) when it is evaluated on a

[1] The distance used in this chapter is Euclidean distance.

sample o_q:

$$\begin{aligned}\triangledown e(w)_u = \frac{\partial e(w)}{\partial w_u} &= \sum_{o_q \in N} \frac{\partial \theta(o_q, w)}{\partial w_u} \\ &= \frac{1}{2} \sum_{o_q \in N} ((\sum_{j=1, j \neq c(o_q)}^{K} \\ & \frac{\|o_q - p_{c_j}\|^2}{\|o_q - p_{c_j}\|_w}) - \frac{\|o_q - p_{C(o_q)}\|^2}{\|o_q - p_{C(o_q)}\|_w}) w_u.\end{aligned} \tag{4.5}$$

Similar to Simba [39], we use gradient over $e(w)$ to obtain $\triangledown e(w_u)$ (we write as $\triangledown_u$ for simplicity). A subset Q is randomly picked from training set N to evaluate $e(w)$. In each iteration we use one object to calculate one term of the vector $\triangledown$ and add it to the weight vector w. We have illustrated that the evaluation of $\triangledown$ is invariant (i.e. $\triangledown_u = \triangledown e(\lambda w_u) \forall \lambda \geq 0$, see Proof 1). Therefore, since w increases by adding $\triangledown_u$ during each iteration, the algorithm typically converges by decreasing the relative effect of the term $\triangledown_u$ (divided by increasing w). Different from Simba [39], we use distance function for uncertain data in Algorithm 6.

Lemma 1 $\triangledown(\lambda w_u) = \triangledown e(w_u)$.

Proof 1 In Formula (4.6), because $\|.\|_{\lambda w} = \lambda \|.\|_w$ which has been described in weighted expected Euclidean distance, Formula (4.6) is equal to Formula (4.5).

$$\begin{aligned}\triangledown(\lambda w_u) = \frac{\partial e(\lambda w)}{\partial w_u} &= \sum_{o_q \in S} \frac{\partial \theta_{o_q}^{\lambda w}}{\partial w_u} \\ &= \frac{1}{2} \sum_{o_q \in N} ((\sum_{j=1, j \neq c(o_q)}^{K} \\ & \frac{\|o_q - p_{c_j}\|^2}{\|o_q - p_{c_j}\|_{\lambda w}}) - \frac{\|o_q - p_{C(o_q)}\|^2}{\|o_q - p_{C(o_q)}\|_{\lambda w}}) \lambda w_u.\end{aligned} \tag{4.6}$$

Selecting features on uncertain data is T (the number of samples of o_i) times more complex than that on certain data. The computational complexity of DBFS

Algorithm 6 Distance Based Feature Selection Algorithm (DBFS)

```
initialize weight vector $w = (1, 1, ..., 1)$;
pick randomly $Q \subseteq N$ when $N$ is training set;
**for** $q = 1...|Q|$ **do**
    pick an object $o_q$ from $Q$;
    **for** $u = 0; u < m; u++$ **do**
        $\nabla_u = \frac{1}{2} \sum_{o_q \in Q}((\sum_{j=1, j \neq C((o_q))}^{K} \frac{||o_q - p_{c_j}||^2}{||o_q - p_{c_j}||_w}) - \frac{||o_q - p_{(o_q)}||^2}{||o_q - p_{c(o_q)}||_w}) w_u$;
        $w_u = w_u + \nabla_u$;
    **end for**
**end for**
$w = \frac{w^2}{w_{max}^2}$;
```

is $O(m|Q|TK)$ where m is the number of features, $|Q|$ is the number of iterations (usually 20 epochs), and K is the number of class labels. If we iterate over the whole training set N, the computational complexity is $O(mNTK)$. DBFS spends time on $||.||_w$ (weighted distance) calculation. DBFS is used in distribution-based approach (DBA). In Algorithm 5, we use DBFS in the training process to evaluate the weight values on feature set A. Then, testing objects are predicted by weighted expected Euclidean distance. In Section 4.4, distribution-based approach (DBA) which uses DBFS to select features is noted as weighted distribution-based approach (Weighted DBA for short).

4.3 Ensemble Learning Model on Uncertain Objects

UK-means is proposed for clustering. In its application of classification, it is not enough for the class representative p_{c_j} (mean vector of the class) to represent a class. In this case, there always exist some objects which are closer to other class representatives.

4.3.1 The Motivation of Ensemble Supervised UK-means

Figure 4.1 gives an example. In Figure 4.1, the red rectangle objects (o_1 (0,0), o_2 (3,0), o_3 (0,3))are belonging to class c_1. The blue circle objects (o_4 (3,3), o_5 (9,0), o_6 (12,3), o_7 (3,6)) are from class c_2. The class representative of c_1 ($\overline{p}_{c_1}$) is (1,1) and the class representative of c_2 ($\overline{p}_{c_2}$) is at point (6.75,3). From the training set, supervised UK-means learned the classifier L_1 with $\overline{p}_{c_1}$ (1,1) and $\overline{p}_{c_2}$ (6.75,3). Then in the classifying process, we assign the training objects based on L_1 and get the result that $o_1, o_2, o_3, o_5, o_6, o_7$ are assigned correctly except o_4 assigned to c_1 belonging to c_2. Thus, the classifier should be improved for the objects belonging to a class but closer to other class representatives.

From the simple instance, we can see that it is difficult for the mean vector of each class (class representative) to classify the objects. In this chapter, we also try to use the ensemble model to combine the weak learners obtained from Adaboost. This framework constructs a number of classifiers from the training objects. The key idea of the method is to combine the weak learners together and the decision is made by weighted voting of the classifiers.

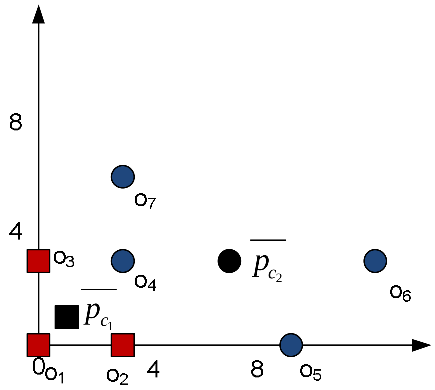

Figure 4.1 The example of misclassification in supervised UK-means

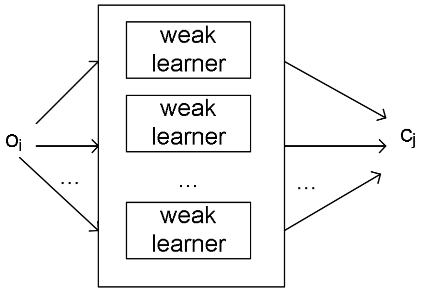

Figure 4.2 The structure of ensemble learning model

4.3.2 Typical Ensemble Learning

Ensemble learning constructs a series of classifiers and predicts the testing objects by (weighted) voting of the classifiers. The structure of the ensemble learning method is shown as Figure 4.2. The object o_i is predicted by weak learners, and the decision is made based on the weighted vote of all the predictions obtained from the weak learners. Ensemble learning has been well studied in the literature [83, 103, 107].

Bagging

Bagging straightly manipulates the training set by randomly drawn N training samples from the original training set. The original training set contains N training objects. The randomly generated training set that contains N training samples is bootstrap replicate of the original one. In each replication, there are 63.2% objects from the original training set appearing more than one time on average [20].

AdaBoost

AdaBoost (Short for Adaptive Boosting) [38] has been widely used to with other algorithms and can improve the performance of other learning algorithms. In Adaboost, subsequent classifiers are built adaptively by in favor of objects that are misclassified in previous classification procedure. In Adaboost, the weak classifiers are learned several times. During each time, the weights of objects in the data set is updated to indicate the importance of training objects for classification. The weights of misclassified objects are increased while the weights of correctly classified objects are decreased. Adaboost can make new classifier focus more on misclassified objects.

4.3.3 Adaptive Supervised UK-means

The ensemble model for supervised UK-means is based on Adaboost [38]. In Adaptive supervised UK-means, each subsequent classifier can be obtained in favor of the objects which are misclassified in previous classifiers. Algorithm 7 shows the Adaptive Supervised UK-means in detail, where $h_u(o_i) \neq C(o_i)$ means o_i is misclassified by h_u ($C(o_i)$ is the class label that object o_i belongs to, $h_u(o_i)$ is the class o_i assigned to by learner h_u). For the misclassified object o_i, the value of $M(o_i)$ is -1. Otherwise, the value of $M(o_i)$ is 1. h_u assigns objects by the minimum expected Euclidean distance. The (weighted) vote is made by $h(o_i) = \sum_{u=1}^{U}(\alpha_u h_u(o_i))$, where

$\alpha_u = ln\frac{1-\epsilon_u}{\epsilon_u}$ (ϵ_u is error rate of h_u). For example, with the assumption $\alpha_1 = 0.5$, $h_1(o_i) = c_1$, $\alpha_2 = 0.3$, $h_2(o_i) = c_1$, $\alpha_3 = 0.2$, $h_1(o_i) = c_2$, o_i is assigned to c_1 by $\alpha_1 + \alpha_2 > \alpha_3$.

In Algorithm 8, the class representative is calculated by weighted mean vector of object which is different from Equation (3.8). Each object is associated with a weight $d_u(o_i)$, which is the weight of object o_i obtained from learner $h_{(u-1)}$. In Algorithm 7, we assign object o_i by the minimum expected Euclidean distance between o_i and the mean vector of class representative $\overline{p}_{c_j}$ ($\overline{p}_{c_j}$ is the output of *Learn* h_t).

Let us go back to the example of Figure 4.1, object o_4 is misclassified in supervised UK-means as we have explained before. In the first iteration, the weight of each objects is initialized as $\frac{1}{7}$. The class representatives ($\overline{p}_{c_1}$ (1,1) and $\overline{p}_{c_2}$ (6.75,3)) misclassify o_4. The new weight of object o_4 is $\frac{6}{7}$ while the new weight of other objects is $\frac{1}{42}$ ($\frac{1}{7}exp(-ln6 \times M(o_i))$). The new classifier is $\overline{p}_{c_1}$ (1,1) and $\overline{p}_{c_2}$ ($\frac{44}{13}$,3) obtained by Formula (4.7). All the objects are classified correctly and ϵ_2 is 0 which makes $\alpha_2 = ln\frac{1}{0}$ positive infinite. Then the second classifier will decide the result. Thus, the ensemble model of Figure 4.1 can assign all objects correctly. This is a special instance because of the 100% accuracy of the second classifier with $\alpha_2 = +\infty$.

4.4 Performance Evaluation

A set of experiments has been performed to compare supervised UK-means with UDT [95]. All codes were written in Java and were run on a Windows machine

Algorithm 7 Adaptive Supervised UK-means based on Adaboost [38]

Input: a set S of N labeled objects $(o_i, c_j), i = 1, 2, ..., N$, class label c_j ($j \in 1, 2, ..., K$), Learn (a learning algorithm), a constant U.

Output: $h(o_i) = \sum_{u=1}^{U}(\alpha_u h_u(o_i))$ (where $h_u(o_i)$ is the decision of object o_i made by learner h_u).

```
for i=0; $i < N$; i++ do
    compute $\overline{o_i}$ of training objects by $\overline{o_i} = \sum_{t=1}^{T} s_{i,t} \times F_i(s_{i,t})$;
end for
for i=0; $i < N$; i++ do
    Initialize $d_1(o_i) = \frac{1}{N}$;
end for
for u=1;$u \leq U$; u++ do
    compute all normalized weight $d_u(o_i) = d_u(o_i) / \sum d_u(o_i)$;
    $h_u = learn(d_u)$;
    $\epsilon_u = \sum_i d_u(o_i)[h_u(i) \neq C(o_i)]$;
    if $\epsilon_u \geq \frac{1}{2}$ then
        stop;
    else
        $\alpha_u = ln\frac{1-\epsilon_u}{\epsilon_u}$;
        update all $d_{u+1}(o_i) = d_u(o_i)e^{(-\alpha_u M(o_i))}$;
    end if
end for
```

with an Intel 2.66GHz Pentium(R) Dual-Core processor and 4GB of main memory.

Algorithm 8 Learn h_t

```
Input: a set S of $N$ labeled objects $\left\{(o_i, c_j), i = 1, 2, ..., N\right\}$, class labels $c_j$ ($j \in \{1, 2, ..., K\}$), the weight of $N$ objects $\{d_u(o_i), i = 1, 2, ..., N\}$.
Output: weak learner $h_u$.
1: for $j = 0; j < K; j++$ do
2:     obtain weak learner by calculating mean vector $\overline{p}_{c_j}$ of $p_{c_j}$ (Formula (4.7));
           $$\overline{p}_{c_j} = \frac{1}{\sum_{i=1}^{|c_j|} d_u(o_i)} \sum_{i=1}^{|c_j|} (d_u(o_i) \times \overline{o}_i)(o_i \in \left\{o_i | C(o_i) = c_j\right\}). \quad (4.7)$$
3: end for
4: output $h_u = (p_{c_1}, p_{c_2}, ..., p_{c_K})$.
```

4.4.1 Data Sets

We run experiments on 4 UCI [11] data sets to study the performance of our algorithms and compare with the work in [95]. The parameters of the chosen data sets used for the experiments are summarized in Table 4.1. The attributes of all the 4 data sets are numerical obtained from measurements. Classifiers are built on the numerical attributes and their “class label” attributes. For the data sets “Iris” and “Breast Cancer”, the accuracy is measured by 10-fold cross validation. For other 2 data sets, a certain number of objects are chosen randomly as testing objects and the performance results are the average of 10 runs.

The 4 data sets contain “point values” without any uncertainty. In most physical measures, the involved random noise follows Gaussian distribution. The quantization noise introduced by digitization of the measured values is described by uniform distribution. Thus, we follow the common practice in the research work of

Table 4.1 Selected data sets from the UCI machine learning repository.

Data Set	Training Tuples	No. of Attributes	No. of Classes	Test Tuples
Iris	150	4	3	10-fold
BreastCancer	569	30	2	10-fold
Ionosphere	311	32	2	40
Segment	2120	14	7	200

this area [5, 22, 58, 73, 82, 94, 95]. For each object o_i on the u-th dimension (i.e. the attribute A_u), the certain value $v_{i,u}$ shown in a data set is considered as the mean of a pdf $f_{i,u}$, defined over an interval $[a_{i,u}, b_{i,u}]$. The range of values for A_u (over the whole data set) is noted and the width of $[a_{i,u}, b_{i,u}]$ is set to $un \times |A_u|$, where $|A_u|$ denotes the width of the range for A_u and un is a parameter to control the uncertainty of data set [82, 94, 95]. We use two methods to generate pdf $f_{i,u}$. One is uniform distribution, which implies the pdf to be $f_{i,u} = (b_{i,u} - a_{i,u})^{-1}$. The other is Gaussian distribution, which we set the standard deviation to be $\frac{1}{4} \times (b_{i,u} - a_{i,u})$ (the same as that in [95]). We use T samples to generate pdf over the interval. The point value is transformed into uncertain samples on Gaussian or uniform distribution by using the controlled parameter un and T samples. To compare with the work in [95], T is set to be 100 and un is from 1% to 20%. The point-value data become uncertain when we apply appropriate error model (un and pdf) for them.

In many real applications (such as real time system), it is difficult for users to tolerate the system if a classification algorithm keeps the users waiting for a long time. Thus, it will be more desirable to improve the efficiency with the trade off of some loss in accuracy.

4.4.2 Execution Time

We separately analyze the “training time” and “testing time”. Table 4.2 and Table 4.3 show the training time and testing time of Averaging (AVG) approach, Distribution-based approach (DBA), Weighted DBA, Ensemble DBA (Adaptive Supervised UK-means) compared with total time of uncertain decision tree (UDT) [95] on 4 selected UCI data sets. AVG can be considered as supervised UK-means for point value data while DBA, Weighted DBA and Ensemble DBA are supervised UK-means to handle uncertain objects. Ensemble DBA costs time on learning classifiers and calculating the weight of objects. In the experiments, *un* does not affect the execution time, so here we use $un = 0.05$. Because only total time is given in UDT [95], we show the total time of UDT in Table 4.2 and Table 4.3. Table 4.2 and Table 4.3 tell us that training time is larger than testing time in AVG, DBA, Weighted DBA and Ensemble DBA. We can also see that the total time of AVG, DBA, Weighted DBA and Ensemble DBA is shorter than that of UDT. On the 4 data sets, AVG is at least 98 times faster than UDT while DBA is at least 15 faster than UDT. Weighted DBA and Ensemble DBA are at least 4 and 6 times faster than UDT.

In Table 4.2 and Table 4.3, Distribution-based approach (DBA) and Weighted DBA are slower than AVG because of the expected distance calculation between uncertain objects and class centers. Weighted DBA has to calculate the weight values over feature set sample by sample which is a bit time consuming. However, the objects used to evaluate weight are chosen from the subset of training set, so

Table 4.2 Training time (Milliseconds).

Data Set	AVG	DBA	Weighted DBA	Ensemble DBA	UDT
Iris	9.2	36	79.7	118.7	909.9
BreastCancer	12.4	331.2	855.7	928.1	8363.6
Ionosphere	18.9	226.5	851.5	648.3	4272.7
Segment	76.4	3035.8	3796.8	7715.7	59090.9

Table 4.3 Testing time (Milliseconds).

Data Set	AVG	DBA	Weighted DBA	Ensemble DBA	UDT
Iris	0	1.5	3.1	7.7	909.9
BreastCancer	1.6	61.0	78.0	85.9	8363.6
Ionosphere	3.1	47.0	60.6	59.2	4272.7
Segment	12.5	490.1	588.3	859.2	59090.9

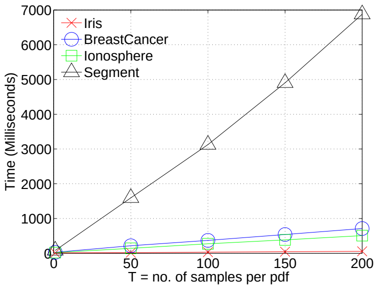

Figure 4.3 Effects of increasing T on Distribution Based Approach

the weight calculation does not take more time than the time used in information gain in UDT. Ensemble DBA costs more time than other DBAs because of the time costing on a series of weak learners. In the experiments, the distribution of samples (uniform or Gaussian distribution) does not affect the execution time of uncertain algorithms. Thus, we just presents the execution time of uncertain algorithms on Gaussian distribution here.

Scalability on T Uncertain object is T times more complex than certain object. If T increases, more time will be cost. The scalability of algorithms on T is shown from Figure 4.3 to Figure 4.5. The time of all the three algorithms increases linearly with T. Moreover, our algorithms are faster than UDT, and the time of our algorithms does not increase as fast as UDT [95] when T increases. From Figure 4.3 to Figure 4.5, we can see our algorithms are scalable on T. Moreover, the scalability of our algorithms are better than UDT.

4.4.3 Accuracy

Figure 4.6 and Figure 4.7 show the accuracy with changing uncertainty *un* under different values of Weighted DBA and Ensemble DBA. In UDT, the accuracy also

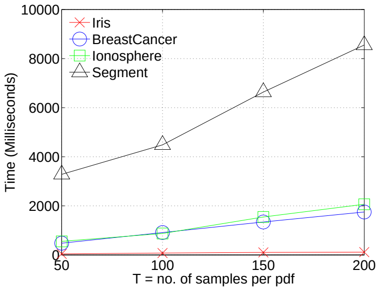

Figure 4.4 Effects of increasing T on Weighted Distribution Based Approach

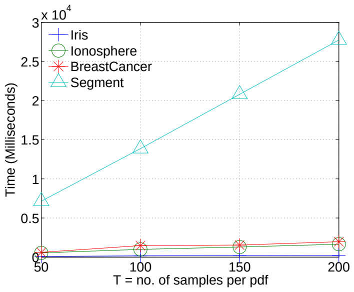

Figure 4.5 Effects of increasing T on Ensemble Distribution Based Approach

changes with the setting of *un*. To compare the effect of *un*, we put the accuracy of AVG which is exact point value algorithm at $un = 0$. In AVG, the point value on each dimension of an instance is the original data from the data sets. From the figures, the accuracy is improved if uncertainty is taken into account. In addition, the controlled parameter *un* which describes the uncertainty of data sets can affect the classifier results. If the controlled parameter *un* is consistent with the uncertainty of the data, the algorithms will attain a better accuracy. For each object o_i on the j-th dimension ($|A_j|$ is the range of attribute A_j), the point value is $v_{i,j}$. If we perturb $v_{i,j}$ by adding a Gaussian noise whose mean equals to 0 and the standard deviation being $\sigma = \frac{1}{4}(un \times |A_j|)$. Thus, the perturbed value is $\tilde{v}_{i,j} = v_{i,j} + \delta_{i,j}$ ($\delta_{i,j}$ is a random generated number following $N(0, \sigma^2)$)[95]. If *un* is consistent with the error model

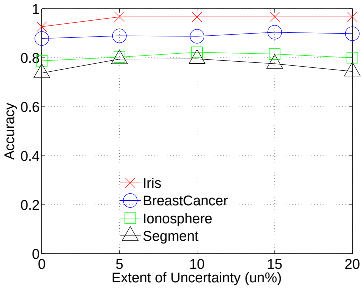

Figure 4.6 Accuracy of Weighted DBA with controlled parameter *un* (Gaussian pdf)

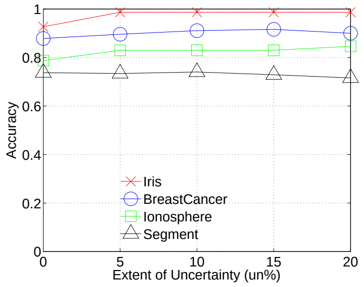

Figure 4.7 Accuracy of Ensemble DBA with controlled parameter *un* (Gaussian pdf)

of the data, the algorithm will be more accurate. From the two figures, we can see the accuracy is different as *un* changes. The closer *un* to the real noise model the better accuracy will attain.

Table 4.4 shows accuracy of our algorithms compared with UDT [95]. Table 4.4 chooses the accuracy of DBA (Gaussian pdf or uniform pdf), Weighted DBA (Gaussian pdf) and Ensemble DBA by the best results on *un*. We compare our algorithms and UDT under the same *un*. From Table 4.4, Weighted DBA and Ensemble DBA both are more accurate than DBA and AVG. AVG can be considered as classification on certain values. The experiments demonstrate that the accuracy is improved if we take value uncertainty into account. From Table 4.2, Table 4.3 and Table 4.4, our algorithms save 78%-99% time on the data sets with 5%-13% accuracy loss. In

Table 4.4 Accuracy.

Data Set	AVG	DBA (Gaussian)	DBA (Uniform)	Weighted DBA	Ensemble DBA	UDT	un
Iris	0.9267	0.94	0.9467	0.9667	0.9667	0.9467	0.2
BreastCancer	0.879	0.893	0.880	0.904	0.913	0.955	0.15
Ionosphere	0.788	0.818	0.798	0.823	0.825	0.915	0.1
Segment	0.738	0.759	0.763	0.78	0.796	0.929	0.05

many real applications (such as real time systems), the users have a higher requirement on execution time than accuracy. In “Iris”, the accuracy of DBA is competitive to UDT, and the accuracy of Weighted DBA and Ensemble DBA is better than UDT. Though the accuracy of UDT outperforms that of DBAs on the other three data sets, DBAs can still be considered if the users prefer to get an acceptable result without waiting for a long time.

4.5 Summary

In this chapter, we study the problem of classification on uncertain objects whose locations are presented by probability density functions (pdf). Supervised UK-means is more efficient than existing algorithms because it is less complex compared with them. Our contributions of this chapter include: i) we build a classifier based on UK-means and experimentally demonstrate that supervised UK-means algorithm can classify uncertain objects more efficiently than existing algorithms; ii) Considering the relevant properties of uncertain data, we extend supervised UK-means to feature selection and Adaboost.

Chapter 5

Classification on Uncertain Data with Multiple Subclasses

In this chapter, we develop a framework to classify uncertain objects with multiple subclasses in this chapter.

Supervised UK-means assumes the classes are well separated. However, in classification, the classes are often in arbitrary shape which makes the boundary between classes concave or convex but not a single line. In this chapter, we first introduce the problem of classification on uncertain data with multiple subclasses in Section 5.1. We briefly describe existing work on cluster number estimation in Section 5.2. We propose supervised UK-means with multiple subclasses (SUMS) in Section 5.3 to tackle the problem caused by objects divided by other classes. Bounded SUMS in Section 5.4 is an improvement of SUMS. Section 5.5 demonstrates the advantage of our work by extensive experiments. Finally we summarize this chapter in Section 5.6.

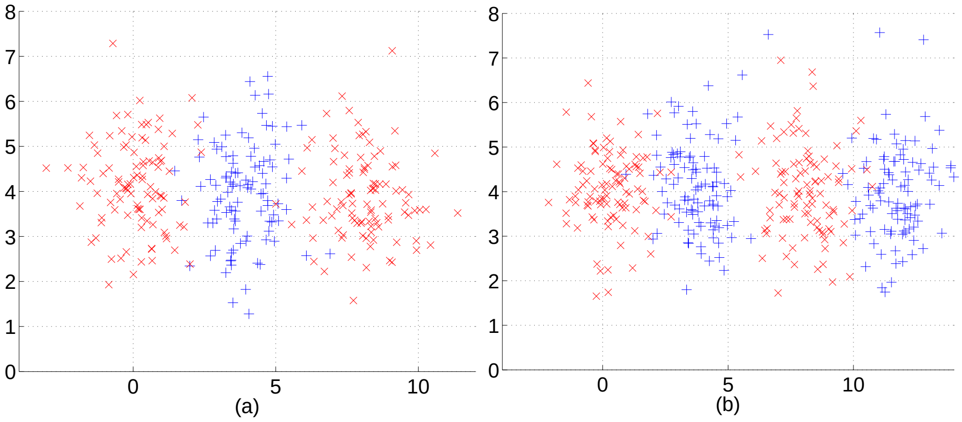

Figure 5.1 (a) An example of one class ('x') divided by another class ('+') (b) Another example of one class ('+') divided by another class ('x').

5.1 Introduction

In some cases, a class's objects are separated (disconnected) by objects from other classes. Figure 5.1(a) shows that one class (represented by 'x') is divided by another class ('+'). We denote the data set as "Middle" which means that one class is divided in the middle. Figure 5.1(b) is another example showing that each class is divided by other classes and the boundary between classes is concave and convex. We denote the example as "Side" with the meaning that each class being divided on two sides. For the above cases, our solution (supervised UK-means with multiple subclasses, or SUMS) is using a multiple class representatives to represent a class.

We consider the problem as the estimation of the number of groups or subclasses (k) in each class. The key idea of estimation of k is to use splitting and/or merging methods to increase and/or decrease the number of clusters, which makes the model fit the data. Each subclass (cluster belonging to a class) can be considered as a Gaussian mixture model. Several algorithms have been proposed to determine *k* automatically [37, 45, 78, 102]. Most of them are based on K-means or Expectation Maximization. In a classification problem, training objects are labeled by class labels. When we consider objects from the same class and further divide them into subclasses, we ignore their labels. In other words, labels of objects are considered during inter-class training, and in the process of the estimation of *k* (the number of subclasses in a class) during intra-class training, the labels of objects from the same class are ignored. On the other side, the estimation of number of subclasses will take extra time. Moreover, the estimation on uncertain objects is more complex than traditional estimation algorithms. The estimation procedure will slow down the supervised UK-means with multiple subclasses (SUMS). We propose a bounded supervised UK-means with multiple subclasses (BSUMS) by adding a bound in SUMS to improve the efficiency and avoid the number of subclasses being overestimated.

5.2 Related Work on Cluster Number Estimation

Some work has been proposed to estimate the number of clusters (subclasses) during data clustering. X-means [78] learns *k* by using K-means. The model for each *k* is obtained by trying many values of *k* in X-means. Each model is scored by Bayesian Information Criterion (BIC) and the highest one is chosen in X-means. Other scoring criteria can also be used in X-means. In X-means, the weakness is the assumption of all the cluster covariances being spherical with the same width. X-means is likely to overfit the data when the clusters are non-spherical. Bayesian K-means [102] uses Maximization-Expectation (ME) to learn a mixture model. ME maximizes over the hidden variables (assignment of examples to clusters), and computes an expectation over model parameters (center location and covariances). The algorithm works well but it is time consuming. G-means (Gaussian means) [45] is based on K-means. The key idea of G-means is projection and statistical test. In G-means, k is initialized by a small number. The cluster is split into two clusters if the objects from the cluster are not from a Gaussian distribution. G-means performs well when the clusters are separable, but it is difficult for overlapping clusters. PG-means (Projected Gaussian means) [45] is proposed to handle more complex cases, especially for overlapping clusters, non-Gaussian data, and so on. Moreover, PG-means is faster than variational Bayesian K-means. PG-means is based on projections and statistical tests to determine whether a whole mixture model fits the data well. PG-means cannot be directly used in our UK-means classification, because PG-means is used for handling certain objects. In Section 5.3.2, we will describe how to modify PG-means for our problem. To avoid overfitting of data, we add a bound to terminate estimating the number of subclasses earlier.

5.3 Supervised UK-means with Multiple Subclasses (SUMS)

5.3.1 SUMS

Algorithm 9 shows supervised UK-means with multiple subclasses (SUMS), where N is the number of training objects, K is the number of class labels, k_l is the number of subclasses of the *l-th* class.

Algorithm 9 Supervised UK-means with Multiple Subclasses (SUMS)

```
Input: training set $\{o_1, o_2, ..., o_N\}$ with class labels $c_j$ ($j \in \{1, 2, ..., K\}$).
Output: learner of subclass representatives.
for $i = 0; i < N; i++$ do
    compute $\overline{o_i}$ of training objects by Equation (3.4);
end for
for $i = 0; i < K; i++$ do
    estimate the number of subclasses ($k_i$) by PG-means and get the subclass
    representatives ($p_{c_{i_1}}, p_{c_{i_2}}, ..., p_{c_{i_{k_i}}}$) by using Algorithm 10;
end for
repeat
    for $m = 0; m < N; m++$ do
        for $i = 0; i < K; i++$ do
            for $j = 0; j < k_i; j++$ do
                compute expected Euclidean Distance by $EED(o_m, p_{c_{i_j}}) =$
                $\sum_{t=1}^{T} F_m(s_{m,t}) ED(s_{m,t}, p_{c_{i_j}})$;
            end for
        end for
        assign object $o_m$ to the subclass with the minimum $EED(o_m, p_{c_{l_q}})$;
    end for
    update all subclass representatives by Equation (5.1);
until all subclass representatives converge
```

First, Algorithm 9 calculates the mean vector of uncertain objects for the purpose of calculating the mean vector of (sub)class representative ($\overline{p}_{c_{j_k}}$). In [13], a semi-supervised model based seeded K-means is applied to K-means clustering. In seeded K-means, not all the objects are labeled, and the labeled objects are selected as seeds to generate initial cluster representatives. All the unlabeled objects and labeled objects are used for clustering until the algorithm converges. Different from the seeded K-means [13], our model is a supervised model which all the n objects are labeled. We use all the n labeled objects to generate K initial class (subclass) representatives. From Line 4 to Line 6, the number of subclass representatives is estimated and the subclass representatives are trained by the objects labeled by the same class. The number of subclasses of each class is estimated by Algorithm 10. The subclass representatives are trained by UK-means based on the objects from the same class. From Line 7 to Line 17, the objects are reassigned to the subclasses by the minimum expected Euclidean distance, then the subclass representatives are updated by the new assignment until the algorithm converges. The mean vector $\overline{p}_{c_{j_k}}$ (k-th subclass representative $p_{c_{j_k}}$ of class c_j) is obtained by Equation (5.1), where $|c_{j_k}|$ is the number of objects assigned to subclass c_{j_k}, and $C(o_i) = c_{j_k}$ means that object o_i is assigned to subclass c_{j_k}.

$$\overline{p}_{c_{j_k}} = \frac{1}{|c_{j_k}|} \sum_{o_i \in \{o_i | C(o_i) = c_{j_k}\}} \overline{o_i}. \tag{5.1}$$

Compared with the work in [13], we consider the uncertainty of objects and the substructure in classes. The time complexity of the algorithm is decided by the time spending on the estimation of k_i (the number of subclasses) and the calculation of subclass representatives. The time complexity of training subclass representatives is $O(NTk_{total})$, where N is the number of training objects, T is the number of samples of object, and k_{total} is the total number of subclasses of all classes. The time complexity of estimation of k_i (the number of subclasses) will be discussed in Section 5.3.2.

5.3.2 Estimation of k_i

PG-means is short for Projected Gaussian means. The key idea of PG-means [37] is to learn a model that contains k_i centers by Expectation Maximization algorithm. The data set and the learned model are both projected to a dimension in PG-means. Kolmogorov-Smirnov (KS) test is applied after the projection. The test is used to check the fitness of the projected model. In [37], PG-means uses standard Gaussian mixture model together with Expectation-Maximization learning while in SUMS we use UK-means instead.

Assume a data set M ($M \backsim N(\mu, \Sigma)$) is sampled from a single Gaussian cluster in d dimension, where $\mu = E[M]$ is a $d \times 1$ mean vector and $\Sigma = cov[M]$ is a $d \times d$ covariance matrix. Given a $d \times 1$ projection vector P of the unit length ($||P|| = 1$), M can be projected along P as $M^{'} = P^T M$. Then, $M^{'} \backsim N(\mu^{'}, \sigma)$, where $\mu^{'} = P^T \mu$ and $\sigma^2 = P^T \Sigma P$. The one-dimensional projection along P can be obtained by cluster model projection [37].

The following are the two hypotheses that used in PG-means [37]:

H_0: The data around the center are sampled from a Gaussian.

H_1: The data around the center are not sampled from a Gaussian.

PG-means repeats this projection and test step several times for a single learned model. If H_0 is rejected by any test with H_1 being accepted which means that the data does not follow the distribution of the model, then one more subclass is added and a new EM learning will start (UK-means in our algorithm). If the null hypothesis (H_0) for a given model is accepted by any test, then PG-means will terminate.

The univariate Kolmogorov-Smirnov (KS) test is used to calculate the fitness of projected model in PG-means after projection. The KS test statistic is $D = max_X|F(X) - S(X)|$ which means the maximum absolute difference between the true cumulative distribution function (CDF) of $F(x)$ with the sample CDF $S(X)$ [37]. The method of k_i estimation is shown as Algorithm 10. PG-means uses UK-means (Line 12) to learn a model containing k_i centers. In our work, we use UK-means instead of EM training [37]. In UK-means, object is only belonging to one subclass.

The worst case is that each object belongs to a subclass, and the time complexity of Algorithm 10 is $O(Jn^2T)$, where J is the number of projections, n is the number of objects from a class, T is the number of samples. Following the work in [37], we use random projection [32] to project both the data and the model. Other possible methods (e.g. principal component analysis) can also be used here. We try to find sufficient but not a large number of projections and tests to discover a fitting model. To make UK-means converge faster (Line 12), each UK-means starts with k_i learned subclass representatives and a new randomly initialized subclass representative. Thus, in practice, UK-means converges much faster than randomly generating all k_i + 1 subclass representatives.

In [37], they followed Dasgupta's conclusion. In Dasgupta's conclusion, Gaussian can be measured by c-separation[1] [32]. In [32, 37], they gave the conclusion

[1] For any $c > 0$, in a d dimension space, assume μ_1 and μ_2 are the two cluster centers, and the spherical covariances Σ of the two clusters are the same for simplicity, c-separation is that the vector

Algorithm 10 Estimation of k_i [37]

Input: The mean vector set (M) of objects labeled by c_i, confidence α, number of projections J.

Output: the number of subclasses (k_i) of class c_i and subclass representatives $(p_{c_{i_1}}, p_{c_{i_2}}, ..., p_{c_{k_i}})$.

```
1:  Initialize $k_i = 1$ and the class with the mean and covariance of $M$.
2:  **for** $j = 0; j < J; j++$ **do**
3:      Randomly generate a $d \times 1$ projection vector $P$.
4:      Project both the model and $M$ to $P$ with the same projection vector.
5:      Use KS test to check the fitness of the model at significance level $\alpha$.
6:      **if** $H_0$ is rejected by any test **then**
7:          break out of the loop.
8:      **end if**
9:  **end for**
10: **if** $H_0$ is rejected by any test **then**
11:     Initialize the $k_i + 1$-th subclass representative (the $k_i$ previously learned plus
        one new subclass).
12:     Run *UK-means* on the $k_i+1$ subclasses to learn $k_i+1$ subclass representatives.
13:     $k_i = k_i + 1$ and go to Line 2.
14: **end if**
15: $H_0$ is accepted by each test; stop and return the model.
```

that if J random projections are performed, the probability with all J projections

m connecting the two centers ($m = \mu_2 - \mu_1$) satisfies the condition $||m|| \geq c\sqrt{trace(\Sigma)}$.

being 'bad'[2] is less than *error*:

$$Pr(J\ bad\ projections) = Erf(\sqrt{1/2})^J < error. \tag{5.2}$$

Where Erf is the standard Gaussian error function. Approximately J times projections are needed to keep the two subclass means c-separated, and the detail is shown as follows [32, 37] :

$$J < log(\epsilon)/log(Erf(\sqrt{1/2})) \approx -2.6198log(\epsilon). \tag{5.3}$$

For example, if ϵ is 0.01, 12 projections are needed. In the experiments, we use $J = 12$ projections to estimate the number of subclasses in a class.

5.4 Bounded Supervised UK-means with Multiple Subclasses (BSUMS)

In Algorithm 10, we can see that UK-means is used to cluster objects and calculate new subclass representatives when any test rejects the null hypothesis (H_0). During each time k_i increases by 1, the objects are reclustered by UK-means. The execution time of k_i estimation is related to the final value of k_i. At last, UK-means will be executed k_i times which is a bit time consuming. Moreover, expected distance calculation in supervised UK-means is T times more expensive than distance calculation in K-means. PG-means assumes each class is a mixture of Gaussian models. In the experiments, we found that sometimes the number of subclass (k_i)

[2] The probability of J is a 'bad' projection, for example, that when do projection, c-separation between cluster means (i.e. μ_1, μ_2) cannot be maintained.

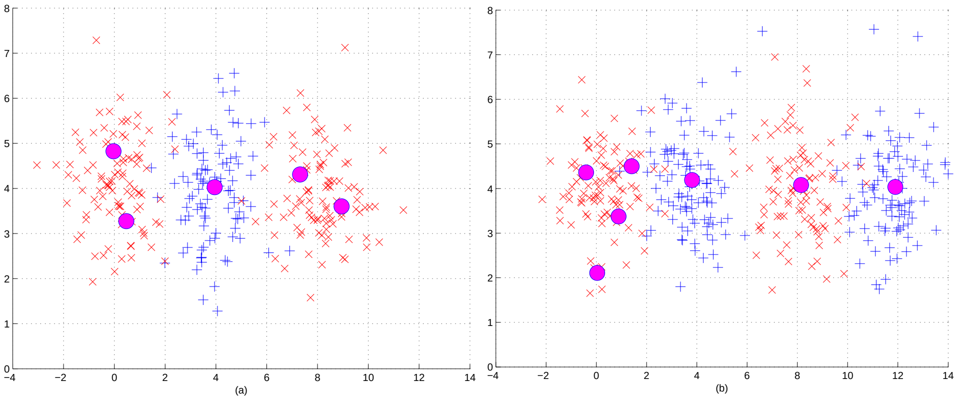

Figure 5.2 (a) Overestimated subclass representatives of "Middle" learned by SUMS (b) Overestimated subclass representatives of "Side" learned by SUMS.

is likely to be overestimated. In Figure 5.2(a), class ('x') is divided into four subclasses. In fact, it is not necessary to further divide the class ('x') into so many subclasses, and just two subclass representative is enough for class ('x'). In Figure 5.2(b), the left subclass is further divided into four subclasses which the number of subclasses of class ('x') is overestimated. To make supervised UK-means with multiple subclasses (SUMS) more efficient, we add an upper bound for terminating the estimation earlier in SUMS by not considering extra subclasses. The bound ($\delta \geq 2$) sets an upper bound for the number of subclasses ($k_i \leq \delta$). Algorithm 10 will end when all the tests accept the null hypothesis (H_0) or k_i is larger than δ. Different from Algorithm 10, in BSUMS Line 6 and 10 is **If any test rejected the null hypothesis** (H_0) **and** $k_i \leq \delta$.

5.5 Experimental Evaluation

In this section, we compare supervised UK-means with multiple subclasses (SUMS) and bounded supervised UK-means with multiple subclasses (BSUMS) with supervised UK-means in [105] and seeded K-means in [13]. All algorithms were written in Matlab and were run on a Windows machine with an Intel 2.66GHz Pentium(R) Dual-Core processor and 4GB of main memory.

5.5.1 Synthetic Data sets

We have done the experiments on the two typical data sets shown in Figure 5.1(a) and Figure 5.1(b). In Figure 5.1(a), the class ('x') is divided by the class ('+'). The number of objects in class ('+') is 100, and the number of objects in class ('x') is 200 with equally distribution on the two sides of class ('+'). The centers of uncertain objects were generated from a Gaussian distribution, whose mean and standard deviation are equal to the class center and σ respectively. The above set of uncertain objects represented by MBRs with size $S \times S$ were generated in $2D$ space $[-100, 100] \times [-100, 100]$. An MBR is divided into $\sqrt{T} \times \sqrt{T}$ grid cells. Each grid cell corresponds to a sample. Each sample is associated with a randomly generated probability value. All probabilities in an MBR are normalized to have their sum equal to 1. Similarly, in Figure 5.1(b), the number of objects in class ('x') is 200, and the number of objects in class ('+') is 200. The class ('x') and the class ('+') are divided into two subclasses by each other. Each subclass has 100 objects.

In Figures 5.3 and 5.4, symbol 'o' is the learned (sub)class representative of a class. In Figure 5.1(a), the centers of class 'x' is $(1, 4)$ and $(8, 4)$, and the center of

Table 5.1 Accuracy of SUMS on synthetic data sets.

Data Set	SUMS	Seeded K-means	Supervised UK-means
Middle	0.9667 (k_{max} = 3)	0.5	0.5333
Side	0.9525 (k_{max} = 5)	0.5	0.5250

class ('+') is $(4,4)$. As Figure 5.3(a) shows, in supervised UK-means, the learned mean vector of class ('x') is $(1.4586,4.1002)$ while the learned mean vector of class ('+') is $(7.3606,4.0012)$. Figure 5.3(b) shows, in BSUMS, the learned class representatives of class ('+') is $(4.2605,4.3418)$ while the learned (sub)class representatives of class ('x') is $(0.4853,4.5173)$ and $(8.4765,4.2470)$. In Figure 5.1(b), the subclass representatives of class ('x') is $(0,4)$ and $(8,4)$ and the centers of class ('+') is $(4,4)$ and $(12,4)$. The learned subclass representatives of class ('x') are $(0.3906,4.1120)$ and $(8.1468,4.0784)$ while the learned subclass representatives of class ('+') are $(3.8085,4.1901)$ and $(11.8984,4.0354)$ (Figure 5.4(b)) in BSUMS. In Figure 5.4(a), the supervised UK-means can only learn one class representative for each class (the class representative of class ('x') is $(2.1115,4.1398)$ and the class representative of class ('+') is $(10.0523,4.0693)$). In Figure 5.3(a) and 5.4(a), the learned class representative is far from some objects from the same class but separated by objects from other classes which makes the accuracy of classifier low. In Figure 5.3(b) and 5.4(b), the learned subclass representatives can improve the performance of supervised UK-means, because the subclass representatives are closer to the subset of objects they are representing. We use the same way to generate the testing data sets of "Middle" and "Side" with 10% size of training sets to measure the accuracy of the algorithms.

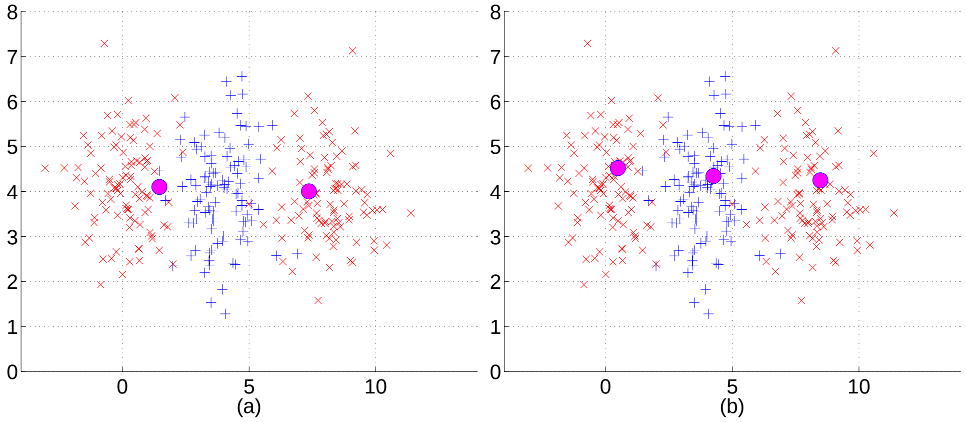

Figure 5.3 (a) Class representatives of “Middle” learned by supervised UK-means (b) Subclass representatives of “Middle” learned by BSUMS.

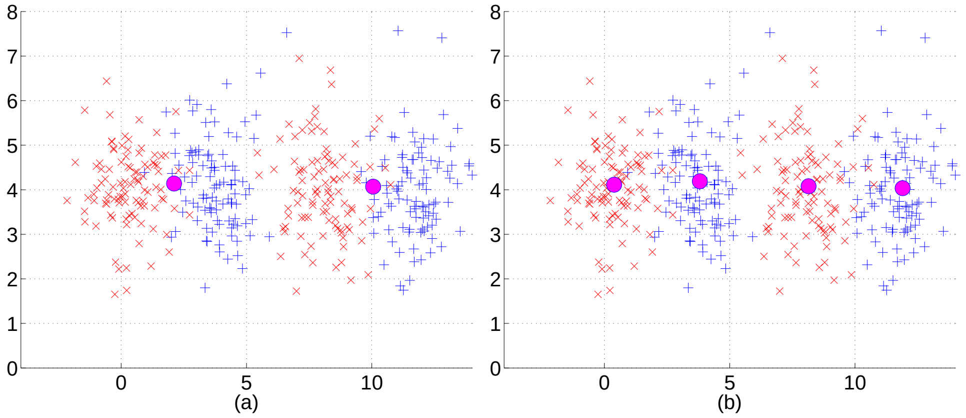

Figure 5.4 (a) Class representatives of “Side” learned by supervised UK-means (b) Subclass representatives of “Side” learned by BSUMS.

We use k_{max} to illustrate the maximum estimated number of subclasses among classes in SUMS. For example, if PG-means estimates $k_1 = 2$ and $k_2 = 3$ in SUMS, we denote k_{max} as 3. Table 5.1 and Table 5.2 show that there is not much difference between the results of SUMS and bounded SUMS (BSUMS). We can see that SUMS and BSUMS can overcome the limitation of supervised UK-means and seeded K-means. From the experiments, SUMS and BSUMS with PG-means can

Table 5.2 Accuracy of BSUMS on synthetic data sets.

Data Set	BSUMS (δ = 2)	BUMS (δ = 3)	BSUMS (δ = 4)	BSUMS (δ = 5)
Middle	0.9667	0.9667	0.9333	0.9333
Side	0.925	0.925	0.925	0.925

learn the subclass representatives of the synthetic data sets while seeded K-means and supervised UK-means just learn the mean of each class. SUMS and BSUMS improves the accuracy of supervised UK-means and seeded K-means. We also did experiments on other synthetic data sets which also show that SUMS and BSUMS can learn subclasses more accurately than supervised UK-means and seeded K-means when the classes are divided by other classes.

5.5.2 Scalability

We also analyze the performance of supervised UK-means with multiple subclasses (SUMS) and bounded SUMS (BSUMS) by changing some parameters in Table 5.3. In base case, each class includes 4 subclasses, and the subclasses are distributed as Figure 5.5. The objects ('+') belong to a class, and the objects denoted by ('x') are labeled by the other class.

Value of δ

In the experiments, δ is used to set the upper bound for the number of learned subclass representatives (k_i). Figure 5.6(a) shows the execution time of BSUMS by varying δ. In BSUMS, both the data and the model are projected to the same dimension by PG-means [37]. In these experiments, if the number of subclasses

Table 5.3 Parameters for experiments using data sets.

Parameter	Description	Baseline Value
K	number of classes	2
T	number of samples per object	49
S	maximum size of MBR, $S \times S$	0.25
D	number of dimensions	2
σ	standard deviation of Gaussian distribution	1
k_{pre}	pre-defined number of subclasses	4
n	number of objects per subclass	50

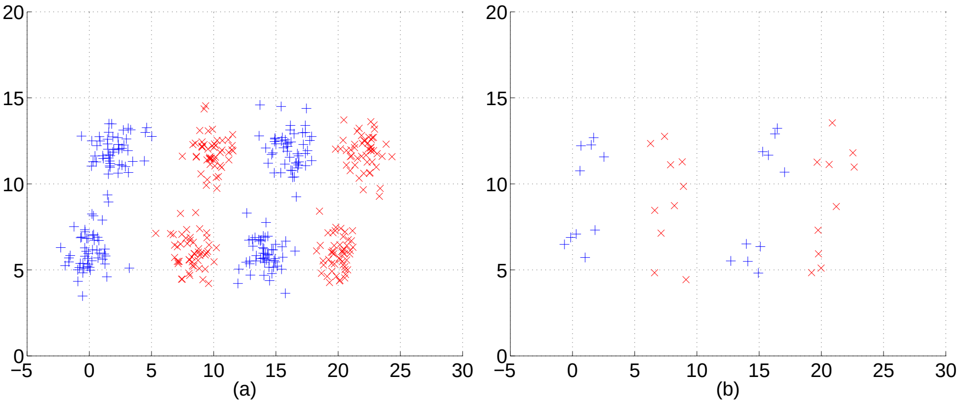

Figure 5.5 (a) Base case data distribution on training data set (b) Base case data distribution on testing data set.

is larger than δ, algorithm 10 will terminate even though the estimated model does not fit the data well. Figure 5.6(a) shows that the execution time increases when δ varies from 2 to 3. But the execution time does not change much when δ is larger than 3, because k_{max} in BSUMS ($\delta = 3, 4$) is 3 on baseline data set. From Figure 5.6 (b), $\delta = 2$ is enough for base case classification.

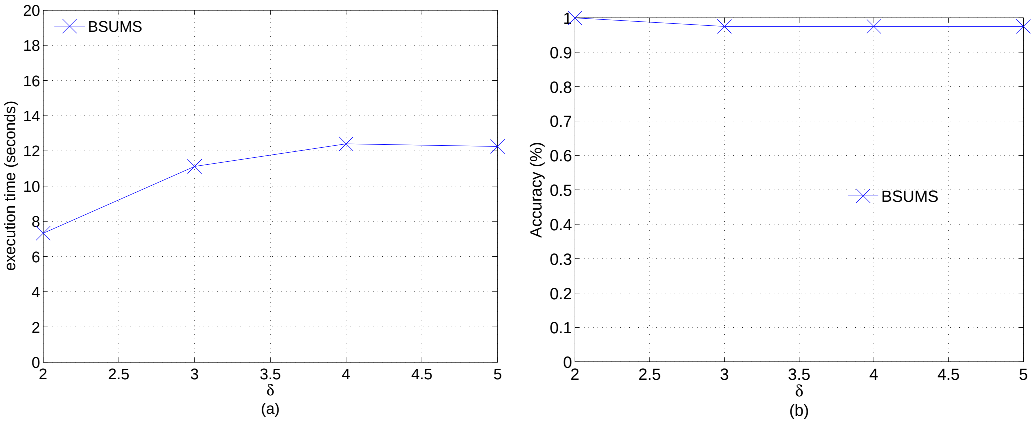

Figure 5.6 (a) Execution time of BSUMS with varying δ on synthetic data sets (b) Accuracy of BSUMS with varying δ on synthetic data sets

Pre-defined Number of Subclasses k_{pre}

In the experiments, we varied the pre-defined subclass number k_{pre} from 2 to 5. The other parameters were kept at baseline values. Figure 5.7(a) shows the execution time of all the algorithms increases as k_{pre} increasing. If k_{pre} is increasing, the total number of objects become larger, because the number of objects per subclass (n) is fixed at 50. SUMS, BSUMS and supervised UK-means cost more time than seeded K-means for the reason of expected distance calculation. Moreover, SUMS and BSUMS increase faster than supervised UK-means and seeded K-means, because SUMS and BSUMS cost more time on PG-means when k_{pre} increases. BSUMS ($\delta = 2$) is faster than SUMS and BSUMS ($\delta = 3, 4$). Figure 5.7(b) shows that the accuracy of SUMS and BSUMS outperforms that of supervised UK-means and seeded K-means when k_{pre} is larger than 2. When k_{pre} is 2, all the algorithms can perform well because the subclasses from the same class are not separated. In the cases of $k_{pre} = 3, 4, 5$, k_{max} is 3 in SUMS and BSUMS ($\delta = 3, 4$). Thus, there is no difference between the accuracy of SUMS and BSUMS ($\delta = 3, 4$). For $k_{pre} = 3, 4$,

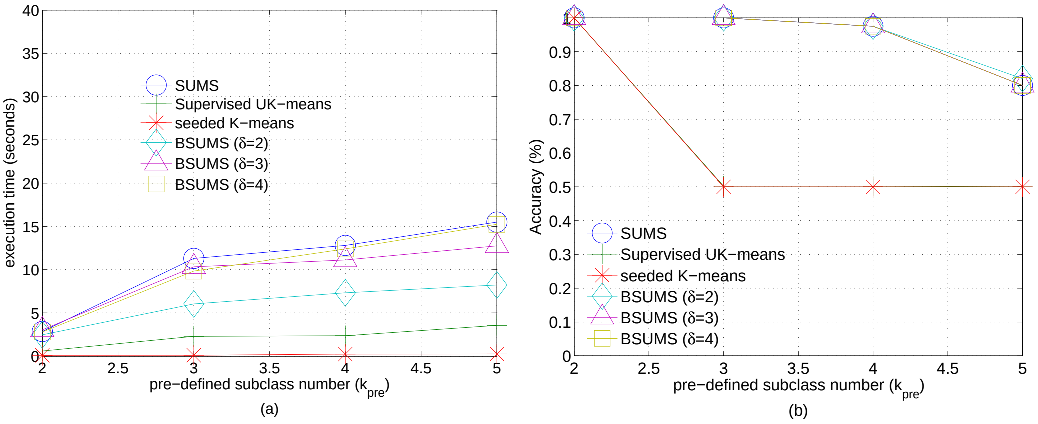

Figure 5.7 (a) Execution time with varying k_i on synthetic data sets (b) Accuracy with varying k_i on synthetic data sets

though k_{pre} is larger than 2, $\delta = 2$ can also learn the main distribution of the data sets in the experiments and attain the same level as $\delta = 3, 4$. The top lines that look so close together are 0.82 ($\delta = 2$, $k_{pre} = 5$), 0.8 ($\delta = 3, 4$, $k_{pre} = 5$ and SUMS), 1 ($\delta = 2, 3, 4$ and SUMS, $k_{pre} = 2$), 1 ($\delta = 2, 3, 4$ and SUMS, $k_{pre} = 3$), 0.975 ($\delta = 2, 3, 4$ and SUMS, $k_{pre} = 4$). To make it clearer, Table 5.4 shows the accuracy of the 6 lines in Figure 5.7(b).

Number of Objects *n*

In the experiments, we varied the object number of each subclass n from 50 to 250. The other parameters were kept at baseline values. Figure 5.8(a) shows the execution time of BSUMS, SUMS, seeded K-means and supervised UK-means. The execution time increases with n varying from 50 to 250, for the reason that the time cost on (expected) distance calculation increases linearly by increasing n. SUMS and BSUMS using PG-means which make themselves increase faster than supervised UK-means and seeded K-means. Seeded K-means is faster than other algorithms, because the distance calculation for certain object is faster than that for

Table 5.4 Accuracy with varying k_{pre} on synthetic data set.

Pre-defined Number of Subclasses (k_{pre})	2	3	4	5
Seeded K-means	1	0.5	0.5	0.5
Supervised UK-means	1	0.502	0.502	0.5
BSUMS (δ = 2)	1	1	0.975	0.82
BSUMS (δ = 3)	1	1	0.975	0.8
BSUMS (δ = 4)	1	1	0.975	0.8
SUMS	1	1	0.975	0.8

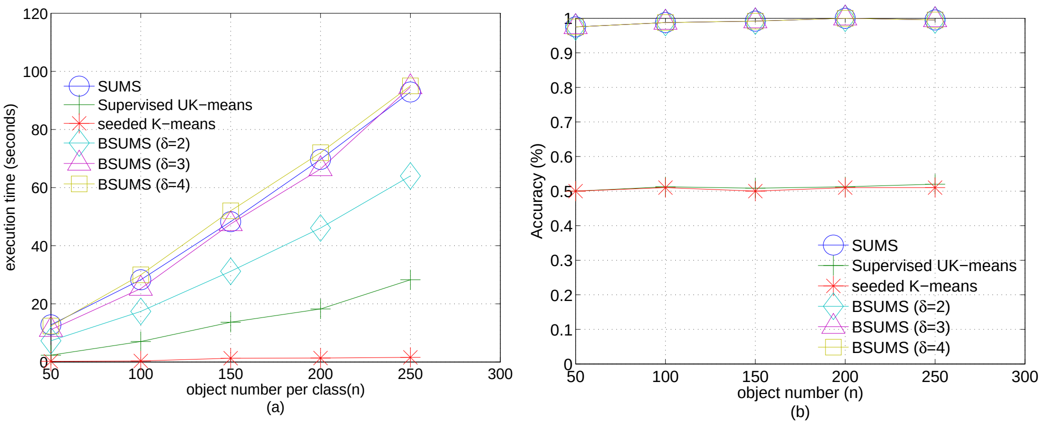

Figure 5.8 (a) Execution time with varying n on synthetic data sets (b) Accuracy with varying n on synthetic data sets

uncertain objects. BSUMS ($\delta = 2$) is faster than SUMS and BSUMS ($\delta = 3, 4$) for the reason that it can terminate the algorithm earlier. Similar to other cases, the accuracy of SUMS and BSUMS performs better than that of supervised UK-means and seeded K-means with varying n (Figure 5.8(b)).

Number of Dimensions *D*

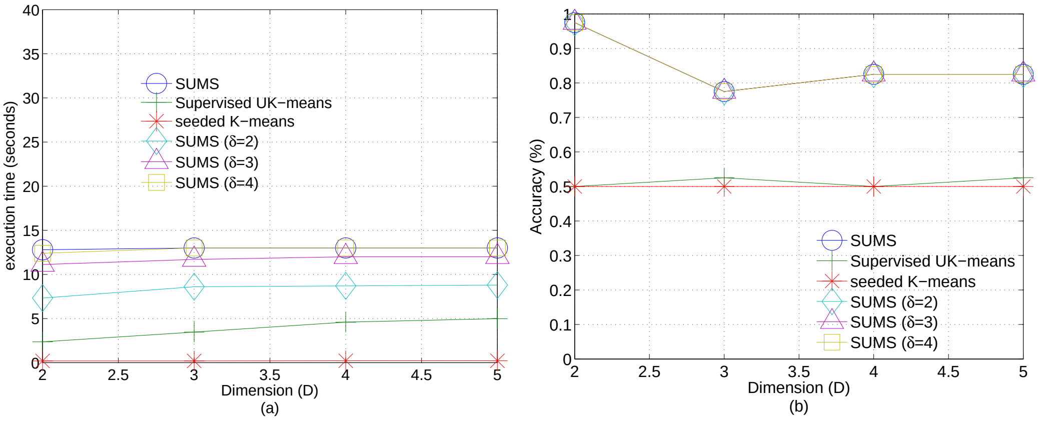

Figure 5.9 (a) Execution time with varying *D* on synthetic data sets (b) Accuracy with varying *D* on synthetic data sets

In the experiments, we varied the dimension number *D* from 2 to 5. The other parameters were kept at baseline values. Similar to the above cases, Figure 5.9(a) shows that the execution time of seeded K-means is faster than other algorithms because of the simplest distance calculation. BSUMS and SUMS cost more time than supervised UK-means for the reason of using PG-means. The execution time of SUMS and BSUMS does not increases fast when *D* become larger, because the time of PG-means is mainly affected by k_i (estimated subclass number). The accuracy of BSUMS and SUMS performs better than that of supervised UK-means and seeded K-means with varying *D* shown by Figure 5.9(b).

Number of Classes *K*

In the experiments, we varied the class number *K* from 2 to 5. Different from other cases, we add a class with 2 subclasses every time. The other parameters were kept at baseline values. Figure 5.10(a) shows that the execution time of all the algorithms. Different from other cases, SUMS and BSUMS is faster than supervised UK-means when *K* is larger than 4. In the experiments, SUMS and BSUMS

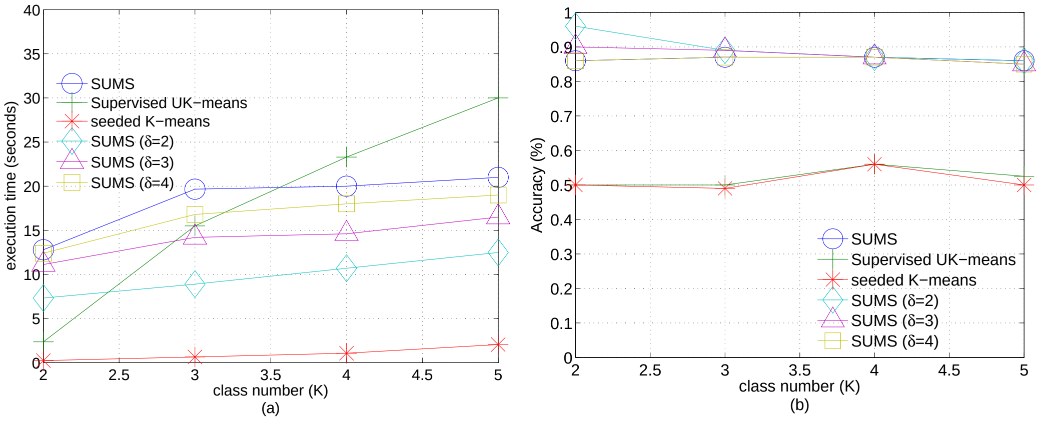

Figure 5.10 (a) Execution time with varying K on synthetic data sets (b) Accuracy with varying K on synthetic data sets

can find the (sub)class representatives more efficiently than supervised UK-means. Supervised UK-means spends more time to make the algorithm converge. Similar to other cases, the accuracy of BSUMS and SUMS performs better than that of supervised UK-means and seeded K-means with varying K which is shown in Figure 5.10(b).

From Figure 5.6(a) to Figure 5.10(a), it shows that the execution time of SUMS, BSUMS, seeded K-means and supervised UK-means increases when the number of classes (K) become larger (as well as the pre-defined number of subclasses (k_{pre}), the number of dimensions (D), the number of objects per subclass (n)). From Figure 5.6(b) to Figure 5.10(b), it is obvious that the accuracy of BSUMS and SUMS is better than that of supervised UK-means and seeded K-means. For the cases in these experiments, the accuracy is affected by the distribution of objects (i.e. overlapping of objects from different classes, subclasses distribution).

Table 5.5 Selected data sets from the UCI machine learning repository.

Data Set	Training Tuples	No. of Attributes	No. of Classes	Test Tuples
Iris	150	4	3	10-fold
BreastCancer	569	30	2	10-fold
Ionosphere	351	32	2	10-fold

5.5.3 Real Data Sets

To demonstrate that SUMS and BSUMS can improve the accuracy of supervised UK-means and seeded K-means [13], we also done experiments on real data sets. The parameters of the selected data sets used for the experiments are summarized in Table 5.5. The attributes of all the data sets are numerical obtained from measurements. Classifiers are built on the numerical attributes and their "class label" attributes. For the chosen data sets, we use 10-fold cross validation to measure the accuracy. The 3 data sets contains "point values" without uncertainty. In those existing research papers, they also use database without uncertainty and then add uncertainty into the data [5, 22, 58, 73, 82, 94, 95]. We follow the common practice in the research work of this area [5, 22, 58, 73, 82, 94, 95] to generate the uncertainty of synthetic data sets and real data sets. Each object is represented by an MBR with size 0.25×0.25 in a multiple dimension space, which is divided into $\sqrt{49} \times \sqrt{49}$ grid cells. Each grid cell corresponds to a sample. Each sample is associated with a randomly generated probability value. All probabilities in an MBR are normalized to have their sum equal to 1.

Table 5.6 Accuracy on real data sets.

Data Set	SUMS	BSUMS	Supervised UK-means	Seeded K-means
Iris	0.927	0.927	0.913	0.899
BreastCancer	0.895	0.895	0.843	0.842
Ionosphere	0.871	0.815	0.706	0.702

In Table 5.6, SUMS and BSUMS with PG-means can classify the objects more accurately compared with supervised UK-means and seeded K-means [13] on real data sets. Supervised UK-means and seeded K-means learn one class representative for each class. However, the objects of a class may be distributed closer to other class representatives. SUMS and BSUMS can train more than one (sub)class representatives for each class by PG-means. In SUMS and BSUMS, PG-means tries to estimate the number of subclasses of a class and learn local subclass representatives which may be closer to the objects belonging to the same class. BSUMS is a bounded SUMS, where PG-means terminates earlier if the estimated number of subclass (k_i) exceeds the bound δ. Thus, the accuracy will be affected. On Iris and BreastCancer, the classification quality is not affected by δ. In Ionosphere, SUMS performs better than BSUMS but the difference is very small (5.6%). The experiments show that supervised UK-means with multiple subclasses (SUMS) and bounded supervised UK-means with multiple subclasses (BSUMS) can improve the accuracy of supervised UK-means and seeded K-means on both synthetic and real data sets .

5.6 Summary

In this chapter, we propose supervised UK-means with multiple subclasses (SUMS) using PG-means (projected Gaussian means) for the purpose of handling objects from the same class disconnected by other classes. To make SUMS more efficient, we propose bounded SUMS (BSUMS) to avoid the number of subclasses being overestimated. Our experimental results demonstrate that both SUMS and BSUMS can overcome the limitation of supervised UK-means and seeded K-means when a class is divided by other classes.

Chapter 6

Conclusions and Future Work

Uncertainty is an inherent characteristic for collecting data. The problem of summarizing uncertain objects poses a number of challenges. In this thesis, we focus on value uncertainty. We summarize our contributions in Section 6.1. Then, we discuss some future work in Section 6.2.

6.1 Conclusions

In this thesis, we investigate the problem of clustering and classification on uncertain data. We model the uncertain data as uncertain objects whose locations are uncertain and described by probability density functions (pdf). Our contributions in this thesis include:

- We develop an effective and efficient clustering framework to discover common patterns among uncertain objects. In previous work, UK-means is reduced to K-means by using expected squared Euclidean distance instead of expected Euclidean distance to overcome the bottleneck of existing techniques. Due to different distance functions used in clustering, we propose Approximate UK-means to reduce the discrepancy by heuristically identifying objects of boundary cases and re-assigning them. In addition, we consider the uncertainty of cluster representative for clustering uncertain objects.

- We develop a classification framework for uncertain objects. Existing algorithms are too complex and time consuming. We use supervised UK-means to classify uncertain objects efficiently with the trade off of some loss in accuracy. To enhance supervised UK-means, we extend supervised UK-means to feature selection and Adaboost respectively.

- In real applications, objects from the same class may be disconnected by other classes. Thus, we propose supervised UK-means with multiple subclasses (SUMS) to tackle the problem. SUMS uses PG-means (projected Gaussian means) to estimate the number of subclasses and then assign objects to their closest subclass representatives. To make SUMS more efficient, a bound is set to avoid subclasses being overestimated which is noted as bounded SUMS (BSUMS).

6.2 Future Work

It is meaningful to extend our work to more sophisticated models for uncertain data. In this thesis, we consider the value uncertainty that exists inherently to uncertain data.

- **Distance Metrics**

  In this thesis, we focus on value uncertainty. Each object is presented by a probability density distribution (pdf). Most work measures the distance between uncertain objects without considering uncertainty of pdf. For example, the pdf of an object is uniform distribution, and the pdf of the other object is gaussian distribution. Thus, the distance between the two objects is more precise if the pdfs of the two objects are considered. However, sometimes the pdfs of objects may be different. In future work, we should also consider difference between the pdfs when we measure the distance between uncertain objects.

- **Categorical Attributes**

  In real applications, some data sets contain categorical attributes. However, few techniques can handle categorical attributes on uncertain data. In future work, we will consider clustering and classification on uncertain data described by numerical attributes as well as uncertain data with categorical attributes.

- **Objects Overlapping**

  Supervised UK-means performs better when the data sets are well separated. It is more effective to apply our work to data sets which are more well separated with less overlapping. In future work, we will try to find an efficient way to define objects overlapping to use supervised UK-means more appropriately.

Bibliography

[1] Serge Abiteboul, Paris C. Kanellakis, and Gösta Grahne. On the representation and querying of sets of possible worlds. In *Proceedings of the Association for Computing Machinery Special Interest Group on Management of Data 1987 Annual Conference*, pages 34–48, 1987.

[2] Charu C. Aggarwal. *Managing and Mining Uncertain Data*, volume 35 of *Advances in Database Systems*. Kluwer, 2009.

[3] Charu C. Aggarwal, Yan Li, Jianyong Wang, and Jing Wang. Frequent pattern mining with uncertain data. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 29–38, 2009.

[4] Charu C. Aggarwal and Philip S. Yu. Outlier detection with uncertain data. In *Proceedings of the SIAM International Conference on Data Mining*, pages 483–493, 2008.

[5] Charu C. Aggarwal and Philip S. Yu. A survey of uncertain data algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, 21(5):609–623, 2009.

[6] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar

Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 94–105, 1998.

[7] Rakesh Agrawal and Edward L. Wimmers. A framework for expressing and combining preferences. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 297–306, 2000.

[8] Sami H. Al-Harbi and Victor J. Rayward-Smith. Adapting -means for supervised clustering. *Applied Intelligence*, 24(3):219–226, 2006.

[9] Ramiz M. Aliguliyev. Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization. *Computational Intelligence*, 26(4):420–448, 2010.

[10] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 49–60, 1999.

[11] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[12] D. Barbará, H. Garcia-Molina, and D. Porter. The management of probabilistic data. *IEEE Transactions on Knowledge and Data Engineering*, 4(5):487–502, 1992.

[13] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. Semi-supervised clustering by seeding. In *Proceedings of the 19th International Conference on Machine Learning*, pages 27–34, 2002.

[14] Omar Benjelloun, Anish Das Sarma, Alon Y. Halevy, and Jennifer Widom. Uldbs: Databases with uncertainty and lineage. In *Proceedings of the 32rd International Conference on Very Large Data Bases*, pages 953–964, 2006.

[15] Thomas Bernecker, Hans-Peter Kriegel, Matthias Renz, Florian Verhein, and Andreas Züfle. Probabilistic frequent itemset mining in uncertain databases. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 119–128, 2009.

[16] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[17] Jinbo Bi and Tong Zhang. Support vector classification with input data uncertainty. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems]*, 2004.

[18] Wei Bian and Dacheng Tao. Learning a distance metric by empirical loss minimization. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 1186–1191, 2011.

[19] Stephan Börzsönyi, Donald Kossmann, and Konrad Stocker. The skyline operator. In *Proceedings of the 17th International Conference on Data Engineering*, pages 421–430, 2001.

[20] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[21] Chee Yong Chan, H. V. Jagadish, Kian-Lee Tan, Anthony K. H. Tung, and Zhenjie Zhang. Finding k-dominant skylines in high dimensional space. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 503–514, 2006.

[22] Michael Chau, Reynold Cheng, Ben Kao, and Jackey Ng. Uncertain data mining: An example in clustering location data. In *The 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 199–204, 2006.

[23] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1112–1127, 2004.

[24] Reynold Cheng, Dmitri V. Kalashnikov, and Sunil Prabhakar. Evaluating probabilistic queries over imprecise data. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 551–562, 2003.

[25] Jan Chomicki. Querying with intrinsic preferences. In *Advances in Database Technology - EDBT 2002, 8th International Conference on Extending Database Technology, Proceedings*, pages 34–51, 2002.

[26] Jan Chomicki. Database querying under changing preferences. *Annals of Mathematics and Artificial Intelligence*, 50(1-2):79–109, 2007.

[27] Jan Chomicki, Parke Godfrey, Jarek Gryz, and Dongming Liang. Skyline with presorting. In *Proceedings of the 19th International Conference on Data Engineering*, pages 717–719, 2003.

[28] Chun Kit Chui and Ben Kao. A decremental approach for mining frequent itemsets from uncertain data. In *Advances in Knowledge Discovery and Data Mining - 12th Pacific-Asia Conference*, pages 64–75, 2008.

[29] Chun Kit Chui, Ben Kao, and Edward Hung. Mining frequent itemsets from uncertain data. In *Advances in Knowledge Discovery and Data Mining - 11th Pacific-Asia Conference*, pages 47–58, 2007.

[30] Nilesh N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB Journal*, 16(4):523–544, 2007.

[31] Nilesh N. Dalvi and Dan Suciu. Management of probabilistic data: foun-

dations and challenges. In *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 1–12, 2007.

[32] Sanjoy Dasgupta. Experiments with random projection. In *Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence*, pages 143–151, 2000.

[33] Frank K. H. A. Dehne and H. Noltemeier. Voronoi trees and clustering problems. *Information Systems*, 12(2):171–175, 1987.

[34] Evangelos Dellis and Bernhard Seeger. Efficient computation of reverse skyline queries. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, pages 291–302, 2007.

[35] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

[36] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[37] Yu Feng and Greg Hamerly. Pg-means: learning the number of clusters in data. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*, pages 393–400, 2006.

[38] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting, 1995.

[39] Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby. Margin based feature

selection - theory and algorithms. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[40] Parke Godfrey, Ryan Shipley, and Jarek Gryz. Maximal vector computation in large data sets. In *Proceedings of the 31st International Conference on Very Large Data Bases*, pages 229–240, 2005.

[41] Kannan Govindarajan, Bharat Jayaraman, and Surya Mantha. Preference queries in deductive databases. *New Generation Computing*, 19(1):57–86, 2000.

[42] Hans Peter Graf, Eric Cosatto, Léon Bottou, Igor Durdanovic, and Vladimir Vapnik. Parallel support vector machines: The cascade svm. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems]*, 2004.

[43] Gösta Grahne. *The Problem of Incomplete Information in Relational Databases*, volume 554 of *Lecture Notes in Computer Science*. Springer, 1991.

[44] Todd J. Green and Val Tannen. Models for incomplete and probabilistic information. *IEEE Data Engineering Bulletin*, 29(1):17–24, 2006.

[45] Greg Hamerly and Charles Elkan. Learning the k in k-means. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems]*, 2003.

[46] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 1–12, 2000.

[47] J. Hartigan. *Clustering Algorithms*. John Wiley and Sons, New York, 1975.

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

[49] Ming Hua, Jian Pei, Wenjie Zhang, and Xuemin Lin. Ranking queries on uncertain data: a probabilistic threshold approach. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 673–686, 2008.

[50] E. Hung, L. Getoor, and V.S. Subrahmanian. Probsem: A probabilistic semistructured database. *Technical Report*, 2002.

[51] Edward Hung, Lise Getoor, and V. S. Subrahmanian. Pxml: A probabilistic semistructured data model and algebra. In *Proceedings of the 19th International Conference on Data Engineering*, pages 467–478, 2003.

[52] Edward Hung, Lei Xu, and Chi-Cheong Szeto. A heuristic on effective and efficient clustering on uncertain objects. In *AI 2010: Advances in Artificial Intelligence - 23rd Australasian Joint Conference, Proceedings*, pages 92–101, 2010.

[53] Tomasz Imielinski and Witold Lipski Jr. Incomplete information in relational databases. *Jounal of The ACM*, 31(4):761–791, 1984.

[54] Ravi Jampani, Fei Xu, Mingxi Wu, Luis Leopoldo Perez, Christopher M. Jermaine, and Peter J. Haas. Mcdb: a monte carlo approach to managing uncertain data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 687–700, 2008.

[55] Bin Jiang and Jian Pei. Outlier detection on uncertain data: Objects, instances, and inferences. In *Proceedings of the 27th International Conference on Data Engineering*, pages 422–433, 2011.

[56] Bin Jiang, Jian Pei, Xuemin Lin, David W. Cheung, and Jiawei Han. Mining preferences from superior and inferior examples. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 390–398, 2008.

[57] Bin Jiang, Jian Pei, Xuemin Lin, and Yidong Yuan. Probabilistic skylines on uncertain data: model and bounding-pruning-refining methods. *Journal of Intelligent Information Systems*, 38(1):1–39, 2012.

[58] Ben Kao, Sau Dan Lee, David W. Cheung, Wai-Shing Ho, and K. F. Chan. Clustering uncertain data using voronoi diagrams. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 333–342, 2008.

[59] Werner Kießling. Foundations of preferences in database systems. In *Proceedings of 28th International Conference on Very Large Data Bases*, pages 311–322, 2002.

[60] Werner Kießling and Gerhard Köstler. Preference sql - design, implementation, experiences. In *Proceedings of 28th International Conference on Very Large Data Bases*, pages 990–1001, 2002.

[61] Donald Kossmann, Frank Ramsak, and Steffen Rost. Shooting stars in the sky: An online algorithm for skyline queries. In *Proceedings of 28th International Conference on Very Large Data Bases*, pages 275–286, 2002.

[62] H.-P. Kriegel and M. Pfeifle. Density-based clustering of uncertain data. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 672–677, 2005.

[63] H.-P. Kriegel and M. Pfeifle. Hierarchical density-based clustering of uncertain data. In *Proceedings of the 5th IEEE International Conference on Data*

Mining, pages 689–692, 2005.

[64] M. Lacroix and Pierre Lavency. Preferences; putting more knowledge into queries. In *Proceedings of 13th International Conference on Very Large Data Bases*, pages 217–225, 1987.

[65] Richard D. Lawrence, George S. Almasi, Vladimir Kotlyar, Marisa S. Viveros, and Sastry Duri. Personalization of supermarket product recommendations. *Data Mining and Knowledge Discovery*, 5(1/2):11–32, 2001.

[66] S. D. Lee, B. Kao, and R. Cheng. Reducing uk-means to k-means. In *Workshops Proceedings of the 7th IEEE International Conference on Data Mining*, pages 483–488, 2007.

[67] Carson Kai-Sang Leung and Dale A. Brajczuk. Mining uncertain data for constrained frequent sets. In *International Database Engineering and Applications Symposium*, pages 109–120, 2009.

[68] David Littau and Daniel Boley. Clustering very large data sets using a low memory matrix factored representation. *Computational Intelligence*, 25(2):114–135, 2009.

[69] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[70] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems 10, [Neural Information Processing Systems]*, 1997.

[71] Takazumi Matsumoto and Edward Hung. Accelerating outlier detection with

uncertain data using graphics processors. In *Advances in Knowledge Discovery and Data Mining - 16th Pacific-Asia Conference*, pages 169–180, 2012.

[72] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley, New York, 1997.

[73] Wang Kay Ngai, Ben Kao, Chun Kit Chui, Reynold Cheng, Michael Chau, and Kevin Y. Yip. Efficient clustering of uncertain data. In *Proceedings of the 6th IEEE International Conference on Data Mining*, pages 436–445, 2006.

[74] Andrew Nierman and H. V. Jagadish. Protdb: Probabilistic data in xml. In *Proceedings of 28th International Conference on Very Large Data Bases*, pages 646–657, 2002.

[75] Jian Pei, Jiawei Han, Hongjun Lu, Shojiro Nishio, Shiwei Tang, and Dongqing Yang. H-mine: Hyper-structure mining of frequent patterns in large databases. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 441–448, 2001.

[76] Jian Pei, Ming Hua, Yufei Tao, and Xuemin Lin. Query answering techniques on uncertain and probabilistic data: tutorial summary. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1357–1364, 2008.

[77] Jian Pei, Wen Jin, Martin Ester, and Yufei Tao. Catching the best views of skyline: A semantic approach based on decisive subspaces. In *Proceedings of the 31st International Conference on Very Large Data Bases*, pages 253–264, 2005.

[78] Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the 17th Inter-*

national Conference on Machine Learning, pages 727–734, 2000.

[79] Claudia Perlich and Foster J. Provost. Distribution-based aggregation for relational learning with identifier attributes. *Machine Learning*, 62(1-2):65–105, 2006.

[80] Biao Qin, Yuni Xia, Sunil Prabhakar, and Yi-Cheng Tu. A rule-based classification algorithm for uncertain data. In *Proceedings of the 25th International Conference on Data Engineering*, pages 1633–1640, 2009.

[81] J. Ross Quinlan. Learning decision tree classifiers. *ACM Computing Survey*, 28(1):71–72, 1996.

[82] Jiangtao Ren, Sau Dan Lee, Xianlu Chen, Ben Kao, Reynold Cheng, and David Wai-Lok Cheung. Naive bayes classification of uncertain data. In *The 9th IEEE International Conference on Data Mining*, pages 944–949, 2009.

[83] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–139, 2010.

[84] Enrique H. Ruspini. A new approach to clustering. *Information and Control*, 15(1):22–32, 1969.

[85] Fereidoon Sadri. Modeling uncertainty in databases. In *Proceedings of the 7th International Conference on Data Engineering*, pages 122–131, 1991.

[86] Anish Das Sarma, Omar Benjelloun, Alon Y. Halevy, and Jennifer Widom. Working models for uncertain data. In *Proceedings of the 22nd International Conference on Data Engineering*, page 7, 2006.

[87] Mika Sato, Yoshiharu Sato, and L. C. Jain. *Fuzzy Clustering Models and Applications*. Physica-Verlag, 1997.

[88] R. Sibson. Slink: An optimally efficient algorithm for the single-link cluster

method. *Computer Journal*, 16(1):30–34, 1973.

[89] M. Tabakov. A fuzzy segmentation method for computed tomography images. *International Journal of Intelligent Information and Database Systems*, 1(1):79–89, 2007.

[90] Kian-Lee Tan, Pin-Kwang Eng, and Beng Chin Ooi. Efficient progressive skyline computation. In *Proceedings of 27th International Conference on Very Large Data Bases*, pages 301–310, 2001.

[91] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[92] Yufei Tao, Reynold Cheng, Xiaokui Xiao, Wang Kay Ngai, Ben Kao, and Sunil Prabhakar. Indexing multi-dimensional uncertain data with arbitrary probability density functions. In *Proceedings of the 31st International Conference on Very Large Data Bases*, pages 922–933, 2005.

[93] Yufei Tao, Xiaokui Xiao, and Jian Pei. Subsky: Efficient computation of skylines in subspaces. In *Proceedings of the 22nd International Conference on Data Engineering*, page 65, 2006.

[94] S. Tsang, B. Kao, K. Y. Yip, W.-S. Ho, and S. D. Lee. Decision trees for uncertain data. In *Proceedings of the 25th International Conference on Data Engineering*, pages 441–444, 2009.

[95] Smith Tsang, Ben Kao, Kevin Y. Yip, Wai-Shing Ho, and Sau Dan Lee. Decision trees for uncertain data. *IEEE Transactions on Knowledge and Data Engineering*, 23(1):64–78, 2011.

[96] Maurice van Keulen, Ander de Keijzer, and Wouter Alink. A probabilistic xml approach to data integration. In *Proceedings of the 21st International*

Conference on Data Engineering, pages 459–470, 2005.

[97] Bin Wang, Gang Xiao, Hao Yu, and Xiaochun Yang. Distance-based outlier detection on uncertain data. In *12th IEEE International Conference on Computer and Information Technology*, pages 293–298, 2009.

[98] Dianhui Wang, Yong-Soo Kim, Seok Cheon Park, Chul Soo Lee, and Yoon Kyung Han. Learning based neural similarity metrics for multimedia data mining. *Soft Computing*, 11(4):335–340, 2007.

[99] Dianhui Wang and Xiaohang Ma. Learning pseudo metric for multimedia data classification and retrieval. In *Knowledge-Based Intelligent Information and Engineering Systems, 8th International Conference, Proceedings. Part I*, pages 1051–1057, 2004.

[100] Jun Wang, Huyen Do, Adam Woznica, and Alexandros Kalousis. Metric learning with multiple kernels. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems*, pages 1170–1178, 2011.

[101] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[102] Max Welling and Kenichi Kurihara. Bayesian k-means as a ”maximization-expectation” algorithm. In *Proceedings of the 6th SIAM International Conference on Data Mining*, 2006.

[103] David Wolpert and William G. Macready. An efficient method to estimate bagging’s generalization error. *Machine Learning*, 35(1):41–55, 1999.

[104] Lurong Xiao and Edward Hung. An efficient distance calculation method for

uncertain objects. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, pages 10–17, 2007.

[105] Lei Xu and Edward Hung. Distance-based feature selection on classification of uncertain objects. In *AI 2011: Advances in Artificial Intelligence - 24th Australasian Joint Conference, Proceedings*, pages 172–181, 2011.

[106] Lei Xu and Edward Hung. Improving classification accuracy on uncertain data by considering multiple subclasses. In *AI 2012: Advances in Artificial Intelligence - 25th Australasian Joint Conference, Proceedings*, pages 743–754, 2012.

[107] Tao Yang, Vojislav Kecman, Longbing Cao, Chengqi Zhang, and Joshua Zhexue Huang. Margin-based ensemble classifier for protein fold recognition. *Expert Systems with Applications*, 38(10):12348–12355, 2011.

[108] Qin Zhang, Feifei Li, and Ke Yi. Finding frequent items in probabilistic data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 819–832, 2008.

[109] Zhenjie Zhang, Yin Yang, Ruichu Cai, Dimitris Papadias, and Anthony K. H. Tung. Kernel-based skyline cardinality estimation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 509–522, 2009.

[110] Wenzhong Zhao, Alex Dekhtyar, and Judy Goldsmith. A framework for management of semistructured probabilistic data. *Journal of Intelligent Information Systems*, 25:1–39, 2004.