Université de Tunis

Institut Supérieur de Gestion de Tunis

Thèse présentée pour l'obtention du titre de

Docteur en Informatique de gestion

Dynamic Clustering within the Belief Function Framework

Réalisée par: Sarra Ben Hariz

Supervisée par: Zied Elouedi, Maître de conférences (ISG, Tunis)

LAboratoire de Recherche Opérationnelle, de DEcision et de Contrôle de Processus (LARODEC)

Novembre 2012

TO THE MEMORY OF MY FATHER

Acknowledgments

Many people have supported me during the achievement of my PhD.
First of all, I am grateful to my advisor Dr. Zied Elouedi for his guidance, endless patience and continuous encouragements throughout the course of this work. Special thanks for you, for your productive comments, fruitful discussions and advices.
Your patience and support helped me overcome many crisis situations and finish this dissertation. I hope that one day I would become as good an advisor to my students as Dr. Elouedi has been to me.
My sincere gratitude to Pr. Khaled Mellouli who introduced me, with my advisor, to the area of research, especially to classification and uncertainty domains.

Thanks to all members of my laboratory LARODEC, at High Institute of Management of Tunis, and to my friends. Their constant support and care helped me overcome setbacks and stay focused on my graduate study. I greatly value their friendship and I deeply appreciate their belief in me.

Most importantly, None of this would have been possible without the love and patience of my family. It has been a constant source of love, concern, support and strength all these years. I would like to express my heart-felt gratitude to my parents, sisters and brother.

I have to give a special mention for the support given by my husband. I warmly appreciate his generosity and understanding. Finally, I dedicate this work to my lovely son.

Abstract

If uncertainty modeling is already important in static clustering, this becomes even more important in dynamic clustering where several classification elements can change over time. Thus, this thesis proposes different dynamic clustering approaches under uncertainty. These developed techniques are able to update cluster rearrangement when any change is detected. This latter may concern the clusters, the attributes or the objects. The handled uncertainty appears in attribute values and is represented and managed through the Transferable Belief Model (TBM), one interpretation of the belief function theory. The first family of techniques aims at handling the dynamic cluster partition of static data sets. The second is adapted to deal with dynamic data set features, whereas the third one is interested in dynamic data set items. All these methods are proposed under uncertainty. Indeed, our objective is to provide methods able to classify objects even those characterized by uncertain attribute values within such dynamic environments. Simulations are performed in order to show the feasibility and the interest of our dynamic belief clustering approaches compared to static ones.

Keywords:
Clustering, Dynamic environment, Uncertainty, Belief function theory

Résumé

Si la modélisation de l'incertitude est déjà importante dans le cadre de la classification automatique, elle devient encore plus importante dans le cadre de la classification non supervisée dynamique où plusieurs éléments de classification peuvent changer au cours du temps.
Ainsi, cette thèse propose différentes approaches dynamiques de classification non supervisée dans un cadre incertain. Ces techniques permettent la mise à jour de la partition des clusters face à n'importe quel changement. Ce dernier est relatif aux groupes, aux attributs ou aux objets. La forme de l'incertitude traitée est relative aux valeurs des attributs et elle est représentée et gérée dans le cadre du modèle des croyances transférables (MCT), une des interprétations de la théorie des fonctions de croyance. La première famille de techniques vise à traiter les partitions dynamiques des ensembles de données statiques. La deuxième est proposée pour gérer les ensembles d'attributs dynamiques, tandis que la troisième s'intéresse aux ensembles d'instances dynamiques. Toutes ces méthodes sont proposées dans un cadre incertain. En effet, notre objectif est de fournir un ensemble de méthodes capables de classer des objets même ceux caractérisés par des valeurs d'attributs incertaines et ce dans de tels environnements dynamiques. Des simulations ont été effectuées afin de montrer la faisabilité et l'intérêt de nos approches dynamiques de classification automatique crédibiliste par rapport à celles statiques.

Mots clés:
Classification non supervisée, Environnement dynamique, Incertitude, Théorie des fonctions de croyance.

Contents

Abbreviations and Notations **x**

List of Figures **xi**

List of Tables **xiii**

Introduction **1**

1 Belief function theory **5**

1.1 Introduction 5

1.2 Basic concepts 6

1.2.1 Frame of discernment 6

1.2.2 Basic belief assignment 7

1.2.3 Focal elements, body of evidence, core 8

1.2.4 Belief function 9

1.2.5 Implicability function 10

1.2.6 Plausibility function 10

1.2.7 Commonality function 12

1.2.8 Belief intervals 13

1.3 Special belief functions 13

1.3.1 Vacuous belief function 13

1.3.2 Categorical belief function 13

1.3.3 Bayesian belief function 14

1.3.4 Consonant belief function 14

1.3.5 Certain belief function 15

1.3.6 Simple support function 15

1.3.7 Dogmatic and non-dogmatic belief functions 15

1.4 The combination rules 16

1.4.1 The conjunctive case 17
1.4.2 The disjunctive case 18
1.5 Discounting belief functions 19
1.6 Decision making 20
1.6.1 Pignistic probability 20
1.6.2 Maximum of credibility 21
1.6.3 Maximum of plausibility 21
1.7 Conclusion 22

2 Data clustering : An overview 23
2.1 Introduction 23
2.2 From data types to data scale 24
2.3 Proximity measures 24
2.3.1 Similarity/Dissimilarity matrix definition 25
2.3.2 Proximity measures versus data types 25
2.3.3 Proximity measures for cluster pairs 26
2.4 Taxonomy of classical clustering techniques 27
2.4.1 Partitional versus Hierarchical 28
2.4.2 Exclusive versus Overlapping 29
2.5 Origins of the standard K-modes method 30
2.5.1 Basic parameters 30
2.5.2 SKM algorithm 32
2.6 Clustering under uncertainty 33
2.7 Online clustering 35
2.8 Clustering: Additional issues 36
2.9 Conclusion 38

3 Improving Belief Clustering via a new initialization method 39
3.1 Introduction 39
3.2 Belief K-modes method 40
3.2.1 Belief data set structure 40
3.2.2 BKM parameters 41
3.2.3 BKM algorithm 43
3.3 Clustering initialization 44
3.3.1 Problem definition 44
3.3.2 Existing initialization approaches 44
3.4 Max-dissimilarity based initialization method under belief function framework 46
3.4.1 Principle 46

3.4.2 Algorithm 47
3.4.3 Illustrative example 48
3.4.4 Experimental analysis 50
3.5 Conclusion 58

4 Handling dynamic partition based on belief clustering 59
4.1 Introduction 59
4.2 Problem definition 60
4.3 Dynamic clustering preliminaries under uncertainty 61
4.3.1 General principle 61
4.3.2 Basic definitions 62
4.4 Optimization based dynamic clustering under uncertainty 64
4.4.1 Dynamic clustering using cohesion minimization under uncertainty 64
4.4.2 Separation maximization based dynamic belief clustering . . . 67
4.4.3 Concluding remarks 70
4.5 Experimentations 71
4.5.1 Experimental strategies 72
4.5.2 Empirical results and discussion 73
4.5.3 Experimental summary 85
4.6 Conclusion 86

5 Belief clustering methods for dynamic data set features 87
5.1 Introduction 87
5.2 Problem presentation 88
5.3 Necessary concepts 90
5.3.1 General assumptions 90
5.3.2 Basic notions 90
5.4 New clustering proposals dealing with dynamic feature set under uncertainty 91
5.4.1 Feature selection method within belief function framework . . 91
5.4.2 Dynamic belief clustering for handling incremental feature set 96
5.4.3 Decremental clustering to reduce feature set under uncertainty 99
5.5 Experimentations 102
5.5.1 Experimental strategies 102
5.5.2 Empirical results and discussion 103
5.5.3 Experimental summary 115
5.6 Conclusion 116

6 Clustering dynamic data items under uncertainty **117**
6.1 Introduction 117
6.2 Problem statement 118
6.3 Basis of dynamic data clustering under uncertainty 119
6.3.1 Global view 119
6.3.2 Basic terminology 120
6.4 Dynamic data processing using belief clustering 121
6.4.1 The detection of significant data step 121
6.4.2 Clustering online data within uncertainty 123
6.4.3 Clustering reduced data in the belief function context 125
6.5 Experimentations 127
6.5.1 Experimental strategies 127
6.5.2 Empirical results and discussion 128
6.5.3 Experimental summary 139
6.6 Conclusion 140

Conclusion **141**

A Experimental settings **144**
A.1 Introduction 144
A.2 UCI data sets description 144
A.3 Data sets in a belief function framework 145
A.3.1 Artificial bba's creation 145
A.3.2 Lenses database as example: From certain to belief version . . 147
A.4 Evaluation criteria 150
A.5 Conclusion 152

Bibliography **152**

Abbreviations and Notations

Abbreviations	
SKM	: Standard K-Modes method.
BKM	: Belief K-Modes method.
$BCDP$	: Belief Clustering for Dynamic Partition.
$BCDDD$	: Belief Clustering for Dynamic Data set Dimensions.
$BCDDF$	: Belief Clustering for Dynamic Data set Features.
$BCDDI$	: Belief Clustering for Dynamic Data set Items.
$IKBKM$	: Incremental K Belief K-modes approach.
$DKBKM$	: Decremental K Belief K-modes approach.
$ISBKM$	: Incremental S Belief K-modes approach.
$DSBKM$	: Decremental S Belief K-modes approach.
$INBKM$	: Incremental N Belief K-modes approach.
$DNBKM$	: Decremental N Belief K-modes approach.
BFS	: Belief Feature Selection method.
$LFSC$	: Local Feature Score Computation approach.
$FRvsFS$	: Feature Ranking versus Feature Selection method.

Notations	
$T = \{X_1, .., X_n\}$	: the set of n instances of the considered data set to cluster.
$T^- = \{X_1, .., X_{n^-}\}$	: the subset of n^- objects from T, with $n^- < n$. It represents the reduced collection.
$T^+ = \{X_1, ..X_{n^+}\}$	: the extended set with $n^+ > n$.
UT, UT^-, and UT^+	: the uncertain counter parts of T, T^-, and T^+ within belief function theory framework.
$P = \{C_1, .., C_k\}$	: the cluster partition of k groups.
$P^- = \{C_1, .., C_{k^-}\}$	: the reduced cluster partition of P, with $k^- < k$.
$P^+ = \{C_1, ..C_{k^+}\}$	: the extended cluster partition of P, with $k^+ > k$.
$P_+ = \{C_{1_+}, .., C_{k_+}\}$	: the updated cluster partition after insertion of instance(s) or attribute(s).
$P_- = \{C_{1_-}, .., C_{k_-}\}$	: the updated cluster partition after deletion of instance(s) or attribute(s).
$A = \{A_1, .., A_s\}$	: the s attributes which characterize the data set.
$A^- = \{A_1, .., A_{s^-}\}$	: the reduced attribute set from A, with $s^- < s$.
$A^+ = \{A_1, .., A_{s^+}\}$	: the extended attribute collection, where $s^+ > s$.
i	: the instance index with $1 \leq i \leq n$.
j	: the attribute index with $1 \leq j \leq s$.
l	: the cluster index with $1 \leq l \leq k$.
$X_i = \{x_{i,1}, .., x_{i,s}\}$	: the object of index i.
$X_i^- = \{x_{i,1}, .., x_{i,s^-}\}$	: the reduced data set object characterized by only A^- attributes.
$X_i^+ = \{x_{i,1}, .., x_{i,s^+}\}$	: the extended data set object characterized by A^+ attributes.
$T_- = \{X_1^-, .., X_n^-\}$	: it represents the data set T collection characterized by a reduced feature set A^-.
$T_+ = \{X_1^+, ..X_n^+\}$	: it is the data set T of extended data set objects characterized by an extended feature set A^+.
UT_-, and UT_+	: the uncertain counter parts of T_-, and T_+, where uncertain attribute values are represented via belief function concept.
D	: denotes the distance sum between two items respectively to a set of attributes.
d	: denotes the distance between two items respectively to only one attribute.

List of Figures

3.1 Clustering accuracy results under uncertainty (Mean) 56
3.2 Time indicators results under uncertainty (Mean) 57

4.1 BKM vs $IKBKM$ clustering accuracy results under uncertainty (Mean) 79
4.2 BKM vs $IKBKM$ time indicators results under uncertainty (Mean) 79
4.3 BKM vs $DKBKM$ clustering accuracy results under uncertainty (Mean) 85
4.4 BKM vs $DKBKM$ time indicators results under uncertainty (Mean) 85

5.1 BKM vs $ISBKM$ clustering accuracy results under uncertainty (Mean) 109
5.2 BKM vs $ISBKM$ Time indicators results under uncertainty (Mean) 109
5.3 BKM vs $DSBKM$ clustering accuracy results under uncertainty (Mean) 115
5.4 BKM vs $DSBKM$ Time indicators results under uncertainty (Mean) 115

6.1 BKM vs $INBKM$ clustering accuracy results under uncertainty (Mean) 133
6.2 BKM vs $INBKM$ time indicators results under uncertainty (Mean) 134
6.3 BKM vs $DNBKM$ clustering accuracy results under uncertainty (Mean) 139
6.4 BKM vs $DNBKM$ time indicators results under uncertainty (Mean) 139

List of Tables

1.1 bba's illustrative examples 16

3.1 Belief data set UT 49
3.2 Clustering accuracy (%)(Certain case) 51
3.3 Time performances (Certain case) 52
3.4 Clustering accuracy (%)(Uncertain case) 54
3.5 Running time(s)(Uncertain case) 55
3.6 Iteration number (Uncertain case) 55

4.1 Clustering accuracy (%)(BKM vs $IKBKM$)(Certain case) 74
4.2 Time performances (BKM vs $IKBKM$)(Certain case) 74
4.3 Clustering accuracy (%)(BKM vs $IKBKM$)(Uncertain case) 76
4.4 Running time(s)(BKM vs $IKBKM$)(Uncertain case) 77
4.5 Iteration number (BKM vs $IKBKM$)(Uncertain case) 77
4.6 Clustering accuracy (%)(BKM vs $DKBKM$)(Certain case) 80
4.7 Time performances (BKM versus $DKBKM$)(Certain case) 80
4.8 Clustering accuracy (%)(BKM vs $DKBKM$)(Uncertain case) 82
4.9 Running time(s)(BKM vs $DKBKM$)(Uncertain case) 83
4.10 Iteration number (BKM vs $DKBKM$)(Uncertain case) 83

5.1 Clustering accuracy (%)(BKM vs $ISBKM$)(Certain case) 104
5.2 Time performances (BKM vs $ISBKM$)(Certain case) 104
5.3 Clustering accuracy (%)(BKM vs $ISBKM$)(Uncertain case) 106
5.4 Running time(s)(BKM vs $ISBKM$)(Uncertain case) 107
5.5 Iteration number (BKM vs $ISBKM$)(Uncertain case) 107
5.6 Clustering accuracy (%)(BKM vs $DSBKM$)(%)(Certain case) . . . 110
5.7 Time performances (BKM vs $DSBKM$)(Certain case) 110
5.8 Clustering accuracy (%)(BKM vs $DSBKM$)(Uncertain case) 112

5.9 Running time(s)(BKM vs $DSBKM$)(Uncertain case) 113
5.10 Iteration number (BKM vs $DSBKM$)(Uncertain case) 113

6.1 Clustering accuracy (%)(BKM vs $INBKM$)(Certain case) 129
6.2 Time performances (BKM vs $INBKM$)(Certain case) 129
6.3 Clustering accuracy (%)(BKM vs $INBKM$)(Uncertain case) 131
6.4 Running time(s)(BKM vs $INBKM$)(Uncertain case) 132
6.5 Iteration number (BKM vs $INBKM$)(Uncertain case) 132
6.6 Clustering accuracy (%)(BKM vs $DNBKM$)(Certain case) 134
6.7 Time performances (BKM vs $DNBKM$)(Certain case) 135
6.8 Clustering accuracy (%)(BKM vs $DNBKM$)(Uncertain case) 136
6.9 Running time(s)(BKM vs $DNBKM$)(Uncertain case) 137
6.10 Iteration number (BKM vs $DNBKM$)(Uncertain case) 137

A.1 Description of UCI categorical databases 145
A.2 Lenses database 148
A.3 Belief Lenses database 150

Introduction

Cluster analysis consists in organizing an unlabeled collection of objects or patterns into separate groups based on their similarities. Thus, clustering task refers to the process of grouping objects into classes or clusters such that objects within the same class are similar in a certain sense, and objects from different classes are dissimilar.

A brief exposition of clustering techniques and grouping them according to some specific baseline methodologies, such as hierarchical and partitioning methods are done. These clustering techniques are used in many domains such as medicine, banking, finance, marketing, security, etc. They work under an unsupervised mode where the class label of each object in the training set is not known a priori.
In addition to these unsupervised classification techniques, there exist those working under a supervised mode where the classes of instances are known in advance helping to construct a model that will be used for classifying new objects. Among them, we mention decision trees, k-nearest neighbor, neural networks, etc.
In this thesis, we will focus on one popular clustering algorithm, as categorical extension of the well-known K-means method (MacQueen, 1967), namely K-modes method (Huang, 1998) since its simplicity and understandability.

The capability to deal with data sets containing any uncertainty form is undoubtedly important due to the fact that this kind of data sets is common in real life data mining applications. However, this problem makes most of the standard classification methods inappropriate for ensuring their task in such context. In order to overcome this drawback, the idea is to combine clustering methods with theories managing uncertainty. Among these theories, let us mention probability theory, fuzzy set theory, possibilitic theory, the belief function theory, the rough set theory, etc.

This work is intended for belief clustering due to its advantages and ability to cluster uncertain data sets. The handled uncertainty, here, appears in attributes val-

ues of data sets to cluster. Only symbolic databases are considered.

Indeed, to deal with this uncertainty form, we will choose the belief function theory. Our choice of this non standard theory is motivated by the fact that this theory offers the possibility to represent and manage uncertain information numerically, partial as well as total ignorance. Furthermore, it is an appropriate framework to handle total certainty and probabilistic knowledge. Thereby, we will adopt BKM (Ben-Hariz et al., 2006) method which is based on both the standard K-modes method (SKM) (Huang, 1998) and the belief function theory (Shafer, 1976).
Let us mention that the other developed belief clustering methods (Denoeux and Masson, 2002; Denoeux and Masson, 2004; Masson and Denoeux, 2008), which are all based on the K-means approach, suppose that attribute values of data sets to cluster are known with certainty and that the managed uncertainty concerns only the cluster memberships.

Likewise, the belief function theory is appropriate to handle uncertainty in classification problems, both supervised and unsupervised ones, as already proved in the literature. As examples, we cite belief decision trees (Elouedi et al., 2001), belief K-nearest neighbors (Denoeux, 1995; Denoeux, 2000), belief clustering (Ben-Hariz et al., 2006; Denoeux and Masson, 2002; Denoeux and Masson, 2004; Masson and Denoeux, 2008), belief rough set classifier (Trabelsi et al., 2011), etc.

Thereafter and in order to improve the clustering quality of such belief clustering, we propose a new initialization method based on the dissimilarity maximization under uncertain context. Indeed, by applying it to BKM approach, we obtain new BKM version. This initial partition selection technique will be also considered by the dynamic belief clustering techniques developed under this thesis work.

The static belief clustering approaches (Ben-Hariz et al., 2006; Denoeux and Masson, 2002; Denoeux and Masson, 2004; Masson and Denoeux, 2008), like the standard certain ones, do not well perform their task in a dynamic environment. Hence, we suggest to extend static clustering concept to deal with different dynamic environment forms. Thus, our thesis deals with the problem of learning cluster partition from data set characterized by uncertain attribute values under belief function theory within dynamic environment. The Transferable Belief Model (TBM) is the considered belief function interpretation here.

Our work is motivated by the need to update clustering solution in such uncertain and non-static environment. For example to plan a loan policy, banks use, on one

hand, client attributes which may be known with uncertainty or partially. On the other hand, they must consider the different classification parameters which change over time.
In fact, in order to compute the loan eligibility, a bank starts by looking a list of client characteristics namely salary, age and gender. All these details are available from the beginning. Then, they segment their client databases into eligible and ineligible customers. This is an initial possible segmentation. After that, other information may be added or old ones may be eliminated. As exemplar, note another loan payment to incorporate and gender characteristic to suppress. These latter may be some expert decisions. Furthermore, other customers must be considered which are those of new bank agencies and old ones must be reduced since they are no longer among the clients of these banks. Otherwise, each customer database which is already classified as eligible and ineligible items may be updated to take into account non well-classified customers. Consequently, each client will be judged as eligible, ineligible or undecided.

Therefore, we propose three dynamic clustering systems based on both K-modes method and belief function theory in order to deal with such above aspects. First one is called Belief Clustering for Dynamic Partition (BCDP) which considers the cluster partition movement for static data sets, more precisely the cluster collection cardinality. The second is Belief Clustering for Dynamic Data set Features (BCDDF) which inserts or eliminates possible feature(s) judged as relevant/irrelevant one(s) to improve the clustering quality. The last one is Belief Clustering for Dynamic Data set Items (BCDDI) that helps at updating cluster partition in order to add or delete a part of our data set object collection.
These proposed solutions aim, all, at updating an already attained cluster partition of uncertain data when any change is detected concerning cluster number, attribute set cardinality or data set object number.

To show the effectiveness and performance of our dynamic clustering approaches, both incremental and decremental ones, we carry some experimentations on real databases which are modified to represent the uncertainty via belief function theory concepts. This validation task is done based on three evaluation criteria clustering accuracy, time requirement and finally number of iterations necessary to cluster partition stability. Then, we compare the results with those obtained from static counter-parts always within uncertainty.

This thesis contains six chapters. Chapters 1 and 2 survey background material on both belief function theory and clustering topic. Thus, Chapter 1 presents this non-classical theory of uncertainty. An overview of basic concepts and special belief

functions is provided. Then other notions, from the combination rules to the decision making, are defined. We detail in Chapter 2 the basics of clustering task and provide one representative taxonomy of such techniques. This chapter also exposes the limits of standard clustering techniques as well as the developed methods which are the corresponding solutions in the literature.
Before going into the main topics of this thesis work, the belief clustering to accommodate uncertainty will be listed in Chapter 3, as well as a new initialization method for BKM method as a possible improvement to the standard random initialization. In fact, this developed initialization method under uncertainty will be considered by all proposed dynamic approaches developed in this thesis.

Our work is dedicated, mainly, to dynamic clustering under uncertainty where a cluster collection must be updated following any change. Thus in this study, some dynamic clustering methods are developed to deal with clustering task within such non-stationary environment. Chapter 4 details appropriate approaches generated to handle dynamic cluster partition of static data sets. Chapter 5 exposes extended techniques for dynamic database features. Chapter 6 presents those for dynamic data set items. Indeed, all proposed approaches are based on the well-known K-modes method and the belief function theory to handle uncertainty. Note that the core concept during our dynamic clustering study is the similarity/dissimilarity measure. All these chapters provide experimental results showing the merits of our dynamic algorithms compared with standard static approach by considering the same experimental hypotheses.

Finally, a conclusion summarizes this work via a general overview of our contributions and suggests some directions for future research. An appendix is provided at the end of this report to complete this thesis. This latter presents the UCI databases and details the artificial creation procedure to introduce the uncertainty and gives the belief lenses database as illustrative example. The used evaluation criteria are also defined within this appendix.

1
Belief function theory

1.1 Introduction

The necessity of representing and handling any forms of imprecision or uncertainty of data leads to the development of several non classical mathematical theories, such as the belief function theory.
The theory of belief functions is considered as a useful tool to represent and manage uncertain knowledge because of its flexibility. It also known as Dempster-Shafer theory or Evidence theory. Initially, it is introduced by Dempster in the late 1960's (Dempster, 1967). Further, it is developed by Shafer in (Shafer, 1976). Indeed, this theory was proposed as a model to represent quantified beliefs.

More recent interpretations of Dempster-Shafer theory have been proposed namely the Transferable Belief Model (TBM) (Smets, 1990b; Smets and Kennes, 1994; Smets and Kruse, 1997), the lower probability model (Walley, 1991), the Dempster's model (Dempster, 1967), etc. The theory of belief functions has become an efficient tool for knowledge representation and uncertain reasoning in many scientific areas. This theory has been widely applied to business, engineering, and medical problems. Its applications include auditing, decision support, financial asset evaluation, information fusion, information retrieval, knowledge management, medical diagnosis, network security, object classification, risk management, etc.

In this work, we deal with the interpretation of the belief function theory as explained by the TBM interpretation. So, it provides a mathematical tool to treat subjective, and personal judgments corresponding to the different parameters of any classification problems and can be easily extended to deal with objective probabilities. Partial beliefs are expressed in a much more flexible way than probability functions do. It also permits to handle partial or even total ignorance concerning classification parameters. Information is handled at the credal level, where beliefs can be represented and combined. Furthermore, decisions are made at the pignistic level through

the pignistic transformation. Reasoning under uncertainty and decision-making are two different cognitive tasks which can be handled at two distinct levels. This is the core hypothesis underlying the TBM context. Indeed, this theory offers interesting tools to combine several pieces of evidence (Shafer, 1986; Smets, 1990a; Smets, 1993), like the conjunction and the disjunction rules of combination.

In this chapter, we introduce an overview of belief function theory and present its basic concepts which are, all, illustrated by examples within the adopted framework namely the TBM framework.

The remainder is organized as follows. First, the belief function basic concepts are exposed. In particular, different representations of a belief functions are defined. Some special belief functions are, then, listed. Hereafter, combination rules are summarized. Next, another concept of this theory is detailed namely the discounting. After that, we present the pignistic level used to make a decision. All the advanced concepts are illustrated by examples.

1.2 Basic concepts

1.2.1 Frame of discernment

The universe of discourse or *the domain of reference* also referred to *the frame of discernment* is a finite non empty set of elementary events to a given problem, called Θ. These events are assumed to be exhaustive and mutually exclusive. This set contains hypotheses about some problem domain.
All the subsets of Θ belong to *the power* of Θ, denoted by 2^Θ and are defined as follows:

$$2^\Theta = \{A : A \subseteq \Theta\} \tag{1.1}$$

Each element of 2^Θ is called *a proposition* or *an event.*

The elements of Θ are called the elementary propositions.
Note that the two extreme events belonging to this power set are : the empty set $\emptyset$ which corresponds to the contradiction relative to the impossible proposition and the set Θ respectively to the tautology or the certain proposition.

In Shafer's model (Shafer, 1976), the frame of discernment is defined as the set of mutually *exclusive* and *exhaustive* hypotheses which means that the solution to a given problem is unique and is necessarily included in this frame of discernment.

However, in the TBM, Smets contests the exhaustivity of the frame of discernment considering that it is difficult to know a priori all the possible hypotheses for some problems and introduces the *open*-and *closed-world assumptions* (Smets, 1990b; Smets and Kennes, 1994; Smets and Kruse, 1997).

Under the *open-world assumption* the frame is not necessarily exhaustive admitting, thus the existence of an unknown proposition (not defined in the frame of discernment) that might be a solution to the considered problem, whereas under the *closed-world assumption* the frame of discernment is exhaustive.

Example 1.1 *We suppose that a captured network traffic can be classified as Normal, Probe (Attacker tries to gain information about the target host) or Denial of Service DoS (Attacker tries to prevent legitimate users from using a service). These two latter are main attack types. Let us mention that there are two other attack categories, not considered in our examples, namely User to Root Attacks (U2R) and Remote to User Attack (R2L).*
Thus, $\Theta = \{Normal, Probe, DoS\}$.

The corresponding power set of Θ *is:*
$2^{\Theta} = \{\emptyset, \{Normal\}, \{Probe\}, \{DoS\}, \{Normal, Probe\}, \{Normal, DoS\}, \{Probe, DoS\}, \{Normal, Probe, DoS\}\}$

1.2.2 Basic belief assignment

The *basic belief assignment* (*bba*), called initially by Shafer *basic probability assignment*, in (Shafer, 1976), represents the impact of a piece of evidence on the different subsets of the frame of discernment Θ. The *bba* is defined as follows:

$$m : 2^{\Theta} \rightarrow [0, 1]$$
$$\sum_{A \subseteq \Theta} m(A) = 1 \tag{1.2}$$

Each quantity $m(A)$, named *basic belief mass* (bbm), is considered to be the part of belief that supports the event A, and that, due to the lack of information, does not support any strict subset of A.

We discuss, here, the nature of the frame of discernment on which the degree of belief is established and presents the distinction between the *open* and the *closed-world* assumptions.

Shafer has initially imposed the normality condition expressed by $m(\emptyset) = 0$. This condition reflects the fact that no belief ought to be allocated to the empty set. Such *bba* is called *a normalized basic belief assignment.*

This condition is relaxed in the TBM, the allocation of a positive mass to the empty set ($m(\emptyset) > 0$) is interpreted as a consequence of the *open-world assumption.* A *bba* that verifies this condition is said to be *subnormal* or *unnormalized.* In other words, the hypotheses making up the frame of discernment are considered not exhaustive (Smets, 1990a). The meaning of the basic belief mass given to $\emptyset$ is analyzed in (Smets, 1992b).

Example 1.2 *Assume the same Θ of Example 1.1.*
The bba related to a piece of evidence concerning the attack type is defined as follows:
$m(\{Probe\}) = 0.3$
$m(\{Normal, Probe\}) = 0.6$
$m(\{DoS\}) = 0.1$

For example, 0.1 represents the part of belief exactly supporting that the attack is DoS.

1.2.3 Focal elements, body of evidence, core

The subsets A of the frame of discernment Θ such that $m(A)$ is strictly positive, are called *the focal elements* of the *bba* m.

The pair (F, m) is called *a body of evidence* where F is the set of all the focal elements relative to the *bba* m.

The union of all the focal elements of m is called *the core* and is defined as follows:

$$\varphi = \bigcup_{A:m(A)>0} A \tag{1.3}$$

Example 1.3 *Let us continue with our example (Example 1.2).*

The subsets $\{Probe\}$, $\{Normal, Probe\}$ and $\{DoS\}$ are the focal elements of the bba m.

So,(F,m) is called the body of evidence such that: $F = \{\{Probe\}, \{Normal, Probe\}, \{DoS\}\}$

The core is defined as follows: $\varphi = \{Probe\} \cup \{Normal, Probe\} \cup \{DoS\} = \Theta$

1.2.4 Belief function

A belief function, denoted *bel*, corresponding to a specific *bba* m, assigns to every subset A of Θ, the sum of masses of belief committed to every subset of A by m (Shafer, 1976).
The belief function *bel* is defined as follows:

$$bel : 2^{\Theta} \rightarrow [0, 1]$$

$$bel(A) = \sum_{\phi \neq B \subseteq A} m(B) \tag{1.4}$$

It must satisfy this condition: $bel(\emptyset) = 0$. While, usually $bel(\Theta) = 1$ which is assured within closed-world assumption as in (Shafer, 1976).
This last imposed condition by Shafer can be ignored and only the following inequality namely $bel(\Theta) \leq 1$ must be required.
The belief function *bel* represents the total belief that one commits to A without being also committed to $\bar{A}$. The bbm $m(\emptyset)$ is not included in bel(A) as $\emptyset$ is both a subset of A and $\bar{A}$.

Properties:

- Sub-additivity:

$$bel(A) + bel(\bar{A}) \leq 1 \tag{1.5}$$

  Contrary to the probability theory the belief function theory increasing beliefs on a proposition A does not necessary require the decrease of beliefs on $\bar{A}$.

- Monotonicity:

$$A \subseteq B \Longrightarrow bel(B) \geq bel(A) \tag{1.6}$$

  Within the closed-world assumption, Θ will get the highest value of bel, whereas $\emptyset$ will get the lowest value.

- $For A, B \subseteq \Theta, A \cap B = \emptyset,$

$$bel(A \cup B) \geq bel(A) + bel(B) \tag{1.7}$$

- m(A) may be expressed by the values of *bel* as follows (Smets, 2002):

$$m(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|} bel(B), \forall A \subseteq \Theta, A \neq \emptyset \tag{1.8}$$

- The bbm $m(\emptyset)$ is computed as follows :

$$m(\emptyset) = 1 - bel(\Theta) \tag{1.9}$$

Example 1.4 *The belief function bel corresponding to the same example (Example 1.2) is defined as follows:*
$bel(\emptyset) = 0;$
$bel(\{Normal\}) = 0;$
$bel(\{Probe\}) = 0.3;$
$bel(\{DoS\}) = 0.1;$
$bel(\{Normal, Probe\}) = 0.6 + 0.3 = 0.9;$
$bel(\{Normal, DoS\}) = 0.1;$
$bel(\{Probe, DoS\}) = 0.1 + 0.3 = 0.4;$
$bel(\Theta) = 0.3 + 0.6 + 0.1 = 1$
For example, 0.9 is the total belief committed to the proposition $\{$*Normal,Probe*$\}$.

1.2.5 Implicability function

The implicability of A is the sum of masses allocated to subsets of A defined as follows:

$$b(A) = \sum_{B \subseteq A} m(B) \tag{1.10}$$

It can be expressed differently by:

$$b(A) = bel(A) + m(\emptyset) \tag{1.11}$$

Thus, we can see that functions b and *bel* coincide when $m(\emptyset) = 0$. However, these two functions need to be distinguished in the subnormal case, so, *bel* has easier interpretation in terms of degrees of belief.

Example 1.5 *By considering the same example (Example 1.2), where* $m(\emptyset) = 0$*, the b function coincides with the bel one.*

1.2.6 Plausibility function

The plausibility function pl associated with a mass function m quantifies the maximum amount of belief that could be given to a subset A of the frame of discernment. It is equal to the sum of the bbm's relative to subsets B compatible with A. The plausibility function *pl* is defined as follows (Barnett, 1991):

$$pl : 2^{\Theta} \mapsto [0, 1]$$

$$pl(A) = \sum_{A \cap B \neq \phi} m(B) \tag{1.12}$$

There is a simple relationship between the belief function *bel* and the plausibility function *pl* associated with a mass function m : for $A \subseteq \Theta$

$$pl(A) = bel(\Theta) - bel(\bar{A}) \tag{1.13}$$

$$= \sum_{B \subseteq \Theta} m(B) - \sum_{B \subseteq \bar{A}} m(B) \tag{1.14}$$

where $\bar{A}$ denotes the complement of the subset A of Θ.

Properties:

- For both closed-world and the open-world assumptions, we have:

$$bel(\emptyset) = pl(\emptyset) = 0 \tag{1.15}$$

- Concerning the frame of discernment, we have:
  For the open-world assumption

$$pl(\Theta) = bel(\Theta) = 1 - m(\emptyset) \leq 1 \tag{1.16}$$

  For the closed-world assumption

$$pl(\Theta) = bel(\Theta) = 1 \tag{1.17}$$

- Over additivity:

$$pl(A) + pl(\bar{A}) \geq 1 \tag{1.18}$$

- Monotonicity:

$$A \subset B \Longrightarrow pl(B) \geq pl(A) \tag{1.19}$$

- $For A, B \subseteq \Theta, A \cap B = \emptyset,$

$$pl(A \cup B) \leq pl(A) + pl(B) \tag{1.20}$$

- $For A \subseteq \Theta,$

$$bel(A) \leq pl(A) \tag{1.21}$$

Example 1.6 *The plausibility function pl corresponding to the bba (see Example 1.2) is defined as follows:*
$pl(\emptyset) = 0;$

$pl(\{Normal\}) = 0.6;$
$pl(\{Probe\}) = 0.3 + 0.6 = 0.9;$
$pl(\{DoS\} = 0.1;$
$pl(\{Normal, Probe\}) = 0.3 + 0.6 = 0.9;$
$pl(\{Normal, DoS\}) = 0.6 + 0.1 = 0.7;$
$pl(\{Probe, DoS\}) = 0.3 + 0.6 + 0.1 = 1;$
$pl(\{\Theta\}) = 0.3 + 0.6 + 0.1 = 1$
For example, 0.1 represents the maximum degree of belief that could be given to the proposition $\{DoS\}$.

1.2.7 Commonality function

Another function related to a basic belief function m is the *commonality function* q. The meaning of the commonality function is not obvious. However, it may represent the total mass that is free to move to every element of A (Barnett, 1991). It is defined as follows:

$$q : 2^{\Theta} \mapsto [0, 1]$$
$$q(A) = \sum_{A \subseteq B} m(B) \tag{1.22}$$

Properties:

- The commonality value relative to the empty set is defined as follows:

$$q(\emptyset) = 1 \tag{1.23}$$

- The commonality value relative to the whole frame of discernment is defined as follows:

$$q(\Theta) = m(\Theta) \tag{1.24}$$

Example 1.7 *The commonality function q corresponding to the same bba, as in the previous examples, is defined as follows:*
$q(\{\emptyset\}) = 1;$
$q(\{Normal\}) = 0.6;$
$q(\{Probe\}) = 0.3 + 0.6 = 0.9;$
$q(\{DoS\}) = 0.1;$
$q(\{Normal, Probe\}) = 0.6;$
$q(\{Normal, DoS\}) = q(\{Probe, DoS\}) = 0;$
$q(\{\Theta\}) = 0$

1.2.8 Belief intervals

The information contained in *bel* concerning a given subset A may be conveniently expressed by the interval $[bel(A), pl(A)]$ called *belief interval* (BI). It lies in the unit interval $[0, 1]$.

If BI(A) has an interval $[0, 1]$, it means that no information about hypotheses is available. On the other hand, if the interval is $[1, 1]$, it means that A has been completely confirmed.

Example 1.8 *Let us consider the setting of Examples 1.4 and 1.6.*
For example, the belief interval corresponding to the proposition $\{Normal\}$ *is given by the interval [0, 0.6], whereas,* $BI(\{Normal, DoS\}) = [0.1, 0.7]$.

1.3 Special belief functions

In order to express particular situations, several special belief functions were developed.

1.3.1 Vacuous belief function

A belief function is said to be *vacuous* if Θ is its unique focal element (Shafer, 1976):

$$m(\Theta) = 1 \ and \ m(A) = 0 \ for \ all \ A \subseteq \Theta, A \neq \Theta \tag{1.25}$$

In other words :

$$bel(\Theta) = 1 \ and \ bel(A) = 0 \ for \ all \ A \subseteq \Theta, A \neq \Theta \tag{1.26}$$

Such *bba*, where Θ is the unique focal element, quantifies the state of *total ignorance* since there is no support given to any strict subset of Θ.

1.3.2 Categorical belief function

It is a normalized belief function such that its *bba* is defined as follows (Mellouli, 1987):

$$m(A) = 1 \ for \ some \ A \subseteq \Theta \ and \ m(B) = 0, \ for \ B \subseteq \Theta, B \neq A \tag{1.27}$$

The unique focal element A is different from the frame of discernment Θ.

1.3.3 Bayesian belief function

A *Bayesian belief function* is a belief function where the focal elements are all singletons. It is defined by the following (Shafer, 1976):

$$m(A) > 0 \ for \ A \subset \Theta \ and \ |A| = 1 \tag{1.28}$$

In other words:

$$bel(A \cup B) = bel(A) + bel(B) \ whenever \ A, B \subset \Theta \ and \ A \cap B = \emptyset \tag{1.29}$$

with $bel(\emptyset) = 0$ and $bel(\Theta) = 1$

Remarks:

- In such context, *bel* becomes a probability distribution
- As in the probability theory :

$$bel(A) + bel(\bar{A}) = 1 \ for \ A \subseteq \Theta \tag{1.30}$$

- $bel = pl$

Probability distributions may be easily represented within belief function theory. Indeed, for the singleton focal sets, m is termed *bayesian* and $pl = bel = P$, where P is a probability distribution.

1.3.4 Consonant belief function

A *consonant belief function* is a belief function whose focal elements are nested. Its corresponding definition is as follows (Shafer, 1976):

$$m(A_i) > 0, i \in \{1, .., n\}, such \ that \ A_1 \subset A_2 \subset ... \subset A_n, \ \forall A_i \subseteq \Theta \tag{1.31}$$

In that case,

$$bel(A \cap B) = min[bel(A), bel(B)] \ \forall \ A, B \ \subseteq \Theta \tag{1.32}$$

or equivalently

$$pl(A \cup B) = max[pl(A), pl(B)] \ \forall \ A, B \ \subseteq \Theta \tag{1.33}$$

Let us note that the associated plausibility function of one consonant belief function is a *possibility* measure. It verifies $pl(A \cup B) = max(pl(A), pl(B))$ for all $A, B \subseteq \Theta$. The

corresponding belief function is the dual *necessity* measure (Dubois and Prade, 1988).

From the above, it is easy to see that belief function theory is thus, a very general framework, including possibility as well as probability theories. In other words, these two last non classical theories are considered as particular cases of the belief function one.

1.3.5 Certain belief function

A *certain belief function* is a categorical belief function such that it has only one focal element and which should be a singleton. Its corresponding *bba* is defined as follows:

$$m(A) = 1 \ and \ m(B) = 0 \ for \ all \ B \neq A, \ B \subseteq \Theta, \ and \ |A| = 1 \tag{1.34}$$

This function represents a state of total certainty as it assigns all the belief to a unique elementary event.

1.3.6 Simple support function

A belief function is said to be a *simple support function (SSF)* if it has at most one focal element different from the frame of discernment Θ. This focal element is called *the focus* of the SSF.

A SSF is defined as follows:

$$m(X) = \begin{cases} \omega & \text{if X} = \Theta \\ 1-\omega & \text{if X = A for some } A \subseteq \Theta \\ 0 & otherwise. \end{cases} \tag{1.35}$$

Where A is the focus and $\omega \in [0, 1]$.

1.3.7 Dogmatic and non-dogmatic belief functions

A belief function is called a dogmatic belief function if and only if its corresponding *bba* m is defined such that $m(\Theta) = 0$. This case involves some previous cases namely certain belief function and categorical belief function.
The non-dogmatic belief function is detected when $m(\Theta) > 0$ (Smets, 1995).

Example 1.9 *Let us consider the same frame of discernment* $\Theta = \{Normal, Probe, DoS\}$, *where* $\{Normal\}$, $\{Probe\}$ *and* $\{DoS\}$ *are the corresponding elementary propositions.*

The following table (Table 1.1), presents the different kinds of belief functions, exposed above, as illustrative examples.

Table 1.1: bba's illustrative examples

$A \subseteq \Theta$	Vacuous	Categorical	Bayesian	Consonant	Certain	SSF	Normalized
$\emptyset$	0	0	0	0	0	0	0
$\{Normal\}$	0	0	0.2	0.1	0	0	0.6
$\{Probe\}$	0	0	0.4	0	1	0	0
$\{Normal, Probe\}$	0	1	0	0.6	0	0	0
$\{DoS\}$	0	0	0.4	0	0	0	0
$\{Normal, DoS\}$	0	0	0	0	0	0.6	0
$\{Probe, DoS\}$	0	0	0	0	0	0	0.3
$\{Normal, Probe, DoS\}$	1	0	0	0.3	0	0.4	0.1

Respectively to these proposed functions, the categorical, the bayesian and the certain ones are all dogmatic belief functions, while, the vacuous, the consonant, the SSF and the normalized cases represent non-dogmatic functions examples.

1.4 The combination rules

One of the most important operations in the Dempster-Shafer theory is the aggregation of several sources of evidence. The belief function theory, as understood in the TBM framework, offers interesting rules for aggregating the basic belief assignments defined over the same frame of discernment and induced from distinct pieces of evidence (Smets, 1992a) and provided by two (or more) sources of information (Shafer, 1986; Smets, 1990a; Smets, 1993).

We can combine them in at least two ways: conjunctively or disjunctively using respectively the conjunctive or disjunctive rules of combination.

The choice of one of these rules of combination to merge pieces of evidence may be guided by meta-belief concerning the reliability of the considered sources. In fact, if we know that both sources of information are fully reliable, then we combine them conjunctively. However, if we know that at least one of the two sources is reliable, then we combine them disjunctively (Smets, 1993).

Let m_1 and m_2 be two bba's defined on the same frame of discernment Θ. These two bba's are collected by two 'distinct' pieces of evidence and induced by two experts

(information sources).

1.4.1 The conjunctive case

For this case, we suppose the reliability of both sources information. Then, this conjunctive combination rule is defined as follows:

$$(m_1 \textcircled{\cap} m_2)(A) = \sum_{B,C \subseteq \Theta; B \cap C = A} m_1(B) m_2(C) \tag{1.36}$$

Properties:

- The conjunctive combination rule is commutative.

$$(m_1 \textcircled{\cap} m_2) = (m_2 \textcircled{\cap} m_1) \tag{1.37}$$

- It ensures the associativity property

$$(m_1 \textcircled{\cap} m_2) \textcircled{\cap} m_3 = m_1 \textcircled{\cap} (m_2 \textcircled{\cap} m_3) \tag{1.38}$$

- The neutral element of conjunctive combination rule is the vacuous belief function m_0, thus:

$$m \textcircled{\cap} m_0 = m \tag{1.39}$$

- Compositionality:
  $(m_1 \textcircled{\cap} m_2)(A)$ is function of A, m_1 and m_2

- The conjunctive rule is not idempotent, so:

$$m \textcircled{\cap} m \neq m \tag{1.40}$$

Remark that Dempster's rule of combination is defined as the conjunctive combination of two normalized bba followed by normalization step as follows:

$$m_1 \oplus m_2(A) = K(m_1 \textcircled{\cap} m_2)(A) \tag{1.41}$$

where $K^{-1} = 1 - (m_1 \textcircled{\cap} m_2)(\emptyset)$ and $m_1 \oplus m_2(\emptyset) = 0$.
K is known as the normalization factor. In fact, the conjunctive rule is considered as the unnormalized Demspter rule of combination dealing with the closed-world assumptions.

Example 1.10 *Let us consider the frame of discernment given in the Example 1.1. Assume we have two distinct sensors S_1 and S_2 providing evidence on the attack type. Two bba's m_1 and m_2 are relative to respectively S_1 and S_2 and defined as follows:*
$m_1(\{Normal\}) = 0.6$;
$m_1(\{Normal, DoS\}) = 0.2$;
$m_1(\{\Theta\}) = 0.2$.

$m_2(\{DoS\}) = 0.4$;
$m_2(\{Normal, DoS\}) = 0.3$
$m_2(\Theta) = 0.3$

Applying the conjunctive rule of combination, we get:
$(m_1 \textcircled{\cap} m_2)(\emptyset) = 0.24$;
$(m_1 \textcircled{\cap} m_2)(\{Normal\}) = 0.36$;
$(m_1 \textcircled{\cap} m_2)(\{Probe\}) = 0$;
$(m_1 \textcircled{\cap} m_2)(\{DoS\}) = 0.16$;
$(m_1 \textcircled{\cap} m_2)(\{Normal, DoS\}) = 0.18$;
$(m_1 \textcircled{\cap} m_2)(\{Normal, Probe\}) = 0$;
$(m_1 \textcircled{\cap} m_2)(\{Probe, DoS\}) = 0$;
$(m_1 \textcircled{\cap} m_2)(\Theta) = 0.06$.

1.4.2 The disjunctive case

The disjunctive rule of combination is the dual of the conjunctive one, it builds the *bba* representing the impact of two pieces of evidence when we only know that at least one of them is to be accepted, but we do not know which one. This rule is defined as follows (Smets, 1993) :

$$(m_1 \textcircled{\cup} m_2)(A) = \sum_{B,C \subseteq \Theta; B \cup C = A} m_1(B) m_2(C) \tag{1.42}$$

Properties:

- The disjunctive combination rule (as the conjunctive rule of combination) is commutative and associative.
- The absorbing element, within the disjunctive rule of combination, is $m_\emptyset$, such that $m_\emptyset(\emptyset) = 1$. Hence:

$$m \textcircled{\cup} m_\emptyset = m \tag{1.43}$$

- Non-idempotency

$$m \textcircled{\cup} m \neq m \tag{1.44}$$

Example 1.11 *By using the two bba's m_1 and m_2 relative respectively to the two distinct pieces of evidence, the result of applying the disjunctive rule is as follows:*
$(m_1 \textcircled{\cup} m_2)(\{Normal, DoS\}) = 0.56$;
$(m_1 \textcircled{\cup} m_2)(\Theta) = 0.44$.

Since the conjunctive rule as well as the disjunctive one are both commutative and associative, combining several pieces of evidence induced from distinct information sources (either conjunctively or disjunctively) may be easily ensured by applying repeatedly the chosen rule.

Remark:
There are other combination rules such as Dubois and Prade's rule (Dubois and Prade, 1988) which is a mix of a conjunctive and a disjunctive behavior and Yager's rule (Yager, 1987). This latter also assumes that the sources tell the truth when they are not in conflict.
Let us eventually note that, in contrast to conjunctive and disjunctive rules, Yager's rule and Dubois and Prade's rule are both not associative.

1.5 Discounting belief functions

Knowing that experts are not fully reliable, a method of discounting seems imperative to update experts' beliefs by taking into account their reliability. This is done to consider the level of expertise of each information source.
If the source is perfectly or totally reliable, then the beliefs does not require an update. On the contrary, when the source is not reliable at all, the mass should be reduced to the vacuous belief function representing total ignorance.
Finally, if the source is partially reliable, the mass should be altered in such a way that the updated values of the mass are proportional to the reliability of this source.

Let (1 - α) be *the degree of trust* assigned to the expert. It quantifies the strength of reliability given to this expert. While α is named *the discounting factor*.
The *bba* defined on the set {reliable, not reliable} is computed as follows (Smets, 1992a):

$$m(reliable) = 1 - \alpha \text{ and } m(not\ reliable) = \alpha \tag{1.45}$$

To update the expert opinion, the following formula must be applied.

$$m^{\alpha}(A) = (1-\alpha)m(A) \ for \ A \subseteq \Theta \tag{1.46}$$

$$m^{\alpha}(\Theta) = \alpha + (1-\alpha)m(\Theta) \tag{1.47}$$

When $\alpha = 0$, the source or expert is totally reliable. However, for $\alpha = 1$, we are faced on the total ignorance case.

Example 1.12 *For one degree of reliability of the expert equal to* 0.6*, the discounted bba of Example 1.2 is as follows:*

$m(\{Probe\}) = 0.4 * 0.3 = 0.12;$
$m(\{Normal, Probe\}) = 0.4 * 0.6 = 0.24;$
$m(\{DoS\}) = 0.4 * 0.1 = 0.04;$
$m(\{\Theta\}) = 0.6 + 0.4 * 0 = 0.6.$

1.6 Decision making

To make a decision, we have to select the most likely hypothesis which may be difficult to solve directly with the basics of the belief function theory where bbm's are given not only to singletons but also to subsets of hypotheses.
In the following, we present one solution allowing to ensure the decision making within the belief function theory namely the pignistic probability developed within the Transferable Belief Model (TBM) (Smets, 1988; Smets and Kennes, 1994; Smets and Kruse, 1997; Smets, 1998). Other propositions will be presented like the maximum of credibility and the maximum of plausibility (Janez, 1996).

1.6.1 Pignistic probability

At any time, beliefs are entertained and updated at the credal level, the pignistic level (pignus = a bet (in Latin)) appears only when a decision needs to be made. Thus, when a decision must be made, beliefs presented, at the credal level, induce a probability measure at the pignistic level.

Based on the TBM, two level mental models are proposed, as follows:

- The credal level where beliefs are entertained and represented by belief functions.
- The pignistic level where beliefs are used to make decisions and represented by probability functions called *the pignistic probabilities.*

When a decision must be made, beliefs held at the credal level induce a probability measure at the pignistic measure denoted $BetP$ (Smets, 1998).
The link between these two functions is achieved by the pignistic transformation formula.

$$BetP(A) = \sum_{B \subseteq \Theta} \frac{|A \cap B|}{|B|} \frac{m(B)}{(1 - m(\phi))}, \text{ for all } A \in \Theta \tag{1.48}$$

Example 1.13 *As all previous examples, the same bba is considered. (Example 1.2). To compute the corresponding pignistic probability BetP, is necessary to make a decision. So, we get:*
$BetP(\{Normal\}) = 0.3$, $BetP(\{Probe\}) = 0.6$ *and* $BetP(\{DoS\}) = 0.1$
We note that is more probable that the attack category will be Probe.

1.6.2 Maximum of credibility

It consists in choosing the hypothesis having the highest value of the belief function bel, that is the most credible hypothesis.
Decision based on the maximum of credibility is considered as a pessimistic approach since it chooses the best hypothesis based on the minimum chance to realize (Janez, 1996).

Example 1.14 *Let us continue with the Example 1.4, and to make a decision by using the maximum of credibility method. We recall that:*
$bel(\{Probe\}) = 0.3$,
$bel(\{Normal\}) = 0$,
$bel(\{DoS\}) = 0.1$.
The largest value of bel is the one assigned to the hypothesis $\{Probe\}$. *Thus, according to the maximum of credibility, the attack type is Probe.*

1.6.3 Maximum of plausibility

Contrary to the maximum credibility criterion, this criterion is considered as optimistic since it takes into account the maximum of chance to realize each hypothesis. It consists in choosing the hypothesis having the highest value of the plausibility function pl which means that we support the hypothesis that gives the less evidence for the contrary hypothesis.

Example 1.15 *Based on the plausibility function values of the three considered elementary events (Example 1.6) to make a decision, we recall that:*
$pl(\{Normal\}) = 0.6$;
$pl(\{Probe\} = 0.9$;

$pl(\{DoS\}) = 0.1$.
Thus, Probe attack is detected.

1.7 Conclusion

In this chapter, the fundamental concepts of the Transferable Belief Model (TBM), one belief function theory interpretation, have been exposed and illustrated by examples. This model distinguishes two cognitive tasks: reasoning and decision making. Reasoning is handled at the credal level where beliefs of experts are represented by belief functions and merged using combination rules. Decision-making is reserved to the pignistic level where beliefs must be transformed into probability measures in order to be able to use classical decision theory.
As presented, the belief function theory provides a convenient tool to handle uncertainty in classification problems, supervised as well as unsupervised.

The next chapter is devoted to the presentation of clustering analysis as unsupervised classification tool, where the corresponding basic concepts will be exposed. This topic besides belief function theory dealing uncertainty, will be at the core of some contributions of this thesis.

2

Data clustering : An overview

2.1 Introduction

Clustering is a discipline devoted to find and to describe homogeneous groups in data: the clusters. It also called cluster analysis, unsupervised classification or data segmentation (Jain and Dubes, 1988). It partitions large data sets into groups according to their similarity. Indeed, it works under an unsupervised mode when the class label of each data set object is not known a priori, comparing to supervised classification tools such as decision trees, K-nearest neighbors, etc. Thus, the goal of clustering is to separate a finite, unlabeled data set into a finite and discrete set of natural, hidden data structures.

A resulting cluster is a set of entities which are alike, while entities from different clusters are not alike. It can also be described in terms of internal homogeneity and external separation. It means that data objects in the same cluster should be similar to each other. Moreover, data objects in different clusters should be dissimilar from one to another.

Data clustering is under vigorous development. Contributing areas of research include data mining, statistics, machine learning, biology, marketing, etc.
Indeed, among directions, in which clustering is used, we mention the data reduction, market research, biology and bioinformatics. The first one consists in adopting the representatives of the defined clusters instead of processing the whole data collection. The second one uses cluster analysis to partition the general population of consumers into market segments and to better understand the relationships between different groups of consumers. In biology, clustering has many applications. In the fields of plant and animal ecology, clustering is used to describe and to make spatial and temporal comparisons of communities. For bioinformatics, clustering is used to build groups of genes with related expression patterns.

To better understand clustering, this chapter presents the basic concepts of this task such as data types and proximity measures. Next, we define a possible taxonomy of classical clustering approaches. The considered clustering method to deal with different dynamic aspects, handled in this thesis work, is also detailed here namely K-modes method. It is followed by some limits of standard clustering algorithms presentation.

2.2 From data types to data scale

Clustering methods are very much associated with data types. Thereby, in (Anderberg, 1973) the author outlines a categorization of data types as well as of data scales appropriate for cluster analysis. In general, recognizing the type and the scale of data will help in selecting the appropriate clustering algorithm to use.

An attribute can be typed as binary, discrete or continuous. Indeed, a continuous feature takes values from an uncountably infinite range set. In comparison, discrete features only have finite number of values. Finally, binary or dichotomous features are a special case of discrete features when they have exactly two values, such as true or false.

For the data scale, known also as the measurement level, the relative significance of numbers is expressed. It can be qualitative or quantitative scale, recognized as nominal and ordinal respectively to the first scale type and interval and ratio for the second one.

In this thesis, we focus on clustering techniques applied to categorical or discrete data with a nominal qualitative scale.

2.3 Proximity measures

Proximity measure is the generalization of both similarity and dissimilarity ones. This metric is usually measured by a distance function defined on pairs of objects or vectors. A variety of distance measures are in use in the various approaches. Because of the variety of feature types and scales, the distance measure must be chosen carefully.

In the following part, we will briefly expose some proximity measures respectively to continuous and discrete variables. In order to detail these distances, we have to start by defining similarity/dissimilarity matrix or proximity matrix concept.

2.3.1 Similarity/Dissimilarity matrix definition

It stores a collection of proximities that are available for all pairs of n objects belonging to a data set to be clustered. This dissimilarity matrix is often represented by n-by-n D matrix. We assume that all proximity matrices are symmetric.
Given this matrix $D = (d_{i1i2})$, $i1, i2 \in \{1, .., n\}$ where n is the number of data objects to cluster. The entries $d_{i_1 i_2}$ form a dissimilarity measure between entities or instances X_{i1} and X_{i2} iff D satisfies these properties:

1. Symmetry : $d_{i1i2} = d_{i2i1}$. The distance is a symmetric function.
2. Non-negativity or positivity : $d_{i1i2} \geq 0$. d is a non negative number.
3. Nondegeneracy or reflexivity,: $d_{i1i2} = 0$ if $X_{i1} = X_{i2}$. The distance of an object to itself is 0.

   Thus, this function is called a *semi-metric*. When a dissimilarity measure additionally satisfies the following property, is referred to as a distance or metric:

4. Triangle inequality : $d_{i1i2} \leq d_{i1i3} + d_{i3i2}$, for X_{i1},X_{i2}, and $X_{i3} \in T$.

2.3.2 Proximity measures versus data types

The choice of the proximity measure to use depends on the data types. Various measures have been suggested which can be categorized as follows. Only, the most used distance measures will be exposed throughout this section.

Proximity measures for continuous variables

One of the most popular measure is the Euclidean distance. It is computed as follows:

$$d(X_{i1}, X_{i2}) = \sqrt{(x_{i1,1} - x_{i2,1})^2 + (x_{i1,2} - x_{i2,2})^2 + ... + (x_{i1,s} - x_{i2,s})^2} \tag{2.1}$$

where $X_{i1} = (x_{i1,1}, x_{i1,2}, ..., x_{i1,s})$ and $X_{i2} = (x_{i2,1}, x_{i2,2}, ..., x_{i2,s})$ are two s-dimensional data objects, where $i1, i2 \in \{1, .., n\}$.
Other distances such as the Manhattan distance, the Minkowski distance, the Maximum distance and the average distance can be used within this numerical context.

Proximity measures for discrete variables

The distance measures for categorical data sets will be presented seen that it is the general discrete case with more than two values.
Indeed, for discrete features that have more than two states, a simple and direct

strategy is to map them into a larger number of new binary features (Kaufman and Rousseeuw, 1990). However, the drawback of this method is the possibility of introducing too many binary variables.
In fact, the more commonly used method is based on the simple matching criterion (Kaufman and Rousseeuw, 1990). For a pair of s-dimensional objects X_{i1} and X_{i2}, the dissimilarity using the simple matching criterion is obtained as:

$$d(X_{i1}, X_{i2}) = \sum_{j=1}^{s} \delta(x_{i1,j}, x_{i2,j}) \tag{2.2}$$

where $\delta(x_{i1,j}, x_{i2,j}) = \begin{cases} 0 & if\ x_{i1,j} = x_{i2,j} \\ 1 & if\ x_{i1,j} \neq x_{i2,j} \end{cases}$
Note that for binary data sets which are a special case of the previous one, Jaccard coefficient is the popularly referenced in the literature. There are many other similarity measures such as the Simple Matching Coefficient, Dice distance and Pearson one as detailed in (Choi et al., 2010)

Proximity measures for mixed variables

For many real databases, objects are described by a mixture of variable types.
One simple approach is to group each kind of variable together, performing a separate cluster analysis for each variable type. This is feasible if these conducted analyses produce compatible results. However, in real applications, it is unlikely that a separate cluster analysis per variable type will generate compatible results.
A more preferable approach is to process all variable types together, performing a single cluster analysis as result. Thus, suppose that the data set contains s variables or attributes of mixed types. The dissimilarity $d(X_{i1}, X_{i2})$ between objects X_{i1} and X_{i2} is defined as:

$$d(X_{i1}, X_{i2}) = \frac{\sum_{j=1}^{s} \delta_{i1i2}^{(j)} d(X_{i1}, X_{i2})^{(j)}}{\sum_{j=1}^{s} \delta_{i1i2}^{(j)}} \tag{2.3}$$

where the indicator $\delta_{i1i2}^{(j)} = 0$ if either (1) $x_{i1,j}$ or $x_{i2,j}$ is missing, or (2) $x_{i1,j} = x_{i2,j} = 0$ and the considered variable j is asymmetric binary; otherwise, $\delta_{i1i2}^{(j)} = 1$, while $d(X_{i1}, X_{i2})^{(j)}$ is computed dependent on its type respectively to the already reported measures above.

2.3.3 Proximity measures for cluster pairs

In some cases, we need to compute the distance between two clusters.
Indeed, by considering two clusters $C_l = \{X_1, .., X_{n_l}\}$ and $C_r = \{Y_1, .., Y_{n_r}\}$ of sizes n_l and n_r respectively, the following measures may be used. Recall that the d component

is defined based on the data type (as detailed in the previous subsection).

The mean-based distance

It consists in measuring the distance between the means or representatives of the two clusters. This is considered the most popular method and defined as:

$$D(C_l, C_r) = d(Q_l, Q_r) \tag{2.4}$$

where Q_l and Q_r are respectively the means of clusters C_l and C_r.

The nearest neighbor distance

The nearest neighbor distance (Williams and Lambert, 1966) between two clusters is computed by considering the minimum distance between tuples of objects of these two groups. Its definition is as follows:

$$D(C_l, C_r) = min_{1\leq\ i\leq n_l, 1\leq\ o\leq n_r} d(X_i, Y_o) \tag{2.5}$$

The farthest neighbor distance

Contrary to the previous one, this measure (Duran and Odell, 1974) considers the maximum of distance between tuples of cluster instances thus:

$$D(C_l, C_r) = max_{1\leq\ i\leq n_l, 1\leq\ o\leq n_r} d(X_i, Y_o) \tag{2.6}$$

The average neighbor distance

Here, all distances between couples of data objects, of both groups, are computed and summed. The resulting measure (Duran and Odell, 1974) is their sum divided by the product of the cluster cardinalities. By the following, it is defined:

$$D(C_l, C_r) = \frac{1}{n_l * n_r} \sum_{i=1}^{n_l} \sum_{o=1}^{n_r} d(X_i, Y_o) \tag{2.7}$$

2.4 Taxonomy of classical clustering techniques

Once we have identified the types of our data and the measures that can be applied to them, we need to choose a specific clustering algorithm to produce groups of similar items. However, different approaches to clustering data can be described. Indeed, a large number of clustering algorithms exist in the literature. It is difficult to provide a crisp categorization of them since these categories may overlap. Below, we outline

the most important traditional clustering categories briefly. Indeed, we distinguish partitional versus hierarchical and exclusive versus overlapping clustering.

2.4.1 Partitional versus Hierarchical

The most discussed distinction among different types of clustering is the set of unnested clusters or nested or in more known terminology partitional or hierarchical.

Partitioning methods

A partitional clustering is simply a division of the data set into non overlapping subsets or clusters such that each data object is in exactly one group.
Given a database of n objects and k clusters to form, a partitioning algorithm organizes the objects into k partitions ($k \leq n$), where each partition represents a cluster. The clusters are formed to optimize an objective partitioning criterion, often called a similarity function or a cost function, so that objects within one cluster are similar, whereas objects of different clusters are dissimilar in terms of the attribute values.
One of the most well-known and commonly used partitioning methods is the K-means method and its extensions. So, within K-means method (MacQueen, 1967) context, each cluster is represented by the mean value of its objects. For the K-modes method extension (Huang, 1998), the categorical data sets are handled, while K-prototypes (Huang, 1997) version deals with mixed data. The clusters of the K-medoids (Kaufman and Rousseeuw, 1990) are represented by one of the objects located near the center of the cluster. Note that there are other variants of these mentioned methods.

Hierarchical methods

When clusters have sub-clusters, we obtain hierarchical clustering represented by a set of nested clusters. It consists in grouping data into a tree of clusters. Hierarchical clustering methods can be further classified into agglomerative and divisive hierarchical clustering, depending on whether the hierarchical decomposition is formed in a bottom-up or top-down fashion.
Agglomerative hierarchical clustering, as bottom-up strategy, starts by placing each object in its own cluster. Then, it merges these elementary clusters into larger ones, until all of the objects are in a single cluster or until certain termination conditions are satisfied.
Divisive hierarchical clustering, as top-down strategy, does the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster. It subdivides the cluster into smaller groups, until each object forms a cluster on its own or until it satisfies certain termination conditions, such as a desired number of clusters is ob-

tained or the distance between the two closest clusters is above a certain threshold distance.
Among the methods of this category, we cite AGNES (Kaufman and Rousseeuw, 1990) as agglomerative technique and DIANA (Kaufman and Rousseeuw, 1990) as divisive one.

2.4.2 Exclusive versus Overlapping

By the following, we highlight another clustering categorization based on the notion of membership degree of objects to groups (soft or hard way).

Exclusive or hard methods

Clustering techniques which assign each object to a single cluster are exclusive approaches. These approaches are also called hard or crisp clustering techniques (Jain et al., 1999). A such clustering algorithm assigns each pattern to one and only one cluster.

Overlapping or soft methods

There are many situations in which a point cloud be placed simultanouesly in more than one cluster and this is addressed by non-exclusive clustering approaches. For instance, a person at university can be both a student and an employee. As an overlapping clustering, the fuzzy clustering allows the object belonging to every cluster with a membership weight that is between 0 and 1. It is the same for the probabilistic clustering techniques which compute the probability with which each instance belongs to each group. Note that other frameworks permit to handle this same hypothesis namely overlapping clusters such as possibilistic and credal clustering.

Remark:
Other clustering techniques can be listed. Firstly, density-based algorithms which typically regard clusters as dense regions of objects in the data space that are separated by regions of low density. Let us mention as example DBSCAN (Density-Based Spatial Clustering of Applications with Noise) (Ester et al., 1996).
Secondly, the grid-based clustering algorithms which have been presented for spatial data. Indeed, many grid-based methods have been proposed, such as STING (Statistical Information Grid Approach) (Wang et al., 1997), CLIQUE (Agrawal et al., 1998).
Thirdly, for the family of model-based clustering algorithms, one uses certain models for clusters and tries to optimize the fit between the data and these mathematical

models. A typical technique in this category is Expectation-Maximization algorithm (Bradley et al., 1999).

As mentioned before, thousands of clustering algorithms have been proposed in the literature. Nonetheless, we will briefly review the K-modes method since it is the adopted one by this thesis work.

2.5 Origins of the standard K-modes method

As is well known, K-means method (MacQueen, 1967) has been a very popular technique for partitioning large data sets with numerical attributes. However, data mining applications frequently involve many data sets that also consist of categorical attributes. One approach (Ralambondrainy, 1995) was proposed to using the K-means algorithm to cluster categorical data. This approach consists in converting multiple category attributes into binary attributes (using 0 and 1 to represent either a category absent or present) and treat the binary attributes as numeric in the K-means algorithm. If it is used in data mining, this approach needs to handle a large number of binary attributes because categorical attributes in data sets often have hundreds or thousands of categories. This will increase both computational and space costs of the K-means algorithm. Another drawback is that the cluster means, given by real values between 0 and 1, not really reflect the characteristics of the clusters.

Indeed, by applying K-means method to categorical data, two main problems are encountered, namely, the construction of clusters' centers and the definition of dissimilarity between objects and clusters' centers. So, the standard K-modes method (Huang, 1998) (SKM) was proposed to tackle the problem of clustering large categorical data sets in data mining.

2.5.1 Basic parameters

K-modes algorithm (Huang, 1998) was proposed as an extension of the K-means one (MacQueen, 1967), since the standard version of this last one deals only with numerical data sets. By using the K-means paradigm, this categorical clustering technique considers a simple matching dissimilarity measure, modes instead of means for cluster representatives, and a frequency-based approach to update modes during the clustering process as discussed in below.

Cluster's mode

Given a cluster $C_l = \{X_1, ...X_{n_l}\}$ of n_l categorical objects, with $X_i = (x_{i,1}, .., x_{i,s})$, $1 \leq i \leq n_l$. Its mode $Q_l = (q_1, .., q_s)$ is defined by assigning to q_j the category most frequently encountered in $\{x_{1,j}, ..., x_{n_l,j}\}$, where $1 \leq j \leq s$ and s is the number of attributes.

The dissimilarity measure

We assume that the set of objects to be clustered is stored in data set T defined by a set of categorical attributes $A_1, A_2, ..., A_s$.
The distance between any two categorical objects $X_{i_1}, X_{i_2} \in T$, with $X_{i_1} = (x_{i_1,1}, ..., x_{i_1,s})$ and $X_{i_2} = (x_{i_2,1}, ..., x_{i_2,s})$, can be defined as follows, where i_1, $i_2 \in \{1, .., n\}$:

$$d(X_{i_1}, X_{i_2}) = \sum_{j=1}^{s} \delta(x_{i_1,j}, x_{i_2,j}), \tag{2.8}$$

where:

$$\delta(x_{i_1,j}, x_{i_2,j}) = \begin{cases} 0 & if\ x_{i_1,j} = x_{i_2,j} \\ 1 & if\ x_{i_1,j} \neq x_{i_2,j} \end{cases} \tag{2.9}$$

$\delta(x_{i_1,j}, x_{i_2,j})$ is considered as the simple matching respectively to one attribute A_j. For these two objects, if all attributes have the same values, this distance measure d will be equal to zero. However, when these two objects are totally distinct, it means that all attributes' values are different, so this measure will be equal to the number of attributes s. In other word, $0 \leq d \leq s$.
It is easy to verify that the function d defines a metric space on the set of categorical objects.

When these parameters are used, the K-modes optimization problem for partitioning a set of n objects described by s categorical attributes into k clusters becomes:

$$Minimize\ P(W, Q) = \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{s} w_{i,l} \delta(x_{i,j}, q_{l,j}) \tag{2.10}$$

subject to:

$$\sum_{l=1}^{k} w_{i,l} = 1,\ 1 \leq i \leq n,\ 1 \leq l \leq k,\ and\ w_{i,l} \in \{0, 1\}$$

where W is an $n \times k$ partition matrix, $w_{i,l}$ is the degree of membership of the object X_i in the cluster C_l (using 1 and 0 to represent either the object X_i is an element of

the cluster C_l or not), and $Q = \{Q_1, Q_2, ..., Q_k\}$.

2.5.2 SKM algorithm

K-modes method is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters fixed a priori.

The main idea is to define k modes, one for each cluster. These modes should be placed in a cunning way because of different location causes different results. However, the standard way is to chose them randomly. The next step is to take each point belonging to the considered data set and associate it to the nearest mode using the simple matching dissimilarity measure. When no point is pending, the first step is completed and an early groupage is done. Thus, we need to recalculate k new modes of the clusters resulting from the previous step based on frequency-based method. After getting these k new modes, a new binding has to be done between the same data set points and the nearest new mode. A loop has been generated. As a result of this loop, we may notice that the K modes change their location step by step until no more changes are done. In other words, modes do not move any more.

This algorithm has as an input a predefined number of clusters, that is the k from its name, besides of the data set to cluster. K-modes algorithm is a simple, iterative procedure, in which a crucial concept is the one of mode. This latter is an artificial point in the space of records which represents all objects of the particular group. The coordinates of this point are the most frequently of attribute values of all examples that belong to this cluster.

The algorithm is composed of the following steps:

1. Select k initial modes, randomly, one for each cluster.
2. Allocate an object to the cluster whose mode is the nearest to it according to the simple matching dissimilarity measure (Equation 2.8). Update the mode of the cluster after each allocation.
3. Once all objects have been allocated to clusters, retest the dissimilarity of objects against the current modes. If an object is found such that its nearest mode belongs to another cluster rather than its current one, reallocate the object to that cluster and update the modes of both clusters.
4. Repeat 3 until no object has changed clusters after a full cycle test of the whole data set.

Although, it can be proved that the procedure will always terminate, the K-modes algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is significantly sensitive to the randomly selected initial cluster modes. Thus, it can be run multiple times to reduce this effect.
The initial mode selection, of step 1, in (Huang, 1998) is ensured via the selection of the first k distinct records or also by considering the frequency concepts. Another naive way to do this is the random selection of k distinct instances as initial modes. This is the adopted approach by (Huang and Ng, 1999).

Advantages:

- Simplicity and understandability.
- The convergence of the algorithm in a finite number of iterations.

Drawbacks:

- The impact of the initial values of the modes. For example, when the random modes initialization is performed, the standard solution is to try a number of different starting points.
- The dependency of the results on the value of k. Unfortunately, there is no general theoretical solution to find the optimal number of clusters for any given data set. A simple approach is to compare the results of multiple runs with different k values and choose the best one according to a given criterion such that the clustering cost function.
- The algorithm unstability due to the mode non-uniqueness problem. There are two extensions of the K-modes algorithm (He, 2006; San et al., 2004) which aim at eliminating the mentioned drawback in the K-modes method by introducing new definitions of cluster modes.

In the following, the clustering under uncertainty issue will be detailed. Indeed, the uncertainty is among the major drawbacks that clustering techniques may encounter.

2.6 Clustering under uncertainty

Actually, there has been a significant research interest in uncertainty management in recent years. This uncertainty has been broadly classified into two types. The first type concerns the object position in a multidimensional space known also as object assignment (soft clustering). The second type regards attribute level or the object

characteristics (uncertain data). Clearly, an hybrid generalization can be defined to deal with both input uncertain data and output uncertain cluster solution.
However, this problem makes most of the standard methods inappropriate to ensure their clustering task in such context. In order to overcome this drawback, the idea is to combine clustering methods with theories managing uncertainty. Among these theories there are probability theory, fuzzy set theory, possibilitic theory, the belief function theory (Ben-Hariz et al., 2006; Denoeux and Masson, 2004; Masson and Denoeux, 2008; Quost and Denoeux, 2010; Ammar et al., 2012), etc.

For instance, fuzzy clustering algorithms for which the most used one is the Fuzzy C-means (FCM) (Bezdek, 1981) generate a soft partition namely a fuzzy partition. The term possibilistic partition, which has been introduced in (Krishnapuram and Keller, 1993), is a more generalized form of the first one by relaxing a certain constraint. Thus, each point is assigned to at least one of the fuzzy subsets with a membership greater than zero. Indeed, this fuzzy set theory was also combined with the K-modes method (Huang and Ng, 1999) to represent and managed crisp cluster partition. The concepts of belief clustering was also recently proposed. Furthermore, two kinds of belief clustering can be distinguished namely belief clustering which is interested to produce soft cluster partition (Denoeux and Masson, 2004; Masson and Denoeux, 2008) and the second form which was developed to deal with uncertain attribute values characterizing the considered data sets (Ben-Hariz et al., 2006) and generating a crisp partition.

Concerning the uncertain class assignment, another approach was detailed in (Peters et al., 2010). The authors proposed two adapted cluster algorithms that employ soft computing approaches, in particular based on fuzzy and rough concepts, and provide a comparison regarding their capabilities to capture uncertainties in such dynamic environment. Another recently work (Bordogna et al., 2006) develops a dynamic hierarchical fuzzy clustering algorithm where the fuzzy clustering algorithm, based on a generalization of the fuzzy C-means algorithm, is iteratively applied to each hierarchical level to identify clusters of the higher level. This same problem of dynamic clustering has been considered also in (Beringer and Hüllermeier, 2007). In this latter, the problem is to cluster the elements of certain data sets within fuzziness. It is respectively to the cluster number learning.

We focus on the uncertain versions of K-Family works such as K-means, K-modes, K-prototypes, etc. Thus, different extensions of these methods have been developed in the literature whether for the first or the second uncertainty form. To ensure this, many combinations of the standard approaches with uncertainty theories such

as the probability, possibility theories and the belief function theory. Let us mention as exemplars fuzzy C-means (Bezdek, 1981), possibilistic K-modes (Ammar et al., 2012), evidential clustering works of Deneoux (Denoeux and Masson, 2002; Denoeux and Masson, 2004). All these research works generate soft cluster collection. On the other hand, belief clustering, as defined by (Ben-Hariz et al., 2006), and possibilistic one developed in (Ammar and Elouedi, 2011), both deal with uncertain attribute values and generates hard collections with non-overlapping clusters.

If uncertainty modeling is already important in static clustering, this becomes even more important in dynamic clustering where several classification elements can change over time. These changes produce uncertainty and that is where uncertainty modeling in dynamic clustering comes into play. That is why, we will expose, thereafter, the dynamic or online clustering concept.

2.7 Online clustering

The problem of pattern maintenance, consisting in updating the data, is also a fundamental task in data management process. Most existing clustering approaches (Jain and Dubes, 1988) suppose that databases are static. Thus, for their updates, the scanning of both old and new information is required. Such algorithms are known as off-line. The proposition of dynamic mining techniques for addition or deletion of any information will be more efficient and this is ensured by taking into account model evolution over time. As examples, we mention (Aranganayagi and Thangavel, 2010; Su et al., 2009; Truta and Campan, 2007).

Dynamic clustering is an important research topic in data mining. Recently, many methods have been proposed in the literature to cluster dynamic data sets.
We define a dynamic data set as a set of elements whose parameters change over time. Indeed, during the recent years, non-stationary data sets have attracted considerable attention in different fields of computer science. The respective extended techniques are able to update cluster partition when any change is detected. This latter concerns the objects, the attributes as well as the cluster partition. Among these propositions, we mention (Aranganayagi and Thangavel, 2010; Truta and Campan, 2007). In (Aranganayagi and Thangavel, 2010), the clusters are created by adding incrementally the data set objects. Indeed, by varying one threshold value, the proposed approach can select the best number of clusters from the obtained result. This algorithm is similar to M-Squeezer (Aranganayagi and Thangavel, 2009) which reads the object one by one and clusters the object based on the threshold value.
Furthermore, in (Truta and Campan, 2007), incremental as well as decremental data

set procedures are provided to cluster dynamic objects sets. These algorithms incrementally manage insert/delete/update data set modifications for the maintenance of k-anonymized versions of large evolving data sets. An online recursive clustering algorithm based on incremental and decremental Support Vector Machine (SVM) is developed to learn evolving clusters from non-stationary data (Boukharouba and Lecoeuche, 2008). It is able to achieve an efficient multi-class clustering in a non-stationary environment based on a new similarity measure and different procedures such as creation, adaptation, fusion and elimination.
All these dynamic classifiers can provide optimal updated models of data within a certain framework.

The above literature review presents some developed methods to deal with uncertainty problem (Section 2.6) or to operate in dynamic environment (Section 2.7). Generally, these two issues are handled independently. However, some works consider them together, but only one uncertainty form is managed via these proposals.
Therefore, these proposed techniques cannot provide a general framework to deal with both aspects. Nonetheless, for the works proposed in (Beringer and Hüllermeier, 2007; Bordogna et al., 2006; Peters et al., 2010), the both aspects are handled but the attribute values are always considered as known with certainty and the treated uncertainty form is respectively to the objects' membership. Moreover, effective dynamic clustering algorithms should be designed to deal with such non standard context.

2.8 Clustering: Additional issues

The number of problems associated with clustering techniques is legion. Common clustering shortcomings can be listed as follows. Thus, a lot of clustering solutions has been emerged during the last years which try to solve such drawbacks.
By the following, we will detail some well-known clustering drawbacks and expose, briefly, some related works.

- Missing data:
  Besides complete data sets for which all objects have known attribute values, there is another category of data collection, in real world applications, namely incomplete data sets if at least one datum has missing value(s) for its attribute(s). Common solutions either fill in the missing values in a pre-processing step (imputation) or ignore the missing data (marginalization) as identified in (Green et al., 2001). Nonetheless, imputed data cannot be considered as reliable as the actually observed data as stated by (Troyanskaya et al., 2001). Despite this, the data imputation techniques remain common.

One developed approach is to replace all missing values with the observed mean for that feature known as the 'row average' method (Bingham and Heikki, 2001). A more sophisticated technique consists to infer the value of the missing feature based on that item's observed features and its similarity to other known items in the data set (Troyanskaya et al., 2001).
An important, general statistical method for handling incomplete data based on the modified expectation-maximization (EM) algorithm was developed in (Ghahramani and Jordan, 1994). In essence, this approach iteratively produces maximum likelihood estimates based on a two-part iteration (E-step and M-step).
Most works based on marginalization principle has focused on supervised methods such as neural networks. In contrast, (Wagstaff, 2004) handles missing values even no labeled training data.
The developed methods to deal with these missings can be also classified, differently, into two groups namely pre-replacing methods, which replace missing values before the data-mining process and embedded methods, which deal with missing values during the data-mining process.

- Outliers detection:
  The existence of possible outliers, which is often regarded as noise, can affect the form of the resulting clusters and distort the clustering algorithm. Then, their detection and removing must be performed.
  Indeed, an outlier can be defined as an observation that is numerically distant from the rest of the considered data collection. It is a datum that deviates markedly from the other ones. Note that a datum is considered as noisy when is a meaningless one. This latter includes any data that cannot be understood. Outliers detection is an extremely important problem with a direct application in a wide variety of domains. The outliers detection techniques can be divided into four main categories: distance-based, distribution-based, density-based and clustering-based approaches.
  Some approaches remove outliers before clustering, while others identify outliers in a post-processing step. Note that where outliers are presented the resulting cluster centroids may not be enough representative. It is often useful to discover outliers and eliminate them beforehand.
  Clustering algorithms like DBSCAN and ROCK (Ester et al., 1996; Guha et al., 1999) can also handle outliers, but their main concern is clustering data set, not outliers detection.

- Sensitivity to initial configuration:
  At the beginning of the clustering process, we have to select an appropriate

initial configuration via:

1. The determination of cluster number:
   In most real life clustering situations, an applied researcher is faced with the dilemma of selecting the number of final clusters. Automatically determining the number of clusters has been one of the most difficult problems in data clustering.
   Non hierarchical procedures usually require the user to specify this parameter before any clustering is accomplished, while the hierarchical methods produce a series of solutions ranging from n clusters to a solution with only one cluster. As such, numerous procedures for determining the number of clusters in a data set have been proposed. When applied to the results of hierarchical clustering methods, these techniques are sometimes referred to as stopping rules. Nonetheless, for partitional methods, they serve as input. Usually, clustering algorithms are run with different cluster numbers. The best value is then chosen based on one predefined criterion.
2. The selection of the initial partition:
   In clustering algorithms, choosing a subset of representative examples is very important task. Such exemplars can be found by randomly choosing an initial subset of data objects and then iteratively refining it, but this works well only if that initial choice is close to a good solution. Note that different initial points often lead to considerable distinct clustering results. More details, respectively to this point, will be given in the following Chapter.

2.9 Conclusion

This chapter presented the basic concepts of clustering analysis. Thus, data types, proximity measures and the taxonomy of classical clustering techniques were detailed briefly. The most known K-modes method as partitional technique handling categorical data sets was exposed since it is the used clustering approach through this thesis work. Then, we have detailed both clustering under uncertainty and online clustering. Thereafter, we have given some additional issues.

K-modes method has some drawbacks where uncertainty pervades data sets. In the next Chapter, we will present belief K-modes method (BKM) as a possible solution to deal with uncertain data sets within belief function theory framework. An improvement of this latter will also be proposed via a new initialization technique. Consequently, the clustering initialization problem will also be addressed, besides the uncertainty one, by the following chapter.

3

Improving Belief Clustering via a new initialization method

3.1 Introduction

In a part of the previous chapter, we have presented the uncertainty problem in context of clustering task. To deal with this drawback, by the following, we will detail the belief K-modes method BKM as a possible solution to handle data sets with uncertain attribute values. It is based on the standard K-modes approach which has already been presented in the previous Chapter.

Both certain and uncertain versions suffer from the random cluster centers initialization problem. In this Chapter, we propose, secondly, to solve this randomness problem leading to a local optimum, a new initialization method. Called max-dissimilarity based initialization method, our proposal tries to converge as possible to the global optimum since the result depends on this initial partition of k clusters.

Indeed, in the first part of this Chapter, we are interested especially in the belief counter part of the K-modes method. We focus on the belief data set structure, the BKM principle parameters and the corresponding algorithm.
The second and third parts of this chapter deal with the initialization problem. Thus, we expose, briefly, a review of related works. The objective is to distinguish the major proposed cluster initialization methods in the literature, respectively to the categorical data, and to present their differences. Then, our proposal is detailed. After its principle and its algorithm exposition, the experimental analysis is reported. This is done in order to assess its performance and to show its comparative results to the standard initialization methods of literature references. Note that this developed initialization technique will be preserved within dynamic context in the remaining of this manuscript.

3.2 Belief K-modes method

The standard version of the K-modes method (SKM) (Huang, 1998) gives good results in a context in which everything is known with certainty. However, the reality is connected to uncertainty and imprecision by nature. Such uncertainty may badly affect the classification performance. Indeed, a good classifier must be able to predict the object class value even when information concerning the object is imperfect. So, the K-modes is inadequate and badly adapted to ensure its role of classification in an environment characterized by a lot of uncertainty and imprecision. That is why searches are oriented to the improvement and the extension of this method, in order to adapt it to such kind of environment. Thus, the idea is to combine theories managing uncertainty and imprecision with the K-modes method. Amongst these theories, there are probability theory, fuzzy set theory, belief function theory and possibility theory.

Hence, this adaptation of K-modes method to an uncertain environment has led to new approaches for instance the fuzzy K-modes method (Huang and Ng, 1999). For this approach, one object does not belong exclusively to a well defined cluster. In fact, it may belong to several clusters with different membership degrees. Indeed, the uncertainty may also appear in the attribute values of the training set used to ensure the cluster construction phase instead of the cluster membership. Therefore, a belief K-modes method (BKM) (Ben-Hariz et al., 2006) was developed, as a new clustering technique based on the standard K-modes technique and using the belief function theory concepts in order to deal with this uncertainty form. Building clusters, for this belief version of K-modes method, needs the definition of appropriate fundamental parameters namely cluster modes computation and the dissimilarity measure. Recall that these parameters must take into account the uncertainty pervading the attribute values of the data set instances.

In the following part of this report, we present the two major parameters of this belief K-modes extension to advance in what follows its algorithm. Moreover, we start by presenting the belief structure of the considered non standard training sets.

3.2.1 Belief data set structure

The structure of the considered training set is different from the standard one. Contrary to the certain training set which includes only precise information, here, we deal with n objects where each of their s attributes may be represented by a *bba* expressing beliefs on its values.

The corresponding *bba* of an attribute A_j, where $1 \leq j \leq s$, is then given by $m_j = \{m(c_j)|c_j \in 2^{\Theta_j}\}$, with $m(c_j)$ is the bbm given to the category c_j. Indeed, each attribute is represented via one conjunction of all possible values and their corresponding masses. Thus, each $x_{i,j} = \{(c_j, m_i(c_j))|c_j \in 2^{\Theta_j}\}$, with $m_i(c_j)$ represents the bbm given to $c_j \in 2^{\Theta_j}$ relatively to object X_i.
This uncertain training set offers a more generalized framework than the standard one.
Based on the two following parameters, the BKM algorithm can be applied and it has the same skeleton as SKM method (Ben-Hariz et al., 2006).

3.2.2 BKM parameters

Cluster mode computation

Given one cluster $C_l = \{X_1, ..., X_{n_l}\}$ of n_l objects, for $1 \leq l \leq k$. An intuitive definition of one strategy of computing the cluster modes within the belief function theory context could be the conjunctive rule of combination (Smets, 1993) generally used as an aggregation operator in such framework. This operator is particularly suitable when distinct sources provide pieces of evidence respectively to the same object. However, in our case, the induced pieces of evidence are related to different object attributes. So, the conjunctive (even disjunctive) rule is not appropriate. On the other hand, the mean operator permits combining bba's respectively to each attribute provided by all n_l objects of one cluster as well as satisfying the commutativity and the idempotency properties (Murphy, 2000). Then, the idea was to apply this operator to this uncertain context (Ben-Hariz et al., 2006).
Thus, the mode of the cluster C_l can be defined by the following belief mode $Q_l = (q_{l,1}, .., q_{l,j}, .., q_{l,s})$, such that:

$$q_{l,j} = \{(c_j, m_l(c_j))|c_j \in 2^{\Theta_j}\} \tag{3.1}$$

Where $m_l(c_j)$ is the relative bbm of attribute value c_j within C_l which is defined as follows:

$$m_l(c_j) = \frac{\sum_{i=1}^{n_l} m_i(c_j)}{|C_l|} \tag{3.2}$$

Where $n_l = |C_l|$ is the number of objects in C_l, while $m_l(c_j)$ expresses the part of belief about the category c_j of the attribute A_j corresponding to this cluster mode. It is the average of all masses provided by the objects in this cluster corresponding to this attribute category. Note that the associative property is ensured via the arithmetic addition of all masses.

Dissimilarity measure

The dissimilarity measure has to take into account the new data set structure, where a *bba* is defined for each attribute per object. It should be able to compute the distance between any object and each cluster mode, respectively to all considered attributes, within such context (bba's representation). This is done to decide their assignments.

Several works are concentrated on the definition of a distance between two basic belief assignments within belief function theory framework. So, a very important number of measures were developed which can be classified into two kinds which are detailed below.

The first category of distance measures within belief function context are those based on the pignistic transformation. For these distances such as (Bauer, 1993; Elouedi et al., 2004; Tessem, 1997; Zouhal and Denoeux, 1998), one unavoidable step is the pignistic transformation of the bba's (transformation from the power set to the frame of discernment). This kind of distances may lose information given by the initial bba's. Besides, we can obtain the same pignistic probabilities by applying the pignistic transformation on two different bba's . So, the distance between the two obtained results does not reflect the actual similarity between the starting bba's values.

The second kind of belief distance measures are those which are applied directly to bba's and not to the pignistic probabilities. These measures are defined on the power set. Let us mention (Fixen and Mahler, 1997; Jousselme et al., 2001) as examples. The first one, developed by Fixen and Mahler, is a pseudo-metric, since the condition of nondegeneracy of one distance metric is not respected. Thus, in BKM framework, the idea was to adapt the belief distance metric defined by (Jousselme et al., 2001) to this uncertain clustering context.

Recall that this distance measure is defined directly on bba's and satisfies more properties than the other exposed measures as detailed in (Ben-Hariz et al., 2006). It can be expressed as follows:

$$D(X_i, Q_l) = \sum_{j=1}^{s} d(m_{i,j}, m_{l,j}) \tag{3.3}$$

where $m_{i,j}$ and $m_{l,j}$ are the relative bba's of the attribute A_j provided by respectively the object X_i and the mode Q_l.
The component d takes into account both the bba's distributions provided by the object X_i and the cluster mode Q_l and one similarity matrix D. This latter is based on the cardinalities of the intersection and the union of the different subsets of the

attribute frame.
Consider, now, m_i and m_l instead of $m_{i,j}$ and $m_{l,j}$, to simplify the following formulas. These two bba's, of the considered object and cluster mode, are defined on the same frame of discernment Θ_j. Indeed, their distance is computed as follows:

$$d(m_i, m_l) = \sqrt{\frac{1}{2}(\vec{m_i} - \vec{m_l})D(\vec{m_i} - \vec{m_l})} \tag{3.4}$$

Another way to write it is:

$$d(m_i, m_l) = \sqrt{\frac{1}{2}(\|\vec{m_i}\|^2 + \|\vec{m_l}\|^2 - 2 < \vec{m_i}, \vec{m_l} >)} \tag{3.5}$$

where $< \vec{m_i}, \vec{m_l} >$ is the scaler product defined by:

$$< \vec{m_i}, \vec{m_l} >= \sum_{w=1}^{2^{\Theta_j}} \sum_{z=1}^{2^{\Theta_j}} m_i(B_w) m_l(B_z) \frac{|\mathrm{B_w} \cap \mathrm{B_z}|}{|\mathrm{B_w} \cup \mathrm{B_z}|} \tag{3.6}$$

with $B_w, B_z \in 2^{\Theta_j}$. This latter is known as power set of this attribute and contains all possible subsets of Θ, and $\| \vec{m} \|^2$ is then the square norm of $\vec{m}$: $\|\vec{m}\|^2 = < \vec{m}, \vec{m} >$ This scalar product is based on the bba's distributions (m_i and m_l) and the elements of one similarity matrix D, whose elements are defined as follows:
$D(B_w, B_z) = \frac{|\mathrm{B_w} \cap \mathrm{B_z}|}{|\mathrm{B_w} \cup \mathrm{B_z}|}$.

3.2.3 BKM algorithm

The BKM algorithm has the same skeleton as SKM one. The belief distance measure (Equation 3.3) must be used instead of the simple matching one (Equation 2.10). Its different steps may be summarized by the following pseudo algorithm:

1. Choose randomly k initial items as initial cluster modes.
2. Partition, initially, the remaining objects of the training set in these k nonempty subsets based on the belief distance measure concept (Equation 3.3).
3. Compute seed points as the cluster modes of the current partition using the averaging rule of combination (Equation 3.2).
4. Assign each object to the cluster with the nearest seed point after computing its distance measures respectively to all cluster modes defined by Equation 3.3.
5. Go back to step 3, stop when no more new assignment. In other word, all objects have no changed clusters. Another stopping criterion may be considered namely the maximum number of iterations.

Clustering accuracy of BKM algorithm depends upon the choice of initial data points (modes) which affect the clustering results as all K-methods (K-means, K-modes, K-medoids, K-prototypes, etc).
Indeed, the BKM method, like the standard version of the K-modes method, produces locally optimal solutions that are dependent on the initial modes. These latter are selected, arbitrary, from the data set to cluster. Since, the algorithm is significantly sensitive to this choice, the BKM algorithm is run multiple times to reduce this effect. In response to this problem, we have developed an appropriate selection method (Ben-Hariz et al., 2008) that will be detailed below (see Section 3.4).

This problem initialization must be studied, by the following, in order to show later the merit of the developed initialization method under belief function framework.

3.3 Clustering initialization

3.3.1 Problem definition

Clustering techniques, more precisely the K-methods family, require initial cluster centers to start and often end up with different clustering results respectively. Therefore, this family of methods is very sensitive to this parameter initialization. Usually, these approaches are run with different initial guesses of cluster representatives, and the attained results are compared in order to determine the best clustering outputs based on the objective function values. Besides, the clustering validation techniques can be used to decide which resulting partition will be selected. To address this issue, some approaches have been proposed by using a better initial seed value selection from the beginning. Generally, the initialization techniques used with numerical data cannot be applicable to categorical one. Considering that our thesis work deals only with these symbolic data, we will expose, by the following, a brief review of the existing initialization methods respectively to the K-modes technique. Indeed to date, few researches are developed for categorical data. The most important ones will be summarized below.

3.3.2 Existing initialization approaches

Although its efficiency, the K-modes algorithm is still sensitive to the initial points. Indeed, a better initial point selection would improve the reliability and accuracy of clustering results.

In (He et al., 2004), a review of some initialization of cluster refinement algorithms is made. This work categorized them into three major families, namely random

sampling methods, distance optimization methods and density estimation methods. Nonetheless, this referenced literature corresponds to clustering numerical databases, while the handled data by this thesis are categorical databases.

Therefore, we will detail, only, the initialization approaches which have been developed for clustering categorical data sets. Generally, the K-modes method is the basis of these approaches.

The most classical initialization method is the random seed. It consists in random selection of k instances (seed points), and assignment of each of the other instances to the cluster with the nearest seed point. This manner is adopted by several clustering works whatever numeric or categorical methods such as in (Huang and Ng, 1999).

Indeed, Huang (Huang, 1998) proposes two more sophisticated approaches for initializing the clusters for K-modes algorithm. In the first method, the first k distinct data objects are chosen as initial k modes, whereas the second method computes the frequencies of all categories for all attributes and assigns the most frequent categories equally to the initial k modes.

Work in (Sun et al., 2002) presents an experimental study on applying Bradley et al.'s iterative initial-point refinement algorithm (Bradley and Fayyad, 1998) to the K-modes clustering to improve its accuracy.

Another developed approach by (Khan and Kant, 2007) generates, initially, a Mode-Pool by running some clusterings to produce different partitions and then extracts initial modes from this pool based on the frequency concept.

Two farthest point heuristics for computing initial cluster centers, always for K-modes algorithm, were presented in (He, 2006). The first heuristic chooses an arbitrary item as the first mode and then, the other ones are selected based on distance concept. The second heuristic uses a deterministic method, based on a scoring function, to select the first mode. For the remaining ones, the same procedure is applied. This heuristic does not explain how to choose a point when several data objects have same scores, and if it randomly break ties, then fixed centers cannot be guaranteed.

A density based method to compute the k initial modes, which has quadratic complexity was also developed (Wu et al., 2007).
Cao et al. (Cao et al., 2009) present an initialization method that considers distance between objects and the density of the objects. A major drawback of this method is

that it has quadratic complexity.

In (Bai et al., 2012) a novel initialization method has been developed which integrates the distance and the density together to select initial cluster centers and overcomes some shortcomings of the previous existing methods. Withal, it requires slightly more computational times than (Cao et al., 2009).

We can conclude that most of literature initialization methods suffer from either one or more of these three drawbacks namely the time complexity, the randomness aspect and the impact of the presentation order of data objects.
In summary, there are no universally accepted methods for obtaining initial cluster centers.

All advanced initialization clustering approaches above have been developed for certain data sets with known and precise attribute values. Moreover, they use the frequency of categories as core concept to ensure their task.
However, for our framework the bba's values to represent uncertain attribute values, are the considered ones instead of the attribute categories. Hence, it will be very interesting to propose a new initialization method which tries to overcome the mentioned shortcomings of the existing approaches as possible. This proposal must be able to ensure its task in such uncertain context within belief function framework.
Note that this method will be able to choose the initial representatives for both cases certain as well uncertain via the belief function theory concepts.

By the following, we will detail the proposed initialization strategy to use within BKM context.

3.4 Max-dissimilarity based initialization method under belief function framework

3.4.1 Principle

As mentioned above, clustering accuracy of BKM algorithm for categorical uncertain data depends upon the choice of initial data points (modes) which affect the clustering results.

The BKM method, like the standard version of the K-modes method, produces locally optimal solutions that are dependent on the initial modes. It consists in, arbitrary, selection of k initial cluster modes from the data set to cluster. Since, the

algorithm is significantly sensitive to this choice, the BKM algorithm is run multiple times to reduce this effect.

As known, the aim of clustering is to group a set of objects into classes of similar objects by decreasing the dissimilarity within the same cluster and increasing this measure between distinct clusters. So, in this section, we propose a new method for the selection of the k initial modes to improve our initial approach. Our method consists in choosing k objects from the data set which provide the maximum of dissimilarity measure between them. It means that the chosen k initial modes are the k objects which are the most far ones. The modes should be placed in a cunning way because different locations cause different results. However, to place them as much as possible far away from each other is the better manner to have the initial k modes. The purpose of this method is to better select initial modes, which can lead to better clustering results.

3.4.2 Algorithm

The initial version of our BKM method starts with k distinct objects, arbitrary chosen, as the initial modes. This is the most popular way to start the K-modes algorithms. In our case, and in order to overcome this drawback, a new method was proposed, instead of this standard one.

The main idea is to select k objects as k cluster modes based on the dissimilarity measure notion, in deterministic way, rather than the random choice.
To this end, we have to compute the dissimilarity matrix of n objects (n*n matrix) and assign firstly the two objects which provide the highest dissimilarity measure to the two first cluster modes. When k is equal to two, the procedure is terminated and the selection of our initial modes is done. Otherwise, for the other objects, we need to compute their distance sum respectively to the already fixed modes and associate to the following mode, the object with highest dissimilarity measure (the most far from the already chosen modes). Thereafter, a new iteration has to be done for the remaining objects. We iterate until the selection of the k initial modes has been ensured.
This algorithm has as inputs:

- The number of modes k.
- The belief data set to cluster UT.
- The dissimilarity matrix D, $n \times n$, where $D_{i,u}$ is the distance between the object X_i and the object X_u based on the distance measure defined by Equation 3.3

with $i, u \in \{1, .., n\}$.

Indeed, the algorithm is composed of the following steps:

(1) Allocate the two objects X_i and X_u which provide the highest dissimilarity measure ($D_{i,u} = max(D)$) to the two first clusters' modes.

(2) For the other objects (n - $number\ of\ chosen\ cluster\ modes$), compute their distances' sum respectively to the already chosen modes

(3) Assign the object with the highest distance measure to the following mode.

(4) Repeat 2 and 3 till the k initial modes have been chosen.

Note that this initialization approach, developed within belief function framework, may also be used for the certain case. Besides, this approach has been proposed, firstly, for static belief clustering. In the following, it will be used, within dynamic context in this thesis work, and consequently applied to all our dynamic proposals.

3.4.3 Illustrative example

Let us illustrate our method by a simple example. Assume that a firm wants to group its staff into two collections by taking into account their attributes.

Let UT be an uncertain training set (see Table 3.1) composed of seven instances characterized by three categorical attributes:

- Qualification with possible values {A, B, C}.
- Income with possible values {High , Low, Average}.
- Department with possible values {Finance, Accounts, Marketing}.

For each attribute A_j, with $j \in \{Qualification, Income, Department\}$, for an object X_i, with $i \in \{1, 7\}$, belonging to the training set UT, we assign a bba $m_{i,j}$ expressing beliefs on its assigned attributes values. To simplify the following notations, we will use 1, 2 and 3 instead of Qualification, Income and Department as attributes. Indeed, these latter are defined respectively on:

$\Theta_1 = \{A, B, C\}$
$\Theta_2 = \{High, Low, Average\}$
$\Theta_3 = \{Finance, Accounts, Marketing\}$.

If we consider that only the department attribute is known with uncertainty. The structure of the data set UT is defined by the the following table.

Table 3.1: Belief data set UT

Objects	Qualification	Income	Department
X_1	A	High	$m_{1,3}$
X_2	B	Low	$m_{2,3}$
X_3	C	Average	$m_{3,3}$
X_4	C	Average	$m_{4,3}$
X_5	B	Low	$m_{5,3}$
X_6	A	High	$m_{6,3}$
X_7	B	Low	$m_{7,3}$

Where:

- $m_{1,3}(\{Finance\})$=0.5; $m_{1,3}(\{Finance, Accounts\})$=0.3 and $m_{1,3}$ (Θ_3)=0.2
- $m_{2,3}(\{Finance\})$=0.8 and $m_{2,3}(\Theta_3)$=0.2
- $m_{3,3}(\{Marketing\})$=0.8; $m_{3,3}(\{Finance, Accounts\})$=0.1 and $m_{3,3}(\Theta_3)$=0.1
- $m_{4,3}(\{Accounts\})$=0.8 and $m_{4,3}(\Theta_3)$=0.2
- $m_{5,3}(\{Marketing\})$=0.8 and $m_{5,3}(\Theta_3)$=0.2
- $m_{6,3}(\{Finance, Accounts\}) = 0.8$ and $m_{6,3}(\Theta_3)$=0.2
- $m_{7,3}(\{Accounts\})$=0.8 and $m_{7,3}(\Theta_3)$=0.2

Let us try to apply the proposed method for selecting the 2 initial modes.
We have, initially, to define the dissimilarities matrix. To this end, for each object X_i, compute the dissimilarities $D_{i,u}$, where $i, u \in \{1, .., 7\}$, using the dissimilarity measure defined by Equation 3.3.
The dissimilarity matrix is defined as follows:

$$;$$

$$\begin{array}{c} \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{array} \begin{array}{c} \begin{array}{ccccccc} X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 \end{array} \\ \begin{pmatrix} 0 & 2.21 & 2.72 & 2.67 & 2.75 & 0.52 & 2.67 \\ 2.21 & 0 & 2.78 & \mathbf{2.8} & 0.8 & 2.57 & 0.8 \\ 2.72 & 2.78 & 0 & 0.78 & 2.06 & 2.77 & 2.78 \\ 2.67 & 2.8 & 0.78 & 0 & 2.8 & 2.57 & 2 \\ 2.75 & 0.8 & 2.06 & 2.8 & 0 & 2.8 & 0.8 \\ 0.52 & 2.57 & 2.77 & 2.57 & 2.8 & 0 & 2.57 \\ 2.67 & 0.8 & 2.78 & 2 & 0.8 & 2.57 & 0 \end{pmatrix} \end{array}$$

Note that $D_{i,u} = 0$, if $i = u$.
$Max(D) = 2.8$. The first occurrence of D, with $D_{i,u} = 2.8$ is obtained for $i = 2$ and $u = 4$.
So, we allocate the two objects X_2 and X_4 to the two initial modes.
If we have $k > 2$, we will iterate until obtaining the k initial modes as explained by the algorithm (see subsection 3.4.2).

3.4.4 Experimental analysis

The objective of this subsection is to study the impact of our new proposed selection method of the initial modes on the clustering results.

To evaluate the performance of the proposed initialization strategy, we use several categorical data sets from the UCI Machine Learning Repository (Murphy and Aha, 1996) which are described by the Appendix A.

Recall that for each data set, the number of clusters is set to be the known number of its class labels. For instance, the number of clusters is set to be 2 on Mushroom data. Note that the following results are originally proposed in (Ben-Hariz et al., 2008).

Comparison and evaluation strategies

We will compare our new method against the both standard K-modes initialization methods, namely the random selection of k distinct instances and the choice of the k first distinct objects as initial modes. In the following, we will denote the first method by BKM_{Init1} and the second one by BKM_{Init2}, while our proposal will be denoted BKM_{Init3}.

We carried out 10 runs of the three tested initialization methods. Uncertain version of the second initialization method has been considered following its adaptation in the belief framework. Indeed, the already developed belief clustering approach (Ben-Hariz et al., 2006) adoptes the first initialization approach.

Results and discussion

Our proposal is compared against both random initialization methods. The clustering accuracy is the common clustering evaluation criterion for the most of initialization literature references. Thus, in our case we will also consider this metric to judge the performance of our algorithm.

Indeed, three versions of BKM methods are considered respectively to three initialization techniques. They are BKM_{Init1}, BKM_{Init2} and BKM_{Init3}, where the first one corresponds to the total randomness selection. the second one considers the k first distinct objects as initial cluster modes. Finally, the third approach selects the initial modes based on our max-dissimilarity based method developed in this Chapter.

Our simulations are performed for two cases, namely, the certain case (Table 3.2) and the uncertain one (Table 3.4) by considering the clustering accuracy criterion. On the other side, as time indicators, we will compute and report the different time running and iteration numbers. These latter are necessary to reach the stability of the final solution. The respective obtained results are tabulated in Tables 3.3 and 3.5.

1. Certain case:
   It consists in testing and comparing the efficiency of random belief K-modes method versions (BKM_{Init1} and BKM_{Init2}) and its improved version (BKM_{Init3}), when there is no uncertainty in attribute values. It means that each attribute is known with certainty and it has a unique value. Recall that the certain belief function concept is used to represent the certain attribute values under belief function theory context.

   Clustering accuracy results The quality of clustering results in such certain framework is reported in the following table for some UCI databases.

Table 3.2: Clustering accuracy (%)(Certain case)

Database	BKM_{Init1}	BKM_{Init2}	BKM_{Init3}
Breast Cancer Wisconsin (BCW)	79.78	80.10	80.25
Car evaluation (CE)	81.72	82.05	82.90
Congressional voting records (CVR)	88.77	88.95	89.05
Hayes-Roth (HR)	77.50	77.65	78.45
Lenses (LS)	88.56	89.20	89.90
Mushroom (MM)	77.25	76.95	79.65
Nursery (NY)	88.23	90.25	92.95
Spect heart (SH)	86.83	86.90	89.51
Tic-Tac-Toe Endgame (TE)	71.30	72.15	75.45
Zoo (ZO)	74.23	74.16	75.58

   Time results To decide which initialization method among the three tested ones guarantees the faster solution, we have to compute the running time and the number of iterations respectively to each of these approaches (see Table 3.3).

Table 3.3: Time performances (Certain case)

Databases	BKM_{Init1}		BKM_{Init2}		BKM_{Init3}	
	Time(s)	Iter	Time(s)	Iter	Time(s)	Iter
Breast Cancer Wisconsin (BCW)	348	6	331	7	305	7
Car evaluation (CE)	452	10	468	11	411	8
Congressional voting records (CVR)	311	9	312	10	268	7
Hayes-Roth (HR)	213	10	186	11	154	9
Lenses (LS)	63	6	59	5	49	5
Mushroom (MM)	540	10	578	11	408	8
Nursery (NY)	648	12	641	12	521	11
Spect heart (SH)	199	8	200	8	187	7
Tic-Tac-Toe Endgame (TE)	297	11	285	11	209	7
Zoo (ZO)	294	10	266	8	161	6

Note that the second and the third columns of Table 3.2 are relative to results of the initial versions of the BKM method which are based on standard selection method for choosing the initial modes (BKM_{Init1} for k random choice and BKM_{Init2} with k first items). However, the last one represents those of our improved BKM version, by inducing the proposed method advanced in Section 3.4 namely BKM_{Init3}. It is the same representation in Table 3.3.

It can be seen that the clustering results have been improved when the modes are chosen by our proposed method, the third column for each database, in comparison to other selection approaches (first and second column values). For example, the accuracy mean for Nursery database goes from 88.23% with BKM_{Init1}, 90.25% with BKM_{Init2} to 92.95% by using our proposal.

We find, as shown by Table 3.3, that the BKM_{Init3} is generally faster to reach final cluster solution than the other ones. For instance, the running time for Nursery base is 521 seconds with BKM_{Init3}, while it is 648 and 641 seconds with respectively BKM_{Init1} and BKM_{Init2}. The iteration number results of this same base are 12, 12 and 11 by using respectively BKM_{Init1}, BKM_{Init2} and BKM_{Init3}.

2. Uncertain case:
   Within this uncertain framework, four uncertainty levels will be studied as defined in Appendix A. By the following, we recall their intervals. Indeed, for each uncertain attribute, we have to specify its uncertainty degree which must be in $]0, 0.25]$, $]0.25, 0.5]$, $]0.5, 0.75]$ or $]0.75, 1]$.

Clustering accuracy results The different results carried out from testing the different methods in uncertain context respectively to the four levels of uncertainty degrees are reported in the following Table 3.4. The average accuracies are also reported in this table.

To conduct experimental comparison and to verify the efficacy of our proposed approach (BKM_{Init3}), let us move to the following table.
Detailed results of four levels are tabulated in Table 3.4 from column 2 to column 13. The three last columns of this table summarize the first ones in order to compute for each database the mean of four levels accuracies.

Time results We conduct time experimentation, here, and the resulting criteria will be reported by the two following tables. As already done, four levels will be studied and the corresponding results will be reported. Therefore, their summary will be computed ans will be also tabulated.

Table 3.4: Clustering accuracy (%)(Uncertain case)

Databases	Level 1			Level 2			Level 3			Level 4			Mean		
	Init1	*Init2*	*Init3*	*Init1*	*Init2*	*Init3*	*Init1*	*Init2*	*Init3*	*Init1*	*Init2*	*Init3*	*Init1*	*Init2*	*Init3*
BCW	82.07	83.11	84.07	81.23	82.07	82.99	79.78	80.10	80.25	73.17	75.45	79.33	79.06	80.18	81.66
CE	82.99	85.01	85.81	82.33	84.11	84.75	81.72	82.05	82.90	72.99	73.15	78.73	80.00	81.08	83.05
CVR	90.12	90.07	91.17	89.11	89.13	90.15	88.77	88.95	89.05	79.45	78.01	80.03	86.86	86.54	87.60
HR	79.14	79.33	80.73	77.95	78.13	79.91	77.50	77.65	78.45	74.11	73.82	74.79	77.18	77.23	78.47
LS	90.41	90.27	92.12	89.13	89.93	90.23	88.56	89.20	89.90	80.13	81.02	82.14	87.06	87.61	88.60
MM	80.45	79.45	92.45	79.13	79.38	82.16	77.25	77.95	79.65	72.47	73.08	73.18	77.33	77.47	81.86
NY	90.77	92.11	93.75	89.47	91.44	93.06	88.23	90.25	92.95	81.13	83.24	84.48	87.40	89.26	91.06
SH	88.13	88.75	92.04	87.17	87.34	90.83	86.83	86.90	89.51	79.19	79.23	80.33	85.33	85.55	88.18
TE	76.77	78.13	82.14	74.75	75.51	79.13	71.30	72.15	75.45	68.71	69.19	70.13	72.88	73.75	76.71
ZO	78.13	78.48	80.14	76.71	76.88	77.99	74.23	74.46	75.58	69.98	70.33	70.72	74.76	75.04	76.11

Table 3.5: Running time(s)(Uncertain case)

Databases	Level 1			Level 2			Level 3			Level 4			Mean		
	Init1	*Init2*	*Init3*	*Init1*	*Init2*	*Init3*	*Init1*	*Init2*	*Init3*	*Init1*	*Init2*	*Init3*	*Init1*	*Init2*	*Init3*
BCW	312	308	310	352	321	331	377	349	352	406	387	368	362	341	340
CE	472	412	398	497	477	412	516	502	457	543	517	482	507	477	437
CVR	297	288	276	312	326	298	352	366	311	393	401	347	339	345	308
HR	213	171	166	241	187	171	268	203	199	316	247	203	260	202	185
LS	59	57	49	64	63	59	71	61	64	78	67	68	68	62	60
MM	498	479	394	591	603	482	632	699	503	689	707	547	603	622	482
NY	403	399	381	442	431	407	705	714	684	713	721	698	566	566	543
SH	194	187	169	211	208	194	243	264	226	312	294	271	240	238	215
TE	291	305	253	379	347	299	406	399	335	476	411	357	388	366	311
ZO	313	293	194	321	304	201	355	331	233	391	389	329	345	329	239

Table 3.6: Iteration number (Uncertain case)

Databases	Level 1			Level 2			Level 3			Level 4			Mean		
	Init1	*Init2*	*Init3*	*Init1*	*Init2*	*Init3*	*Init1*	*Init2*	*Init3*	*Init1*	*Init2*	*Init3*	*Init1*	*Init2*	*Init3*
BCW	6	7	5	7	7	6	8	8	8	11	10	10	8	8	7
CE	11	9	10	13	11	12	15	13	14	16	15	15	14	12	13
CVR	8	8	8	9	10	9	11	12	10	13	14	12	10	11	10
HR	11	8	8	13	10	9	14	12	12	15	13	12	13	11	10
LS	5	6	5	6	7	6	8	6	7	8	7	7	7	7	6
MM	9	9	7	11	11	9	13	14	11	14	14	12	12	12	10
NY	11	10	9	12	14	12	15	16	15	16	16	15	14	14	13
SH	7	7	6	8	9	8	9	10	10	13	12	11	9	10	9
TE	10	9	9	12	11	10	14	13	11	15	13	12	13	12	11
ZO	9	7	6	10	8	7	12	10	8	13	11	11	11	9	8

Using our selection initial method BKM_{Init3}, shows that it has a better accuracy compared with repectively to BKM_{Init1} and BKM_{Init2}. For example for Hayes-Roth database, this accuracy mean is 78.47% by using our proposal BKM_{Init3}, while it is 77.18% with BKM_{Init1} and 77.23% with BKM_{Init2}.

Note that the accuracy decreases in average from Level 1 to Level 4 (the uncertainty degree increases) as shown by previous table (Table 3.3).

In Tables 3.4 and 3.5, we report the average time performances of the different studied uncertainty levels. This table shows that BKM_{Init3} has a good effect on reducing both the running time and the number of iterations. So, it is generally faster than BK_{Init1} and BKM_{Init2}.
For example, the time requirement for Hayes-Roth database goes from 260 seconds with BKM_{Init1} and 202 seconds with BKM_{Init2} to 185 seconds with our initialization method. It is the same for the iteration number criteria. By considering this same database, this number is 13 and 11 using respectively BKM_{Init1} and BKM_{Init2}, while it is only 10 by applying the new approach developed within this chapter.

Altough, faster solutions which are provided by using our developed initialization method BKM_{Init3} for almost all databases, one exception is made if we consider the number of iterations. Indeed, for Car evaluation database, BKM_{Init2} reaches the stability in 12 iterations while for our method this number is 13.

The average accuracy and time indicators are graphically represented by Figures 3.1 and 3.2. These figures allow us to visualize the positive results of BKM_{Init3} compared to those giving by BKM_{Init1} and BKM_{Init2}.

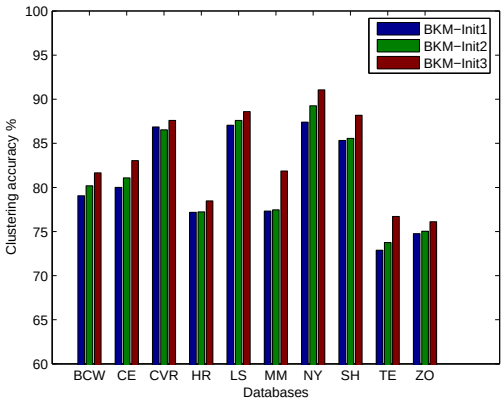
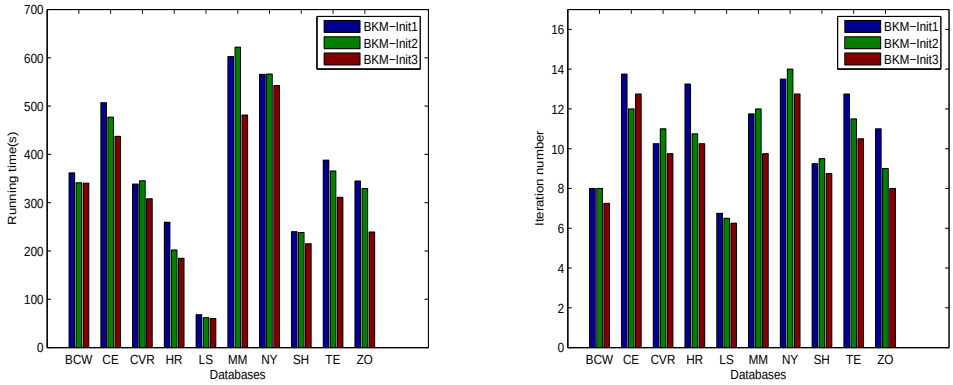


Figure 3.1: Clustering accuracy results under uncertainty (Mean)



(a) Running time(s) (b) Iteration number

Figure 3.2: Time indicators results under uncertainty (Mean)

As in the certain framework, this uncertain one also attests the efficiency of our initialization method developed under uncertainty based on the three used evaluation criteria namely accuracy, running time(s) and number of iterations.

The above results show that the proposed approach deals with uncertain instances as good as with certain instances. For instance, if we analyze the results shown in Figure 3.1, we remark that the clustering accuracy (uncertain case) remains high in average as in the certain case. These results, certify that our proposed approach is also well adapted to classify instances with uncertain attribute values.

Experimental conclusion

As a conclusion, we have to note that these values are purely experimental and that depend on the used database and even on the used uncertainty degree and the considered uncertainty percent (number of uncertain items) within a given database.

This is clear that the first tested initialization approach BKM_{Init1} suffers from the total randomness problem, while the main drawback of the second one is the object order. In fact, our proposal solves these drawbacks. This is thanks to use of dissimilarity measure (no random choice and no impact of the object order).

Low computational time complexity is experimentally obtained (see Table 3.4) by using our proposal to select the initial modes.

During all experimentations, our proposal namely BKM_{Init3} has always reached one final solution with a high clustering quality (more than 70% for all cases and with all tested databases).

3.5 Conclusion

The belief K-modes method is efficient in clustering categorical data sets with uncertain attribute values. This latter combines the standard K-modes method and the belief function theory concepts to deal with data sets characterized by such uncertainty form. Nonetheless, its use of random initialization compromises its effectiveness and its ability to converge into a global optimum. Therefore, a possible improvement of the BKM method results via the new proposed starting method is ensured. Its basic idea focuses on maximizing the dissimilarity between all selected modes.

Indeed, by using this initialization process the attained performances are generally better than to the other tested methods on all data sets. The order presentation of the input data is not important for our proposal and the randomness aspect is eliminated. In fact, this developed approach will also be considered to adopt within the handled dynamic context in the following chapters.

In the rest of our thesis work, we will detail our new solution, denoted by dynamic belief clustering framework in order to ensure the clustering task under combined, uncertain and dynamic environment.

4

Handling dynamic partition based on belief clustering

4.1 Introduction

Recently, some research works focus on dynamic clustering problem. Some of them use such dynamic principal to learn the best number of clusters or to cluster online data sets.

The major literature references represent new versions of already standard clustering approaches to handle mainly the incremental process. Indeed, using (Aranganayagi and Thangavel, 2009; Aranganayagi and Thangavel, 2010) methods, the data are processed one by one and the clusters are created consequently. These latter handle certain categorical data, the frequency is their core concept. In (Boukharouba and Lecoeuche, 2008), a SVM-based approach was proposed to deal with both cases increasing as well decreasing. This method works under numerical framework.
Authors in (Beringer and Hüllermeier, 2007; Bordogna et al., 2006; Peters et al., 2010) present a more sophisticated methods dealing with both uncertain and dynamic clustering parameters. Although, these methods are developed for uncertain context, the only handled uncertainty form concerns the cluster membership of different items under fuzzy set theory framework. Nonetheless, it will be interesting to propose combined dynamic methods which are able to ensure their task in dynamic environment for uncertain data sets. The managed uncertainty, here, is related to the attribute values, while the dynamic aspect is respective to the cluster collection cardinality. The belief function theory framework will be used in this work to represent and manage these uncertain attribute values.

One of the main goals of this thesis is to update the cluster partition of uncertain data in presence of any change respectively to different classification parameters. This

is must be done without performing the reclustering again. Hence, in this chapter, we will present new clustering algorithms that are able to update, dynamically, cluster partition of uncertain data set to create new or to suppress old cluster(s). These proposed methods extend well known algorithm developed in the context of static categorical clustering namely K-modes algorithm. Its uncertain version, recently proposed in (Ben-Hariz et al., 2006) and improved in (Ben-Hariz et al., 2008), is the adopted one. This latter is named Belief K-Modes method (BKM) respectively to the belief function framework.

The first part of this chapter details firstly the problem studied here. Then, the general principle of our approaches is presented. Thereafter, the basic distances relative to dynamic belief K-modes methods are defined. After that, all dynamic procedures are described by considering both studied optimization problems. To this end, their ideas and algorithms are given. To evaluate the performance of our proposals, a comparative experimental study of dynamic versus static methods is conducted, in the second part of this chapter.

4.2 Problem definition

Several clustering techniques are available in the literature (Jain and Dubes, 1988). Most of them are static or off-line meaning that the clustering parameters are entirely available at the beginning of the process and do not change over time. Unfortunately, such assumptions are often violated in real-world applications given that it is expensive or because it takes a long time to obtain all information from the beginning. Thus, these latter must be dynamically detected or modified.

To deal with such problems, in recent years, so-called dynamic clustering have attracted considerable attention in different fields of computer science. Uncertain extensions are also proposed to handle the uncertainty drawback as mentioned earlier. Indeed, dynamic clustering approaches are needed in order to maintain an up-to-date clustering structure. To be more general, preferably, these proposed techniques must be able to handle also uncertainty problem.

Here, the cluster number increase as well as its decrease will be considered dynamically. It seeks a better understanding of the generated cluster partition. It is a new clustering approach family that is able to adapt the cluster collection and generate an extended or a reduced partition of imperfect data under the belief function framework. The handled uncertainty exists in attribute values.

Let us try to distinguish between these both cases via the following real application example.
Indeed, it is natural to assume that customers, with similar attributes or requirements, want the same kinds of products. Thus, they must be grouping together. This clustering segmentation task widely uses clustering techniques to discover such groups. As instance, the clustering of households is done on the basis of some specificities such as income, age, number of purchases, number of children, etc.
In fact, the household behaviors led to their segmentations. An adopted segmentation in two groups by one supermarket may be thereafter, updated seen that this not well represent the actual requirements of its customers. Then, the marketing expert decides to increase the household group number. This is done in order to more represent the reality. The opposite situation may be encountered, and the elimination of one existing group could be performed. On the other side, the different characteristics of these households may be known with uncertainty.

One of our objectives is to develop new dynamic clustering under uncertainty that will be able to update any partition of observations following their increase or decrease by one or more groups.

The dynamic belief clustering can be formally defined as follows: Given one cluster partition $P = \{C_1, C_2, .., C_l, .., C_k\}$ which is the result of the BKM execution, the dynamic process consists in updating this partition following the increase or the decrease of its cardinality. To this end, the distance measure concept will be used. Thus, we will obtain P^+ or P^- respectively to incremental or decremental procedures.

To simplify this problem formulation, its principle input and output are defined in the following with as goal the minimization or the maximization of one cost function:

- Input: $P = \{C_1, C_2, .., C_l, .., C_k\}$.
- Output: P^+ or P^-.

4.3 Dynamic clustering preliminaries under uncertainty

4.3.1 General principle

In most cases, clustering algorithms produce clusters with low intra-cluster distances (high inter-cluster similarity). This will be considered as the core of the developed methods. Knowing that all K-modes versions seek to optimize an objective function, it will be the same for the proposed dynamic clustering approaches. Indeed, the op-

timization problems of all proposed approaches will be defined respectively to this distance notion.

For all developed incremental as well as decremental clustering methods, an input partition is provided as a point of departure. This latter results from the BKM run since the handled data sets, here, are all categorical and uncertain. In fact, like in belief static framework (Ben-Hariz et al., 2008), within dynamical one, Belief Clustering for Dynamic Partition ($BCDP$) context, objects may also be characterized by uncertain attribute values represented through bba's concept.

Based on this cluster partition of k groups, we have to generate an extended one of k^+ groups or a reduced one of only k^- following an updating process. This issue (increase or decrease of the initial cluster collection) arises when:

1. The expert chooses to increase or decrease the partition cardinality to improve the collection understanding and interpretation.
2. The clustering results which are provided, using the offline technique and producing k groups, are with poor quality.
3. The clusters' densities (high or low values) necessitate their splitting or their merging.

Let us mention that this last case can appear when non-stationary data clustering is performed. Indeed, a large number of absorbed data may change the density of clusters, so that the splitting procedure will be necessary, especially, when the clusters' number is not known a priori and can increase. When items are eliminated, low clusters' densities may be detected. Thus, clusters fusion must be performed to deal with this problem.
For all these three cases, we have to improve an appropriate objective function which will be exposed in the following.

Indeed, the similarity/dissimilarity measures are the basic concepts to develop these techniques since these distances play a key role in the intuitive notion of clustering. In the following, we will define all measure categories used by the advanced dynamic strategies.

4.3.2 Basic definitions

Some distance metrics which are judged as the core concepts of the developed dynamic clustering approaches will be defined in the following.

- The intra-cluster distance:

$$DM(C_l) = \sum_{i=1}^{n_l} D(X_i, Q_l). \tag{4.1}$$

It measures, for one cluster, the distance between all its objects X_i and its cluster mode Q_l, where $i \in \{1, .., n_l\}$ and $l \in \{1, .., k\}$ and $n_l = |C_l|$.
Indeed, this within-cluster distance verifies this inequality: $0 \leq DM(C_l) \leq n_l * s$, where s is the attribute number. A distance zero means that all objects of this cluster are identical to its mode, while $n_l * s$ value means the opposite case: total dissimilarity of the mode and the items of this group. This distance measures the cluster cohesion or compactness.

- The inter-cluster distance:

$$DI(C_l, C_r) = \sum_{i=1}^{n_l} D(X_i, Q_r) + \sum_{u=1}^{n_r} D(X_u, Q_l) \tag{4.2}$$

with n_l and n_r are respectively the cardinality of the cluster C_l and C_r. Q_l and Q_r are their modes, for $l, r \in \{1, .., k\}$. The D measure is the one advanced in Chapter 3 relative to belief context (see Equation 3.3).
This is a modified version of the average neighbor distance already defined by Equation 2.7 by considering for one cluster all its objects, while for the other one only its representative since this latter summarizes all cluster items.
Similarly, this between-cluster distance verifies this inequality: $0 \leq DI(C_l, C_r) \leq n_l * s + n_r * s$, from full similarity to total dissimilarity.
Note that it is possible to consider any proximity measures for cluster pairs as detailed by subSection 2.3.3. This measures the separation of cluster C_l to C_r one.

- The inter-object dissimilarity:

$$D(X_{i_1}, X_{i_2}) = \sum_{j=1}^{s} d(x_{i_1,j}, x_{i_2,j}). \tag{4.3}$$

with i_1 and $i_2 \in \{1, .., n\}$ and $j \in \{1, .., s\}$. The d component is computed as detailed in Chapter 3 by Equation 3.4 under belief function framework. Thus, $0 \leq D(X_{i_1}, X_{i_2}) \leq s$.
The distance sum of an object X_i respectively to all other objects, of one con-

sidered data set of n instances, is defined by:

$$DS(X_i) = \sum_{u=1, u \neq i}^{n} D(X_i, X_u) \tag{4.4}$$

This distance sum is as follows: $0 \leq DS(X_i) \leq (n-1) * s$.

All exposed measures must be adapted to uncertain context within belief function theory framework by adopting the already defined belief distance in Chapter 3.
Let us recall that the first distance (4.1) reflects the cluster cohesion, compactness or hightness, while the second measure (4.2) is a cluster separation indicator.

4.4 Optimization based dynamic clustering under uncertainty

4.4.1 Dynamic clustering using cohesion minimization under uncertainty

In general, it is assumed that clustering techniques seek to minimize the intra-cluster distance sum. Therefore, we will consider this for both developed incremental and decremental methods. Thus, these approaches will be called Incremental K Belief K-modes Method ($IKBKM_{Min}$) and Decremental K Belief K-modes Method ($DKBKM_{Min}$). Indeed, these techniques will be developed to increase or decrease the cluster number k to reach respectively k^+ or k^- clusters.
In order to carry this, let us recall the intra-cluster minimization function definition as follows:

$$Minimize \sum_{l=1}^{k} \sum_{i=1}^{n_l} D(X_i, Q_l) \tag{4.5}$$

with n_l is the cluster cardinality of the cluster C_l. It is the simplest and most widely used criterion measure for clustering problems. Note that there are other manners to define the intra-cluster distance such as the sum of distances between all its objects. In other word, it consists in minimizing the maximum of intra-cluster distance.

Incremental clustering to increase the cluster number

$IKBKM_{Min}$ **idea**

The main idea is to select one cluster, among our k initial ones, to reorganize based on its intra-cluster distance value. The cluster with the maximum intra-dissimilarity measure is the worst organized, so it will be the firstly rearranged one hoping the improvement of its compactness. Therefore, we have to look for the object(s) which must move to new cluster. This is done based on the sum of dissimilarities of one item corresponding to all other objects within this same cluster. In fact, the most

dissimilar object(s) respectively to all other instances must be reassigned to improve the local cluster cohesion and consequently the global clustering compactness. This is considered as the *initialization step*, while the *updating phase* seeks to obtain the clusters' stability. It is ensured via the monitoring of the intra-cluster distances, following the objects reassignment, to minimize their sum.

Algorithm outlines

The objective of the algorithm is to minimize the intra-dissimilarity sum, so the stopping criterion will be reached when between two successive iterations there is not any improvement of the intra-dissimilarity sum (cost function). Another stopping criterion may be considered namely the maximum number of iterations $noMaxIter$ since this incremental version might iterate for a long time with very small improvement in the intra-cluster distance sum.

The following $IKBKM_{Min}$ algorithm summarizes, in a rather informal way, the main steps of the incremental proposal. The initialization phase is detailed by steps 1 to 3, while all other steps constitute the updating phase.

$IKBKM_{Min}$ **algorithm (**P **partition,** $k * 1$ **intra-cluster vector,** $n * n$ **inter-object matrix,** $NoMaxIter$**)**

Data: P partition, $k * 1$ intra-cluster vector, $n * n$ inter-object matrix, $NoMaxIter$
Result: P^+ partition, F_I
begin
Initialization phase:
1. Derive the cluster with the max(DM) among the k*1 intra-cluster vector elements. //It is possible to derive more than one cluster when the expert wishes to add several //clusters.
2. Compute the initial total cohesion by using Equation 4.5.
3. Create a new cluster initially empty.
Updating phase:
4. The objects with max(DS) (Equation 4.4) of the cluster with max(DM) are assigned to the cluster with min(DM).
5. Update the cluster modes and intra-cluster distances (Equation 4.1) of the two altered clusters as well as the total cohesion.
6. If an improvement (reduction) of the total cohesion is detected and $NoMaxIter$ is not reached, repeat from step 4, else terminate.
7. $F_I \leftarrow$ *Updated total cohesion.*
end

Initially, the cluster with the maximum of internal distance is considered as the departure point of the incremental process, while the cluster with the minimum value is the new one which is empty at the beginning. The intra-distance of this latter is equal to zero.

Let us mention that F_I parameter is equivalent to the objective function to minimize (Equation 4.5), while I corresponds to the incremental process.
The elements of the $k * 1$ intra-cluster vector are computed based on Equation 4.1, whereas $n*n$ inter-object matrix components are defined by Equation 4.3. This latter matrix is already defined in subSection 2.3.1.

Note that the user may decide of the cluster number x to add to an initial partition. Thus, the first step selects the clusters giving the x first intra-cluster dissimilarities following their ranking in descending order. It consists in x cluster splitting. Each cluster is split into two groups. The new one contains object(s) which are the most dissimilar/far from all other ones in the original group. However, the updating phase remains the same.

Decremental clustering to reduce the cluster number

$DKBKM_{Min}$ **idea**

Intuitively the decremental process is the reversal of incremental one and vice versa. Indeed, by considering the same objective function, the decremental version will be detailed as follows. Having k clusters, we want to obtain only k^- ones without any object deletions. To reduce this number, we have to decide which two groups among the initial ones will be merged?
To this end, all fusion possibilities must be tested (2 clusters among k initial ones). The corresponding number is C_k^2. To decide which combination is better, the resulting intra-cluster distances must be computed and compared to original one. Recall that we obviously want to minimize this measure. Thereafter, $DKBKM_{Min}$ algorithm iterates until stability during its updating phase.

Algorithm outlines

Let us indicate that the same stopping criteria as the incremental method are considered here. $DKBKM_{Min}$ algorithm outlines our first proposed decremental approach.

For the updating process, the F_I must be replaced by F_D seen that the decremental context is considered here.
The intra-cluster measure is always considered under this decremental context. Indeed, the reassignment of objects from one cluster to another is based on the cohesion of these latter, the departure cluster and the arrival one, as with $IKBKM_{Min}$.

$DKBKM_{Min}$ algorithm (P partition, $k * 1$ intra-cluster vector, $n * n$ inter-object matrix, $NoMaxIter$)

Data: P partition, $k * 1$ intra-cluster vector, $n * n$ inter-object matrix, $NoMaxIter$
Result: P^- partition, F_D
begin
Initialization phase:
1. Compute the initial total cohesion by Equation 4.5.
2. Test all fusion possibilities C_k^2 which is equal to $k * (k-1)/2$.
3. Preserve the solution which provides the best reduction of the cluster cohesion (computed as the difference between the cluster cohesion of the generated cluster, following the merging operation, and that of two original clusters).
Updating phase:
4. The objects with max(DS) (Equation 4.4) of the cluster with max(DM) is/are assigned to the cluster with min(DM).
5. Update the cluster modes and the intra-cluster distances (Equation 4.1) of the two altered clusters as well as the total cohesion.
6. If an improvement (reduction) of the total cohesion is detected and $NoMaxIter$ is not reached, repeat from step 4, else terminate.
7. $F_D \leftarrow$ *Updated total cohesion.*
end

Let us mention that within this decremental framework, the expert may decide which group(s) eliminate and consequently their number (one or more). Therefore, the initialization phase tests only $k-1$ fusions instead of $k*(k-1)/2$ ones. Steps 1 and 3 of this phase are the same as in the case where the expert is not involved. Thereafter, the updating phase must be performed to reorganize the initial collection.

4.4.2 Separation maximization based dynamic belief clustering

Clustering problems usually have one of the following forms namely the minimization or the maximization problems. Since, we have already dealt with the first one in the last section, let us now consider the second alternative and try to propose the two respective dynamic versions (incremental and decremental approaches).

Indeed, several kinds of inter-cluster distance may be mentioned and considered such as those exposed in Chapter 2 (Equations 2.4, 2.5, 2.6 and 2.7). The used maximization function, here, considers for one cluster all its objects, while for the other one only its representative. This latter is adopted since it summarizes all items of the cluster obviously:

$$Maximize \sum_{l,r=1,r>l}^{k} (\sum_{i=1}^{n_l} D(X_i, Q_r) + \sum_{u=1}^{n_r} D(X_u, Q_l)) \tag{4.6}$$

This optimization criterion consists, in other word, in maximizing the minimum of inter-cluster distance.

Incremental version

$IKBKM_{Max}$ **idea**

The incremental version relative to the maximization objective function consists of initialization and updating phases like for $IKBKM_{Min}$ one. Firstly, we proceed to split one existing group, but which cluster must be split?
In order to decide, all k possible divisions must be revised (each time one cluster is split into two). To choose which alternative to adopt, the inter-cluster distances must be calculated with as objective their sum maximization. The same updating process as the already exposed dynamic approaches, in subsection 4.4.1, is here applied.

Algorithm outlines

Note that the splitting operation (Step 1) consists in the creation of two clusters from an initial one. The most far observations from all the other ones will be reassigned into new empty cluster.

$IKBKM_{Max}$ algorithm (P partition, $k*k$ inter-cluster matrix, $k*1$ intra-cluster vector, $n*n$ inter-object matrix, $NoMaxIter$)

Data: P partition, $k*k$ inter-cluster matrix, $k*1$ intra-cluster vector, $n*n$ inter-object matrix, $NoMaxIter$
Result: P^+ partition, F_I
begin
Initialization phase:
1. Test all k splitting possibilities.
2. Preserve the solution which provides the maximum of separation improvement (computed as the difference between the cluster separation sum of the two resulting clusters respectively to all other $k-1$ groups ($2*(k-1)$ inter-cluster distances) and that of the original cluster to split respectively also to $k-1$ clusters).
Updating phase:
3. The objects with max(DS) (Equation 4.4) of the cluster with max(DM) are assigned to the cluster with min(DM).
4. Update the cluster modes and the intra and inter-cluster distances (Equations 4.1 and 4.2) of the two altered clusters as well as the total separation (Equation 4.6).
5. If an improvement (increase) of the total separation is detected and $NoMaxIter$ is not reached, repeat from step 3, else terminate.
6. $F_I \leftarrow$ *Updated total separation.*
end

The second step of the updating phase computes the inter-cluster distances of the two altered clusters and updates the respective global quality (total separation). When the inter-dissimilarity improvement (growing) is negligible or the predefined iterations number is reached, the process is terminated. This step also recomputes the intra-clusters distances seen that they are the considered ones during the object assignments.
Like with the first incremental method, here, the expert may also participate and provide any additional information concerning the number of clusters to create.

Decremental counter-part

$DKBKM_{Max}$ idea

In order to produce smaller partition, at first we have to find which two groups among the initial ones will be merged as already done under minimization context. This is ensured via the proposed inter-cluster maximization process. It is the aim of the initialization phase. After that, many rearrangements between objects on the remaining k^- clusters will be possible during the second main phase namely the updating phase.

Algorithm outlines

To recapitulate, $DKBKM_{Max}$ algorithm outlines the proposed maximization-based decremental approach. First, the inter-cluster distance concept is used to decide which clusters to merge (step 1). Indeed, the two most similar groups will be fused. Then, the update process is performed. During this phase, the intra-cluster distances are also considered for the object assignments, while the internal ones are updated iteratively to achieve their maximal sum.

$DKBKM_{Max}$ algorithm (P partition, $k * k$ inter-cluster matrix, $k * 1$ intra-cluster vector, $n * n$ inter-object matrix, $NoMaxIter$)

Data: P partition, $k*k$ inter-cluster matrix, $k*1$ intra-cluster vector, $n*n$ inter-object matrix, $NoMaxIter$
Result: P^{-} partition, F_D
begin
Initialization phase:
1. Derive the two clusters which are the most similar ones with the minimum of external distance or separation (Equation 4.6).
2. Compute the corresponding internal distance or compactness (Equation 4.5) of the resulting cluster.
Updating phase:
3. The objects with max(DS) (Equation 4.4) of the cluster with max(DM) is/are assigned to the cluster with min(DM).
4. Update the cluster modes and the intra and inter-cluster distances (Equations 4.1 and 4.2) of the two altered clusters as well as the total separation (Equation 4.6).
5. If an improvement (increase) of the total separation is detected and $NoMaxIter$ is not reached, repeat from step 3, else terminate.
6. $F_D \leftarrow$ *Updated total separation.*
end

Hereby, our decremental method ($DKBKM_{Max}$) gives to user the possibility to choose the cluster(s) to eliminate and implicitly their number.

4.4.3 Concluding remarks

To summarize, it is clear that for both versions of incremental process, named $IKBKM_{Min}$ and $IKBKM_{Max}$, the starting point is to decide which cluster, among the initial ones, must be split, while for the decremental versions, called $DKBKM_{Min}$ and $DKBKM_{Max}$, is which two groups should be merged. This concerns the initialization stage which extends or reduces by only one group the input cluster collection

without any rearrangement currently. In order to detect the possible changes, thereafter, we seek to reach the clusters stability by the objects reassignments based on the internal cluster distances. This is done, iteratively, during the updating stage.

To recapitulate, $IKBKM_{Max}$ must test all k splitting possibilities, whereas $IKBKM_{Min}$ derives only the cluster with the highest cohesion simply. On the other side, $DKBKM_{Min}$ must check all cluster fusion possibilities, while $DKBKM_{Max}$ selects the two clusters with the minimum separation to merge.
Indeed, these splitting and fusion tests are computationally expensive. Knowing that that the updating phase is almost the same for both versions of incremental or decremental proceudre, we can than conclude that $IKBKM_{Max}$ will be more expensive than $IKBKM_{Min}$. On the other side, $DKBKM_{Min}$ is more expensive than $DKBKM_{Max}$.

Recall that the standard K-modes algorithm requires time proportional to the product of the number of data items n, number of features s, number of clusters k and the number of iterations $NoMaxIter$. Each developed extension of this approach, incremental as well decremental techniques, must take into consideration all these parameters.

All proposed dynamic clustering approaches within the belief function framework can be considered as probabilistic techniques when the attribute uncertainty is expressed via the Bayesian belief function. Besides, via the *bba* concept, we can represent the certainty (tautology). Thus, the developed methods may be considered as generalization of the certain case.

4.5 Experimentations

In order to empirically asses the power of our proposals, we performed a set of experiments using data sets from the UCI repository (Murphy and Aha, 1996).
Several tests on the belief counter-part of the UCI real databases are performed to evaluate our methods in such dynamic uncertain environment. A brief description of the selected databases is given in Appendix A. Their belief counter-parts are created artificially. This creation procedure is also detailed by this same Appendix.

The objectives to carry experimentations on the chosen UCI real-world databases are:

1. Evaluate our two proposed incremental approaches under uncertainty $IKBKM_{Min}$ and $IKBKM_{Max}$ and conduct their comparison.

2. Evaluate our two proposed decremental approaches under uncertainty $DKBKM_{Min}$ and $DKBKM_{Max}$ as well as their comparative study.
3. Compare all obtained results with those given by the static method BKM.
4. Check the performance of our proposals for improving the clustering quality (first criterion).
5. Check the performance of our dynamic methods for reducing the running time (second criterion) and the iteration number (third criterion).

Hence, it consists in showing the feasibility and efficiency of our proposals within such uncertain dynamic context. Better results of our dynamic developments compared to the static ones are also shown for both cases certain and uncertain contexts.

4.5.1 Experimental strategies

The updating of the cluster partition cardinality requires an initial partition to consider as a departure point of all the dynamic process to perform. This input will be the BKM output namely the cluster collection of k groups.
Let us mention that the adopted BKM version is among the tested ones in Chapter 3 namely BKM_{Init3}. Actually, this developed new initialization approach under belief function framework will be the used one within the dynamic framework studied later. For case of simplification, BKM notation replaces BKM_{Init3} one for this Chapter and the subsequent ones.

However, in our experimentations, the evaluation of the belief dynamic clustering techniques requires two different strategies. The adopted one depends on the version to test (incremental or decremental).

For the two developed framework evaluation (cohesion-based methods or separation-based ones), two cases must be considered the dynamic and static ones. To this end, we have to proceed by applying the BKM approach initially and the generated partition will be the main input parameter to each dynamic method.

Under this dynamic framework, by giving the BKM partition and by using our incremental or decremental process we will update it and finally the produced (extended or reduced) cluster number must be equal to the actual class number of the considered data set since it is known a priori. If the incremental context is the considered one, the BKM method should produce $k-1$ clusters. Otherwise, for the decremental framework, the generated partition consists of $k+1$ groups.

To better evaluate the performance of our proposed clustering approaches under uncertainty and dynamic environment, we compare the obtained experimental results with those resulting from the static counter-part namely BKM method. It is done by performing the clustering again with the actual cluster number.

4.5.2 Empirical results and discussion

In order to corroborate the efficiency of the $IKBKM$ and the $DKBKM$ approaches, several experimental studies on real data sets with known cluster number were performed. The respective results were compared with those of a static clustering method BKM.

Preliminary results of these dynamic approaches, are summarized in (Ben-Hariz and Elouedi, 2010b; Ben-Hariz and Elouedi, 2010a; Ben-Hariz and Elouedi, 2010c; Ben-Hariz and Elouedi, 2014).

Indeed, the clustering accuracy as well as the time indicators are used to judge the dynamic clustering proposals performances. Such work is especially motivated by online applications for which a potentially changing clustering structure must be maintained over time more precisely following any detected data set change.

The respective experimental results for UCI repository data sets of Table A.1 (see Appendix A) are reported below. By applying both incremental as well as decremental procedures, the considered evaluation criteria (clustering accuracy criterion and time performance criteria) are computed and exposed to show our competitive results.

Let us mention that to control one form of randomness of these methods (equidistant items during the cluster assignment task), these techniques are launched ten times. The random choice of center location at the start of the algorithms, is handled by using our improved method BKM_{Init3} noted in the following by BKM.

In fact, during this subsection, there are two goals. The first goal is to test the dynamic proposals on benchmark data sets. The second one is to compare results of static and dynamic methods on one hand. On the other hand, the results of two developed versions respectively to incremental and decremental contexts are also confronted.

$IKBKM$ simulations

We start our evaluation process by the BKM versus $IKBKM_{Min}$ and $IKBKM_{Max}$ methods study. Following tables and figures detail the different results relative to the selected UCI databases firstly (certain case), and their belief counter-parts secondly (uncertain case). Indeed, for the three considered evaluation criteria, the results will be reported for both cases certain and uncertain ones.

1. Certain case:
   The following two tables show the experimental results for certain case, where certain belief functions are used to represent the certain databases.

   Clustering accuracy results We start our evaluation task by the clustering accuracy criterion to judge the quality of the resulting cluster partitions by using the dynamic methods. These results, tabulated in Table 4.1, will also be compared to those attained via the static approach.

Table 4.1: Clustering accuracy (%)(BKM vs $IKBKM$)(Certain case)

Databases	BKM	$IKBKM$	
		$IKBKM_{Min}$	$IKBKM_{Max}$
Breast Cancer Wisconsin (BCW)	85.30	92.06	87.40
Car evaluation (CE)	84.53	95.01	89.87
Congressional voting records (CVR)	85.65	93.57	89.75
Hayes-Roth (HR)	84.75	90.91	89.97
Lenses (LS)	84.65	89.94	89.91
Mushroom (MM)	84.96	89.79	87.17
Nursery (NY)	83.12	88.92	88.50
Spect heart (SH)	84.88	89.62	85.88
Tic-Tac-Toe Endgame (TE)	83.88	90.24	87.71
Zoo (ZO)	85.07	90.36	88.26

   Time results The second and third criteria are computed and reported in Table 4.2 for some UCI databases.

Table 4.2: Time performances (BKM vs $IKBKM$)(Certain case)

Databases	BKM		$IKBKM_{Min}$		$IKBKM_{Max}$	
	Time(s)	Iter	Time(s)	Iter	Time(s)	Iter
Breast Cancer Wisconsin (BCW)	205	6	183	5	185	5
Car evaluation (CE)	347	9	312	6	338	7
Congressional voting records (CVR)	218	6	203	4	227	5
Hayes-Roth (HR)	148	8	118	5	129	6
Lenses (LS)	48	5	38	4	44	5
Mushroom (MM)	383	7	325	6	371	8
Nursery (NY)	407	11	381	10	394	11
Spect heart (SH)	166	6	131	4	158	5
Tic-Tac-Toe Endgame (TE)	208	7	193	5	203	6
Zoo (ZO)	174	6	152	4	163	5

Based on accuracy criterion results in this certain context, tabulated in Table 4.1, we can conclude that $IKBKM_{Min}$ approach generates the most accurate clustering solutions for all tested databases. For example, the cluster collection induced from Mushroom database has 84.96% as clustering accuracy by using the static method, while it is 89.79% and 87.17% with respectively $IKBKM_{Min}$ and $IKBKM_{Max}$. Besides, we find from Table 4.2 that $IKBKM_{Min}$ is faster to update the cluster collection than BKM and $IKBKM_{Max}$. For instance for this same database, the time requirement goes from 325 seconds with the minimization-based approach to 371 seconds using the maximization-based one and 383 seconds respectively to their static counterpart. The smallest iteration number for this database is given by $IKBKM_{Min}$ which is equal to 6.

From the previous Tables (4.1 and 4.2), we observe that both $IKBKM$ dynamic versions are generally better than BKM static method in most cases within certain context.

2. Uncertain case:
   In the next part, all uncertain results respectively to the four considered uncertainty levels (Level 1, 2, 3 and 4) will be reported. More detailed information of this uncertainty parameter are exposed in Appendix A.

   Clustering accuracy results In Table 4.3, the uncertain results are detailed respectively to four studied uncertainty levels. Thereafter, an illustrative of all this values will be represented to summarize these uncertain experimentations via mean accuracy values per database for each tested approach (the two last columns).

   Time results The time requirements, by considering these different uncertainty degrees (Level 1, 2, 3 and 4), are also reported. Thus, the time indicators are computed and the mean values are reported by the follows in Table 4.4 and Table 4.5 to prove the efficiency of our incremental proposals.

Table 4.3: Clustering accuracy (%)(BKM vs $IKBKM$)(Uncertain case)

Databases	Level 1			Level 2			Level 3			Level 4			Mean		
	BKM	$IKBKM$		BKM	$IKBKM$		BKM	$IKBKM$		BKM	$IKBKM$		BKM	$IKBKM$	
		Min	Max		Min	Max		Min	Max		Min	Max		Min	Max
BCW	84.63	90.09	87.31	82.19	88.55	85.39	80.98	86.66	84.12	79.15	84.16	82.17	81.74	87.37	84.80
CE	83.37	94.25	88.83	82.72	90.95	86.78	81.22	89.11	84.96	79.35	86.51	82.08	82.16	90.20	85.66
CVR	84.96	92.71	88.73	83.26	89.70	86.46	81.64	87.62	84.54	80.44	84.55	82.13	82.83	88.65	85.46
HR	84.51	90.49	87.35	81.97	90.29	85.54	80.32	87.19	83.45	78.93	83.86	81.25	81.43	88.71	84.39
LS	84.22	89.91	87.24	81.40	88.25	84.83	79.17	86.50	83.04	77.11	83.71	80.37	80.48	87.09	83.87
MM	83.81	88.12	87.10	80.00	88.09	84.61	77.69	86.47	82.03	75.31	83.04	78.49	79.20	86.93	83.06
NY	82.94	88.05	86.65	79.75	87.61	83.86	78.28	86.16	81.67	76.22	83.04	79.47	79.30	86.46	82.91
SH	81.95	89.46	84.59	79.22	86.53	82.12	77.62	83.70	80.19	75.96	80.77	78.53	78.69	85.12	81.36
TE	81.31	88.86	85.32	78.19	86.35	82.47	76.29	84.77	80.71	75.05	81.74	78.11	77.71	85.43	81.65
ZO	84.15	89.02	86.89	81.83	87.46	85.22	79.83	85.89	83.36	78.03	83.55	80.85	80.96	86.48	84.08

Table 4.4: Running time(s)(BKM vs $IKBKM$)(Uncertain case)

Databases	Level 1			Level 2			Level 3			Level 4			Mean		
	BKM	$IKBKM$		BKM	$IKBKM$		BKM	$IKBKM$		BKM	$IKBKM$		BKM	$IKBKM$	
		Min	Max		Min	Max		Min	Max		Min	Max		Min	Max
BCW	211	188	194	296	284	281	333	321	341	440	408	411	320	300	307
CE	358	321	339	389	374	381	458	421	431	602	533	586	452	412	434
CVR	263	234	248	294	271	283	351	342	368	411	364	428	330	303	332
HR	164	143	158	193	179	176	240	231	258	314	234	238	228	197	208
LS	54	49	51	56	49	52	59	57	56	85	73	77	64	57	59
MM	406	399	407	509	458	467	598	526	548	737	587	814	563	493	559
NY	461	411	424	521	508	511	624	608	611	821	797	832	607	581	595
SH	197	186	189	330	296	294	418	385	379	585	465	509	383	333	343
TE	296	271	284	231	258	220	327	312	328	474	392	466	332	308	325
ZO	197	172	181	248	227	239	271	250	262	415	373	387	283	256	267

Table 4.5: Iteration number (BKM vs $IKBKM$)(Uncertain case)

Databases	Level 1			Level 2			Level 3			Level 4			Mean		
	BKM	$IKBKM$		BKM	$IKBKM$		BKM	$IKBKM$		BKM	$IKBKM$		BKM	$IKBKM$	
		Min	Max		Min	Max		Min	Max		Min	Max		Min	Max
BCW	6	5	6	7	6	6	9	7	7	10	8	8	8	7	7
CE	10	8	9	12	9	10	14	12	13	15	12	12	13	10	11
CVR	7	5	5	9	6	7	12	9	9	13	10	11	10	8	8
HR	9	7	8	11	8	9	13	10	11	13	10	11	12	9	10
LS	6	5	5	7	5	6	9	8	8	11	10	11	8	7	8
MM	9	8	9	11	10	12	14	12	14	15	15	16	12	11	13
NY	12	12	13	14	13	14	15	14	14	14	11	12	14	13	13
SH	8	6	7	10	8	7	12	12	11	14	12	13	11	10	10
TE	9	6	7	11	8	9	13	10	11	11	9	10	11	8	9
ZO	6	4	6	8	6	7	10	8	9	12	7	8	9	6	8

As expected, the performances of $IKBKM_{Min}$ and $IKBKM_{Max}$ are comparable with better clustering accuracies for $IKBKM_{Min}$ version.
Nevertheless, these $IKBKM$ approaches provide better precisions than the static version namely BKM as tabulated in these tables.

By Table 4.3, we expose the mean values of different databases results and for all three tested methods (static versus dynamic ones). This is based on the already detailed experimentations. These average accuracies confirm the detailed one. Indeed, as example, the mean accuracy for Breast Cancer Wisconsin database goes from 81.74% in static context to 87.37% with $IKBKM_{Min}$ and 84.80% with $IKBKM_{Max}$ for dynamic framework.

We can also conclude that the accuracy decreases when the uncertainty increases (from Level 1 to Level 4). It is true for all databases and for all classifiers (static and dynamic ones).

Notice also that $IKBKM$ time performance outperforms BKM as given by Table 4.4 and Table 4.5. Indeed, the results of the two incremental approaches are comparable, with a better performance of the $IKBKM_{Min}$ approach in terms of time indicators which exceed the $IKBKM_{Max}$ ones, as shown by the previous Tables (4.4 and 4.5). Thus, $IKBKM_{Min}$ version surpasses the $IKBKM_{Max}$ counter-part as well as the static one. For example for Breast Cancer Wisconsin database, the mean running time goes from 300 seconds with $IKBKM_{Min}$ to 307 seconds using $IKBKM_{Max}$ and 320 seconds with BKM approach. It is the same interpretation for the variability of the iteration number (8, 6 and 7 respectively to BKM, $IKBKM_{Min}$ and $IKBKM_{Max}$).

By the following figures, we confirm graphically the comparative results of both static and dynamic frameworks. The represented values are the average ones of the three tested criteria.

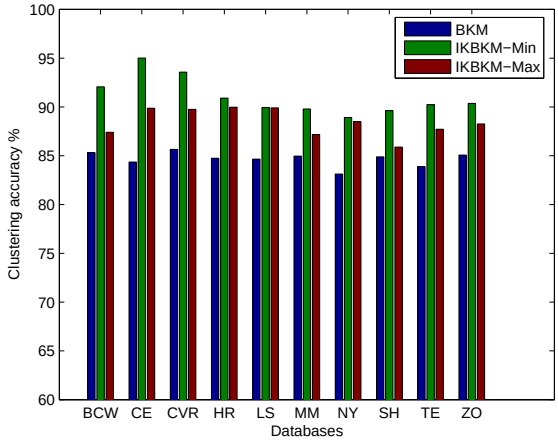
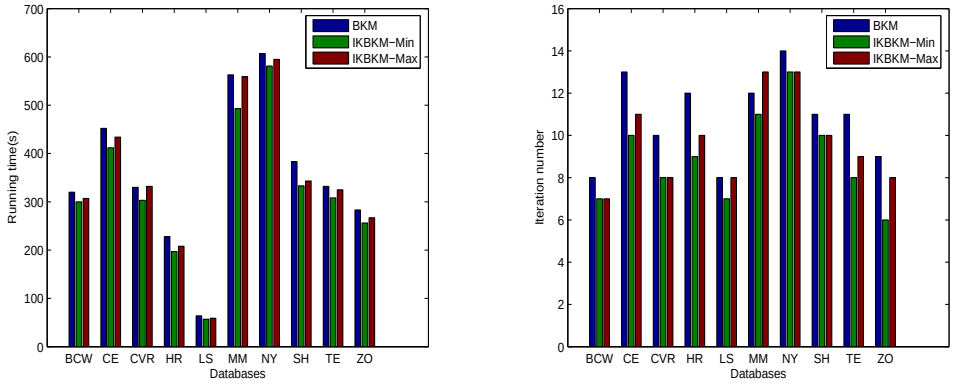


Figure 4.1: BKM vs $IKBKM$ clustering accuracy results under uncertainty (Mean)



(a) Running time(s) (b) Iteration number

Figure 4.2: BKM vs $IKBKM$ time indicators results under uncertainty (Mean)

$DKBKM$ simulations

We move now to the evaluation of the second familly of dynamic approaches which is relative to the decremental process. The same indicators will be reported in order to judge the efficiency of our proposals. By the following, we describe the different $DKBKM$ results respectively to both $DKBKM_{Min}$ and $DKBKM_{Max}$ versions by considering certain and uncertain contexts. The BKM results are also reported below.

1. Certain case: By the following, the attained results in such certain context will be presented. Thus, accuracy and time indicators are computed for static and dynamic methods.

Clustering accuracy results Firstly, the clustering accuracy criterion is computed for some actual databases. The results are tabulated in the below table.

Table 4.6: Clustering accuracy (%)(BKM vs $DKBKM$)(Certain case)

Databases	BKM	$DKBKM$	
		$DKBKM_{Min}$	$DKBKM_{Max}$
Breast Cancer Wisconsin (BCW)	85.30	87.00	91.41
Car evaluation (CE)	84.53	89.88	92.40
Congressional voting records (CVR)	85.65	89.79	91.78
Hayes-Roth (HR)	84.75	88.95	90.98
Lenses (LS)	84.65	88.64	91.93
Mushroom (MM)	84.96	87.74	90.19
Nursery (NY)	83.12	87.25	88.87
Spect heart (SH)	84.88	86.85	90.81
Tic-Tac-Toe Endgame (TE)	83.88	86.93	89.95
Zoo (ZO)	85.07	88.46	90.37

Time results The running time and the necessary number of iterations to update the cluster collection respectively to the three tested methods are reported in Table 4.7.

Table 4.7: Time performances (BKM versus $DKBKM$)(Certain case)

Databases	BKM		$DKBKM_{Min}$		$DKBKM_{Max}$	
	Time(s)	Iter	Time(s)	Iter	Time(s)	Iter
Breast Cancer Wisconsin (BCW)	205	6	202	5	187	5
Car evaluation (CE)	347	9	323	9	309	8
Congressional voting records (CVR)	218	6	207	6	199	5
Hayes-Roth (HR)	148	8	132	7	124	6
Lenses (LS)	48	5	47	5	38	4
Mushroom (MM)	383	7	386	8	379	7
Nursery (NY)	407	11	384	8	379	7
Spect heart (SH)	196	6	207	7	193	6
Tic-Tac-Toe Endgame (TE)	208	7	203	6	214	6
Zoo (ZO)	174	6	158	5	147	5

From Table 4.6, we find that $DKBKM_{Max}$ is more accurate approach than $DKBKM_{Min}$ and BKM ones, within this certain framework. For example, the mean accuracy induced from Nursery database using $DKBKM_{Max}$ is 88.87% and 85.25% using $DKBKM_{Min}$, while BKM generates only 82.12% as average value.

Faster results are also attained when the $DKBKM_{Max}$ is used for almost all bases. Indeed, the application of this dynamic approach decreases both running time and the number of iterations. So, BKM needs more cost calculation than both $DKBKM_{Min}$ and $DKBKM_{Max}$ classifiers. Indeed, $DKBKM_{Max}$ is the best one for almost all databases. As example, the running time of

Nursery database goes from 407 seconds with BKM classifier to 379 seconds using $DKBKM_{Max}$ one. An intermediate running time is 384 seconds with $DKBKM_{Min}$ method. The iteration numbers are 11, 7 and 8 respectively.

Based on the Tables 4.6 and 4.7, we confirm that both $DKBKM$ dynamic proposals generally surpass the static method in terms of accuracy and time performances.

2. Uncertain case:
   The following tables detail the uncertainty results for the four uncertainty levels. Thereafter, their summary will be also reported. This latter corresponds to the mean values computation of all three evaluation criteria (accuracy, running time and iteration number).

   Clustering accuracy results Firstly, Table 4.8 is presented to detail the clustering accuracy values for all used databases. The summary of these experimental results will be also reported in the same table.

   Time results It is the same as the incremental context, here, the mean values of time indicators will be those reported in addition to the detailed ones. All obtained results are tabulated in the following.

Table 4.8: Clustering accuracy (%)(BKM vs $DKBKM$)(Uncertain case)

Databases	Level 1			Level 2			Level 3			Level 4			Mean		
	BKM	$DKBKM$		BKM	$DKBKM$		BKM	$DKBKM$		BKM	$DKBKM$		BKM	$DKBKM$	
		Min	Max		Min	Max		Min	Max		Min	Max		Min	Max
BCW	84.63	86.28	90.01	82.19	85.41	87.96	80.98	83.76	86.50	79.15	81.71	84.53	81.74	84.54	87.25
CE	84.17	88.72	90.53	82.72	87.65	90.44	81.22	85.69	89.51	79.35	82.92	86.44	82.16	86.49	90.48
CVR	84.96	88.02	90.79	83.26	86.39	89.82	81.64	84.83	87.91	80.44	82.44	85.45	82.83	85.42	88.99
HR	83.51	87.11	90.55	81.97	85.46	88.76	80.32	83.35	87.07	78.93	80.92	83.93	81.43	84.21	88.07
LS	82.22	87.74	90.25	81.40	85.20	88.68	79.17	83.17	87.02	77.11	80.72	84.81	80.48	84.21	87.69
MM	83.81	86.43	89.69	80.00	83.66	87.52	77.69	81.55	85.68	75.31	78.51	82.26	79.20	82.54	86.29
NY	82.94	86.83	88.01	79.75	83.69	87.31	78.28	81.73	85.46	76.21	79.30	82.27	79.30	82.89	86.01
SH	81.95	85.51	88.81	79.22	82.85	86.59	77.63	80.81	84.16	75.96	78.46	81.81	78.69	81.91	85.35
TE	81.31	85.87	89.04	78.19	83.23	86.68	76.29	80.43	84.80	75.05	77.77	81.55	77.71	81.82	85.52
ZO	84.15	86.17	88.83	81.83	84.02	87.01	79.83	82.44	85.36	78.03	80.59	83.35	80.96	83.31	86.14

Table 4.9: Running time(s)(BKM vs $DKBKM$)(Uncertain case)

Databases	Level 1			Level 2			Level 3			Level 4			Mean		
	BKM	$DKBKM$		BKM	$DKBKM$		BKM	$DKBKM$		BKM	$DKBKM$		BKM	$DKBKM$	
		Min	Max		Min	Max		Min	Max		Min	Max		Min	Max
BCW	211	208	191	352	308	278	404	322	322	454	382	366	355	305	289
CE	358	339	321	457	398	384	512	427	411	541	483	421	467	412	384
CVR	263	252	233	311	277	227	384	309	299	399	323	302	339	290	265
HR	164	161	145	199	182	164	253	215	189	288	252	199	226	202	174
LS	54	49	41	64	51	48	81	76	72	92	83	74	73	65	59
MM	461	429	411	503	489	443	594	513	504	603	541	517	540	493	469
NY	427	394	386	597	538	521	733	633	618	741	638	624	624	541	537
SH	241	274	256	308	288	262	313	299	276	384	374	362	312	309	289
TE	335	280	266	348	289	276	414	391	341	442	402	388	381	338	308
ZO	233	184	163	248	254	239	317	299	253	323	302	289	280	260	236

Table 4.10: Iteration number (BKM vs $DKBKM$)(Uncertain case)

Databases	Level 1			Level 2			Level 3			Level 4			Mean		
	BKM	$DKBKM$		BKM	$DKBKM$		BKM	$DKBKM$		BKM	$DKBKM$		BKM	$DKBKM$	
		Min	Max		Min	Max		Min	Max		Min	Max		Min	Max
BCW	6	5	6	8	8	7	9	8	7	10	9	8	8	7	7
CE	10	9	9	14	11	11	14	12	12	15	12	12	13	11	11
CVR	7	6	5	10	7	5	11	8	7	12	9	8	10	7	6
HR	9	8	8	12	10	9	14	12	10	15	13	10	12	11	9
LS	6	6	5	7	4	6	8	6	6	8	7	6	7	6	6
MM	10	11	9	11	11	9	13	12	10	13	13	10	11	12	9
NY	12	9	8	14	12	12	17	16	15	17	16	15	15	13	12
SH	10	8	7	13	10	8	13	11	9	13	12	11	12	10	10
TE	11	7	7	11	7	7	12	10	9	13	11	10	12	9	8
ZO	8	5	5	8	6	5	10	8	7	10	9	7	9	7	6

According to the results under uncertainty, in previous tables, both decremental versions provide high accuracy values for all testing data where all values are greater than 80%. Obviously, it is indicative of the efficacy of such developed dynamic approaches. We observe that, in most cases, $DKBKM$ accuracies are slightly higher than for BKM on one hand. On the other hand, $DKBKM_{Max}$ results are better than those giving by $DKBKM_{Min}$ for all databases. As example, let us notice Zoo database for which the mean accuracy goes from 80.96% with BKM static method to 83.31% with $DKBKM_{Min}$ and 86.14% with $DKBKM_{Max}$.

Furthermore, $DKBKM_{Max}$ provides better running time values for all testing data as showing by Table 4.9. It is shown that our dynamic procedures are faster to reach the final cluster partition than static approach. The mean running time by using $DKBKM_{Max}$ is 236 seconds for Zoo database, while it is 280 seconds with BKM method and 260 seconds with $DKBKM_{Min}$.

The reported iteration number by applying $DKBKM_{Max}$ (see Table 4.10) are smaller than the BKM's ones, as well as the $DKBKM_{Min}$ ones, for almost all data sets. This improves the efficiency of these developed decremental methods besides their effectiveness quality more precesily $DKBKM_{Max}$. $DKBKM_{Min}$ provides also better results than BKM method. This latter needs great number of iterations to reach the stability of the solution. This number goes from 9 to 6 for Zoo base.

Remark that from Level 1 to Level 4, meaning the uncertainty increase, the clustering accuracy decreases, while the time indicator values increase. This is general for all tested databases and by applying any method among BKM, $DKBKM_{Min}$ or $DKBKM_{Max}$.

All these concluding remarks respectively to these accuracy and time indicator results within such uncertain context, are more shown by Figures 4.3 and 4.4.

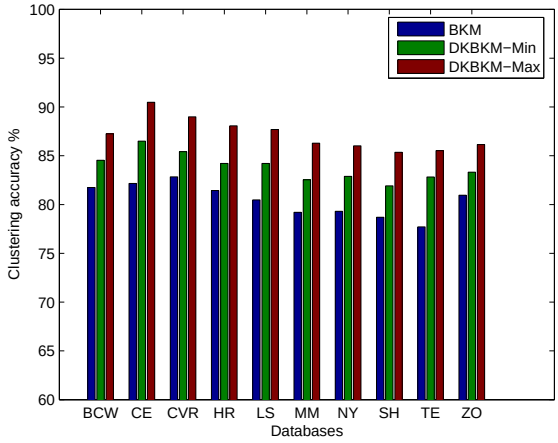
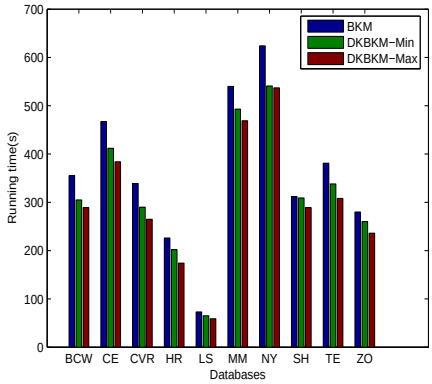
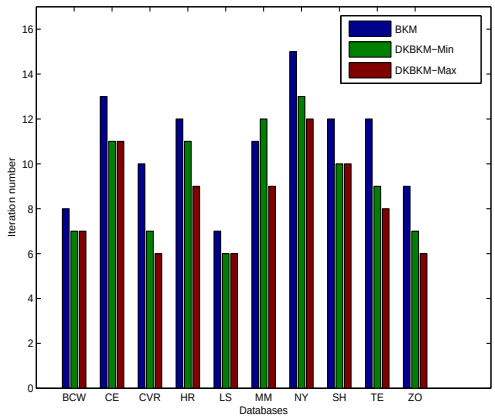


Figure 4.3: BKM vs $DKBKM$ clustering accuracy results under uncertainty (Mean)



(a) Running time(s)



(b) Iteration number

Figure 4.4: BKM vs $DKBKM$ time indicators results under uncertainty (Mean)

4.5.3 Experimental summary

The success of all our developed methods, incremental as well as decremental, is demonstrated on real life benchmark data sets. The advantages of such dynamic approaches are the better accuracies of the attained collections and the speed of convergence expressed by the iteration number and the running time which are significantly decreases compared to static method.

Based on clustering accuracy criterion and time complexity as well as number of iterations results, we draw the conclusion that our developed methods are more

appropriate than BKM to use in order to update the cluster partition when its cardinality increases or decreases by one under uncertainty.

It is also observed that the $IKBKM_{Min}$ version outperforms the $IKBKM_{Max}$ one for the incremental proposals, while for the decremental procedures the $DKBKM_{Max}$ surpasses the $DKBKM_{Min}$ approach.

All these developed dynamic approaches under uncertainty are summarized in (Ben-Hariz and Elouedi, 2014).

4.6 Conclusion

In this chapter, we have presented a novel dynamic belief clustering framework called Belief Clustering for Dynamic Partition ($BCDP$) by extending the uncertain K-modes method version (BKM) to a non-static environment. It consists in dynamically increment (IKBKM) or decrement (DKBKM) the resulting BKM clusters or partition to guarantee better partitions, faster, without performing the reclustering again. This is done by considering the similarity/dissimilarity measures concept. We have developed two versions relative to incremental and decremental processes by considering, firstly, the intra-cluster dissimilarity minimization and then, the inter-cluster distance maximization as optimization problems.

The attribute set characterizing database objects may also change over time besides their uncertainty characteristic. Investigating this idea in more details and developing methods suitable for analyzing such data sets is an important topic of ongoing work. Thus, in the next chapter, we will study this dynamic aspect.

5

Belief clustering methods for dynamic data set features

5.1 Introduction

The current information explosion has led to the generation of an ever-increasing volume of data. Moreover, the real databases may have frequent updates, this implies that existing clustering algorithms are not suitable for maintaining clusters in such dynamic environment, and they have been struggling with the problem of updating clusters without frequently performing complete reclustering. In addition to, the already mentioned environment updates related to the cluster number, in the previous chapter, we will present, here, another kind of update which can be encountered within such non static environment namely the dynamic attribute or feature set.

In fact, some of the already used features may be redundant, some may be irrelevant and some can even misguide clustering results. Then, reducing their number may increase comprehensibility and ameliorate the problem that some unsupervised learning algorithms break down for high dimensional data. On the other hand, another possibility can be treated within this same context namely the detection of new attribute(s), judged as relevant, to consider for the clustering process. As illustrative example in medicine, the decision of expert to incorporate new patients' symptoms, considered as relevant, during the disease classification problem. Elimination of some, judged as irrelevant, may also be mentioned.

Indeed, there are few approaches reported in the literature that take into account the problem of adapting the result of a clustering when the object feature set is extended or reduced. This issue is more studied under supervised classification. Within unsupervised classification, the dynamic aspect is general related to the data or the clusters, while the feature set is considered known apriori. However, in (Serban

and Câmpan, 2005; Câmpan and Serban, 2006), authors have developed adaptive extensions for two traditional clustering algorithms (k-means and Hierarchical Agglomerative Clustering). This work concerns only the increasing case. There is also the possibility to treat the sequential use of these features in the clustering process, one by one. An example of such approach is mentioned in (Jain et al., 1999).
However, the decreasing case has not been reported by none of literature works.
Thus, the development of clustering methods which are able to ensure their task in both incremental and decremental contexts must be done. It is the aim of our proposals presented later under uncertainty.

In this chapter, we will expose more this problem in Section 2. Then, the basic concepts of the studied framework will be defined. A new feature selection method under uncertainty is also proposed in the following section. A part of this same section will be dedicated to the developed incremental and decremental belief clustering approaches. Finally, we report the different attained results in order to show the effectiveness of our dynamic framework in comparison with the static one.

5.2 Problem presentation

We propose a novel dynamic belief clustering framework called Belief Clustering for Dynamic Data set Features ($BCDDF$) by extending the uncertain K-modes method version (BKM) to non-static environment more precisely to non-stationary feature set. This is the first subfamily of Belief Clustering for Dynamic Data set Dimensions $BCDDD$ framework. Indeed, the first dimensions handled by this thesis work is the attribute set, while the second one will be the object set. However, both incremental as well as decremental procedures will be detailed. They are denoted Incremental S Belief K-modes method ($ISBKM$) and decremental S belief K-modes approach ($DSBKM$). Note that S represents the initial attribute number to increase or to reduce respectively to these both proposed versions.

The objective is to incorporate a newly arrived information into the existing model without reclustering all altered objects, since there are several applications where the attribute set characterizing the objects evolves.

The same real application presented in Chapter 4 to illustrate the motivation of our dynamic proposals may be considered here. Indeed, the household segments obtained on the basis of for example age, income, gender and number of children must be updated when the marketing expert would like to incorporate other information as instance number of purchases besides the already handled attributes. An

already considered characteristic during the household segmentation task, may be then suppressed seen that it is judged as irrelevant one. As example, let us mention the nationality. Therefore, an updating of the partition of these households must be conducted. Besides, the uncertainty may pervade any attribute characterizing these households (original, extended or reduced attributes).

Within non-incremental clustering approaches, all objects must be present, with a known attribute set, and this before beginning the classification process. It is a necessary condition for starting the clustering task. Indeed, an already developed BKM method (Ben-Hariz et al., 2008) starts with fixed classification parameters. This latter deals with the uncertainty problem.

Due to the increasing of data volume and their dynamic nature, classical clustering algorithms are limited in their use within such environment for both cases certain as well as uncertain one. However, the proposed approaches, here, must be able to re-partition the objects set in a dynamic environment in addition to the uncertainty management by adopting BKM approach.

The two main procedures developed to ensure this information integration to cluster structure are as follows:

1. Incremental process: produces an updated partition of extended objects.
2. Decremental process: returns an altered partition of reduced objects.

An extended object means an object characterized by A^+ attributes, whereas a reduced one is characterized by only A^- features.
These two methods will be exposed below. To ensure their task, these approaches study in which conditions an extended/a reduced object is still correctly placed in its current cluster. To conduct this dynamic update, firstly, the selection of the attribute(s) to add or to eliminate must be performed when the expert does not provide this information.

This studied problem can be formally defined as follows:

- Input: $P = \{C_1, C_2, .., C_l, .., C_k\}$.
  Each cluster C_l is one group represented via its mode Q_l in a s-dimensional feature space under uncertainty.
- Output: P_+ or P_-.
  These cluster collections are characterized by respectively A^+ (of s^+ attributes)

and A^- (of s^- attributes) set. They result from incremental or decremental procedures.

5.3 Necessary concepts

5.3.1 General assumptions

Based on the distance concept, the following dynamic proposals will try to update an already attained collection of k groups. This is ensured by incorporating or eliminating one or more features which characterize the data set observations.
Recall that these latter values are known with uncertainty and thus, represented and managed via belief function concepts, more precisely the *bba* one.

Different cases must be distinguished namely:

- For incremental context:
  - The expert mentions exactly which feature(s) must be incorporated.
    → Obviously, for this case, the feature selection process is not useful.
  - Only the number of features to insert is known and it is provided by the expert.
    →For this alternative, the feature selection must be conducted to decide exactly which one(s) must be integrated to our initial cluster collection.
- For decremental context:
  - Expert mentions which feature(s) to suppress.
    →No feature selection step for this case.
  - One or more features must be eliminated as mentioned by the expert, but only their number is provided.
    →Firstly, the feature selection preprocessing step is made.

All these issues will be more handled later. Indeed, the detailed algorithms presented in the following subsections, may take into account all the above assumptions.

5.3.2 Basic notions

The following adapted notions will be used under the $BCDDF$ context.

- The distance of an attribute A_j measuring its impact:

- Within one uncertain data set UT of n instances:
  It computes the distance measure between all pairs of objects by considering only this feature. It is computed by:

$$d_{j,UT} = \frac{1}{2} \sum_{i1=1}^{n} \sum_{i2=1, i2 \neq i1}^{n} d(x_{i1,j}, x_{i2,j}) \tag{5.1}$$

  This is an indicator of the data scatter.
- Within one cluster C_l of n_l objects:
  It computes the dissimilarity measure between the cluster mode and all its instances respectively to this feature. Its definition is as follows:

$$d_{j,l} = \sum_{i=1}^{n_l} d(x_{i,j}, q_{l,j}) \tag{5.2}$$

  Note that it measures the cluster cohesion but by considering only one attribute. Thus, it reflects the impact of this feature on the compactness of this group.
  To determine its impact respectively to the partition P of these k clusters, the sum must be computed as: $d_{j,P}{=}\sum_{l=1}^{k} \mu_l d_{j,l}$, where μ_l reflects the weight of the corresponding cluster C_l. This must be specified by the expert. For our study, all clusters have the same weight, thus $\forall\ l \in \{1, .., k\}$, $\mu_l = 1$.

- The revised set R:
  It consists of items which must be revised, in terms of membership, following the extension or reduction of attribute set characterizing our data set to cluster.

All advanced distance measures must be defined under uncertainty as already detailed in Chapter 3 (Equation 3.4). Moreover, the d component is computed based on (Jousselme et al., 2001).

5.4 New clustering proposals dealing with dynamic feature set under uncertainty

5.4.1 Feature selection method within belief function framework

The proposed feature selection approach principle depends on the dynamic context to consider. If the incremental framework is the one to study, indeed, the attribute(s) to introduce has not yet participated at the cluster construction. However, if the decremental procedure is analyzed, then the clusters, already produced, take into account the attribute(s) to eliminate which will be selected via our proposal.

Generally, feature selection algorithms can be categorized as either filter or wrapper (John et al., 1994) approaches. Filter approaches preselect the features, and then apply the selected feature subset to the clustering algorithm, whereas, the wrapper ones incorporate the clustering algorithm in the selection feature process.

Within our dynamic system, there are two ways to process the feature selection. Independently of the clustering algorithm for the incremental context and based on the cluster collection when the decremental one is considered. Our proposal can be considered as filter approach within incremental framework, while as wrapper one for decremental context.

The exhaustive research to detect the most irrelevant attribute(s) to eliminate or the most relevant one(s) to introduce is an expensive solution which is only practical for a few attribute number. In fact, this leads to explore a space of all possibilities which is judged as impractical and costly computation. Another problem of selection attribute, here, is the uncertainty parameter.

The solution to these problems is to apply an appropriate heuristic method. Thus, in this part of our work, we propose an heuristic method of feature selection based on the belief function theory that is, on one hand, able to compute the uncertain attribute(s) score(s) expressed by its/their dissimilarity measure. To indicate, on the other hand, the feature(s) to select following their generated ranking.

The aim is to introduce or reduce by one or more attributes the initial feature set. When this extension or reduction is directly fixed by one expert, no selection step is required. It is the simplest case. Thus, the incremental or the decremental methods, developed below, will be applied directly. Otherwise, a pre-processing selection must be done.

This pre-processing selection is a two-stage procedure. Thus, firstly, it computes all feature scores based on the dissimilarity measure. Results of this first stage are used in the second one in order to rank, select and preserve only the most relevant/irrelevant features. We will detail the two feature selection phases under uncertainty, namely the local feature score computation process ($LFSC$) and the feature ranking versus feature selection step ($FRvsFS$), as follows.

Local feature score computation step

Let us assume that the k clusters, resulting by applying BKM clustering method, or all data set objects constitute the starting point of the whole selection approach.

To decide of the features ranking (preference order), the dissimilarity measure is proposed as core concept. Then for each attribute, we have to compute its dissimilarity measure. This distance measures its impact on the intra-cluster quality/compactness or on the data set scatter.

In the first case, the attribute having the highest distance value represents the one who participated least in the cluster collection formation. Thus, it is judged as the most irrelevant one, while the second case is the opposite. Here, the attribute with the lowest measure is the most unimportant one.

All these distance measures, computed within this first step, will be used as input parameters for the second step in order to decide which attribute(s) to add or to eliminate. This is done following their ranking.

From feature ranking to feature selection step

The notion of preference has a central role in our study. Indeed, the strict preference and the indifference are the fundamental used comparative concepts. An attribute can be preferred to another one meaning that it is most important than the other one, if and only if it has participated more in cluster's construction or it will be more relevant for the data set segmentation. This is done by comparing the scores of these two attributes respectively to an appropriate metric. If they are judged indifferent, it means that both have participated or will participate equitably in cluster's formation. Indeed, the preference is expressed via $\succ$, whereas, the indifference is represented by $\approx$.

By using the input distances respectively to all initial s attributes, their ranking is directly built based on these distances sort in an ascending or descending order.
For two attributes A_{j_1} and A_{j_2}, where $j_1, j_2 \in \{1, .., s\}$, we define these two relations:
- The preference: $A_{j_1} \succ A_{j_2}$ iff $d_{j_1,P} < d_{j_2,P}$ (decremental case) or $d_{j_1,UT} > d_{j_2,UT}$ (incremental case).
- The indifference: $A_{j_1} \approx A_{j_2}$ iff $d_{j_1,P} = d_{j_2,P}$ (decremental case) or $d_{j_1,UT} = d_{j_2,UT}$ (incremental case).

Giving an arrangement respectively to considered attributes, only the most impor-

tant features must be preserved if the decremental case is handled. The attribute(s) which is/are judged as the most irrelevant one(s), via the general feature ranking process, must be eliminated. Thus, a feature ranking is generated respectively to these scores by their sort in an increasing order. We assume that all clusters have the same impact for this step and their corresponding weights are all equal to one $\mu_l = 1$, $l \in \{1, .., k\}$. (see Section 5.4).

Another case must be mentioned, respectively to the incremental process, for which the descending order is the appropriate one to consider in order to generate the feature ranking.

Let us mention that two attributes or more, having the same score, can be selected as the most irrelevant/relevant ones and thus must be eliminated/added.

The belief feature selection algorithm can be described by the following pseudo code:

Belief Feature Selection algorithm (UT, P, nb_1)

Data: UT, P, nb_1
//Two cases arise $nb_1 = 1$ (incremental process based on the dataset) or $nb_1 = k$ (decremental framework based on the k clusters).
Result: $index(es)$
begin
1. Local feature score computation.
2. Feature ranking versus feature selection.

end

In order to detail more the developed selection feature approach under uncertainty, denoted BFS, the following pseudo algorithms of the two phases, namely $LFSC$ and $FRvsFS$, are proposed.

Local feature score computation algorithm(UT, P, nb_1)

```
Data: UT, P, nb_1
Result: D_{j,P} or D_{j,UT} // ∀j ∈ {1,..s}
begin
    if nb_1 = k then
        for Each attribute A_j, 1 ≤ j ≤ s do
            D_{j,P} ← 0
            for Each cluster C_l, 1 ≤ l ≤ k do
                Compute D_{j,l} by Equation 5.2
                D_{j,P} ← D_{j,P} + D_{j,l}
    else
        for Each attribute A_j, 1 ≤ j ≤ s do
            Compute D_{j,UT} by Equation 5.1
end
```

For the incremental case ($nb_1 = 1$), UT input parameter contains the same observations which are already clustered statically by applying BKM. But these items are, now, characterized by other feature set. It consists in their extension.

Feature ranking versus feature selection algorithm($D_{j,UT}$, $D_{j,P}$)

```
Data: D_{j,UT}, D_{j,P} // ∀j ∈ {1,..s}
Result: index(es)
begin
    if ∀D_{j,UT} = 0 then
        D_j ← D_{j,P}, with j ∈ {1,..,s}.
        Sort these distances in an ascending order.
        Generate directly the ranking R_s of s attributes.
        Preserve the index(es) of the last attribute(s) which is/are judged as the most
        irrelevant one(s) (the least preferred one(s)).
    else
        if ∀D_{j,P} = 0 then
            D_j ← D_{j,UT}, with j ∈ {1,..,s}.
            Sort these distances in a descending order.
            Generate directly the ranking R_s of s attributes.
            Preserve the index(es) of the first attribute(s) which is/are judged as the
            most relevant one(s) (the most preferred one(s)).
end
```

The selection belief feature process consists in the execution of the two already exposed algorithms to return finally the index(es) of the most irrelevant features to eliminate or the preferred one(s) to introduce. Recall that the considered feature set contains uncertain attribute values represented via the belief function theory concepts.

5.4.2 Dynamic belief clustering for handling incremental feature set

Definition

Given an extended data set, with one new attribute, it must be measured with respect to old s attributes as well as this new arrival. After this extension, any object X_i becomes X_i^+ characterized by s^+ attributes, where $s^+ > s$. Firstly, we will consider that $s^+ = s+1$. We try to incorporate this new attribute in the current partition, initially obtained by applying BKM approach, and reclustering all extended data set items starting from this current collection using our proposal namely $ISBKM$ method. As a final result, we have to classify each extended object in the closer cluster based on its distance to the corresponding extended mode. Thus, an updated partition P_+ will be attained.

To this end, starting with the current partition, we will study in which conditions an extended object X_i^+ is still correctly placed in its current cluster C_r^+ (with corresponding mode Q_r^+), which is judged the closer one before this attribute insertion. This is ensured via the checking of the following proposition. Then, the main idea is to divide our data set objects, already assigned to k groups, into two sets those which remain unchanged and those which must be reallocated to one revised set defined by subsection 5.3.2 and Equation 5.3.

In this subsection, the handled issue is in which condition an extended observation remains stable in its current cluster?

Proposition 5.1 *Given:* $d(m^{\Theta_{s+1}}\{X_i^+\}, m^{\Theta_{s+1}}\{Q_r^+\}) \leq d(m^{\Theta_{s+1}}\{X_i^+\}, m^{\Theta_{s+1}}\{Q_l^+\})$, *where* X_i^+ *is an extended object and* C_r^+ *is its current cluster, with* $1 \leq l \leq k$, $l \neq r$. *It holds that:* $D(X_i^+, Q_r^+) \leq D(X_i^+, Q_l^+)$

Thus, the object X_i^+ is closer to the mode Q_r^+ than to any other mode Q_l^+.
It is among those which remain unchanged. Otherwise, this object must be revised.

Proof

X_i^+ before its extension namely X_i is correctly classified by BKM. Thus, $D(X_i, Q_r) \leq D(X_i, Q_l)$, where $1 \leq l \leq k$, $l \neq r$. It means that this item is closer to its cluster than to any other cluster.
Thereafter this extension, the distance of this item to its cluster and the other ones can be defined as follows:
$D(X_i^+, Q_r^+) - D(X_i^+, Q_l^+) = \sum_{j=1}^{s^+} d(m^{\Theta_j}\{X_i^+\}, m^{\Theta_j}\{Q_r^+\}) - \sum_{j=1}^{s^+} d(m^{\Theta_j}\{X_i^+\}, m^{\Theta_j}\{Q_l^+\})$

$D(X_i^+, Q_r^+) - D(X_i^+, Q_l^+) = (\sum_{j=1}^{s} d(m^{\Theta_j}\{X_i\}, m^{\Theta_j}\{Q_r\}) + d(m^{\Theta_{s+1}}\{X_i^+\}, m^{\Theta_{s+1}}\{Q_r^+\})) - (\sum_{j=1}^{s} d(m^{\Theta_j}\{X_i\}, m^{\Theta_j}\{Q_l\}) + d(m^{\Theta_{s+1}}\{X_i^+\}, m^{\Theta_{s+1}}\{Q_l^+\}))$

Since $\sum_{j=1}^{s} d(m^{\Theta_j}\{X_i\}, m^{\Theta_j}\{Q_r\}) = D(X_i, Q_r)$ and $\sum_{j=1}^{s} d(m^{\Theta_j}\{X_i\}, m^{\Theta_j}\{Q_l\}) = D(X_i, Q_l)$, Then:
$D(X_i^+, Q_r^+) - D(X_i^+, Q_l^+) = (D(X_i, Q_r) - D(X_i, Q_l)) + (d(m^{\Theta_{s+1}}\{X_i^+\}, m^{\Theta_{s+1}}\{Q_r^+\}) - d(m^{\Theta_{s+1}}\{X_i^+\}, m^{\Theta_{s+1}}\{Q_l^+\}))$

Giving: $D(X_i, Q_r) \leq D(X_i, Q_l)$ and
$d(m^{\Theta_{s+1}}\{X_i^+\}, m^{\Theta_{s+1}}\{Q_r^+\}) \leq d(m^{\Theta_{s+1}}\{X_i^+\}, m^{\Theta_{s+1}}\{Q_l^+\})$. *Then,*
$D(X_i^+, Q_r^+) - D(X_i^+, Q_l^+) \leq 0$

As the distance measures are non-negative, it results that: $D(X_i^+, Q_r^+) \leq D(X_i^+, Q_l^+)$.

We can conclude that one object remains, initially, unchanged if its distance to its current cluster's mode, respectively to the attribute to introduce, is judged as the smallest one compared to all other cluster modes.

Incremental Algorithm

We have to update, here, the initial partition facing the new information which consists in the addition of one attribute. By then, there are two steps: initialization and updating. The first stage consists in the cluster modes extension on one hand. On the other hand, the division of our data into two sets those which remain stable (within their clusters) and those which must be assigned to the revised set R. Then, after the reallocation of revised set instances to the existing groups, the updating phase is performed in order to reach the partition stability.

The proposed incremental algorithm has the following parameters:

Inputs:
$UT^+ = \{X_1^+, .., X_i^+, .., X_n^+\}$: the set of s^+-dimensional extended objects, with $s^+ > s$.
$P = \{C_1, .., C_l, .., C_k\}$: the output partition of the BKM method characterized by only s attributes.
$noMaxIter$: the maximum number of iterations allowed.
In fact, the algorithm might iterate for a long time with very small changes in the clusters, so we have consider this parameter as a stopping criterion.

Output:
The partition $P_+ = \{C_{1_+}, .., C_{l_+}, .., C_{k_+}\}$ of UT^+ objects.

A brief description of the $ISBKM$ algorithm is given below.

$ISBKM$ **algorithm** (UT^+, P, $noMaxIter$)

Data: UT^+, P, $noMaxIter$
Result: P_+
begin
Initialization phase
1. Determine all extended modes Q_l^+ after the attribute introduction.
2. $R = \emptyset$.
3. **for** *Each extended object* X_i^+ **do**
if *The Proposition 5.1 is checked* **then**
This object remains stable in its current cluster
else
This extended datum must be revised: $R = R \cup \{X_i^+\}$.
4. Update the cluster modes after these allocations.
5. The objects of the revised set should be assigned to appropriate clusters according to their distances.
Updating phase
6. Once all objects have been allocated to clusters, retest the dissimilarity of objects against the current modes.
if *An object is found such that its nearest mode corresponds to another cluster rather than its current one* **then**
7. Reallocate the object to that cluster.
8. Update the modes of both clusters (departure and arrival group).
9. Reiterate (go to step 6) until no object has changed clusters after a full cycle test of the whole data set or the noMaxIter is performed.
end

The revised set R within this incremental context is defined as follows:

$$R = \{X_i^+ |\ Proposition\ 5.1\ is\ invalid\} \tag{5.3}$$

Remark:
The input UT^+ parameter results from the fusion of the initial data set UT and the attribute(s) to insert. Indeed, the index/indices of these latter is/are detected by the already developed feature selection approach when the expert does not specify which ones must be incorporated. Thus, a simple pre-processing fusion step must be performed.

5.4.3 Decremental clustering to reduce feature set under uncertainty

Definition

It consists in dynamically decreasing the feature set of s attributes and updating the clustering result to guarantee a high quality partition. This is done by considering the similarity/dissimilarity measure concept.
This feature deletion concerns irrelevant ones. This occurs for redundant features. Indeed, when the first one provides the same information as the second one with regard to discriminating cluster, only one must be retained.
On the other hand, for example, we can consider one feature irrelevant if by omitting the other ones we have only one cluster, which is uninteresting result. Thus, this feature, with only one value, does not contribute to cluster discrimination.

We have, initially, to run the BKM algorithm to build k clusters characterized by all s uncertain attributes as point of departure of the developed decremental clustering process. This latter tries to obtain an updated cluster collection described via only the most relevant attributes.
Then, how to manage this partition to update it without reclustering our reduced data set again**?**

The key idea of our dynamic algorithm is the same considered one within the incremental context. It also uses the distance measure and the revised set concepts.

Proposition 5.2 *Let:* $d(m^{\Theta_{s-1}}\{X_i^-\}, m^{\Theta_{s-1}}\{Q_r^-\}) \geq d(m^{\Theta_{s-1}}\{X_i^-\}, m^{\Theta_{s-1}}\{Q_l^-\})$, *where* X_i^- *is a reduced object and* C_r^- *is its current cluster, with* $1 \leq l \leq k$, $l \neq r$.
Then: $D(X_i^-, Q_r^-) \leq D(X_i^-, Q_l^-)$.

Then, the object X_i^- is closer to the mode Q_r^- than to any other mode Q_l^-.
Thus, it is among those which remain unchanged. Otherwise, this object must be revised.

Proof

$D(X_i^-, Q_r^-) - D(X_i^-, Q_l^-) = \sum_{j=1}^{s^-} d(m^{\Theta_j}\{X_i^-\}, m^{\Theta_j}\{Q_r^-\}) - \sum_{j=1}^{s^-} d(m^{\Theta_j}\{X_i^-\}, m^{\Theta_j}\{Q_l^-\})$

$D(X_i^-, Q_r^-) - D(X_i^-, Q_l^-) = (\sum_{j=1}^{s} d(m^{\Theta_j}\{X_i\}, m^{\Theta_j}\{Q_r\}) - d(m^{\Theta_{s-1}}\{X_i^-\}, m^{\Theta_{s-1}}\{Q_r^-\})) - (\sum_{j=1}^{s} d(m^{\Theta_j}\{X_i\}, m^{\Theta_j}\{Q_l\})$-$d(m^{\Theta_{s-1}}\{X_i^-\}, m^{\Theta_{s-1}}\{Q_l^-\}))$

$D(X_i^-, Q_r^-) - D(X_i^+, Q_l^+) = (D(X_i, Q_r) - D(X_i, Q_l)) + (d(m^{\Theta_{s-1}}\{X_i^-\}, m^{\Theta_{s-1}}\{Q_l^-\}) - d(m^{\Theta_{s-1}}\{X_i^-\}, m^{\Theta_{s-1}}\{Q_r^-\}))$

Knowing that $D(X_i, Q_r) \leq D(X_i, Q_l)$, *where* $1 \leq l \leq k$, $l \neq r$, *Thus:*
$D(X_i^-, Q_r^-) - D(X_i^-, Q_l^-) \leq d(m^{\Theta_{s-1}}\{X_i^-\}, m^{\Theta_{s-1}}\{Q_l^-\}) - d(m^{\Theta_{s-1}}\{X_i^-\}, m^{\Theta_{s-1}}\{Q_r^-\})$

By considering the advanced proposition, then the inequality above becomes:
$D(X_i^+, Q_r^+) - D(X_i^+, Q_l^+) \leq 0$

As the distance measures are non-negative, it results that: $D(X_i^-, Q_r^-) \leq D(X_i^-, Q_l^-)$.

We can conclude that one object remains, initially, unchanged if its distance to its current cluster's mode, respectively to the attribute to eliminate, is judged as the highest one compared to all other cluster modes.

Decremental algorithm

Within this decremental context, it consists on updating the initial cluster collection when one or more attribute(s) are suppressed. Two steps will be conducted, namely the initialization and the updating, as within the incremental context.
The $DSBKM$ algorithm's inputs and outputs are defined as follows:

Inputs:
P: BKM final partition characterized by s attributes.
$noMaxIter$: the maximum number of iterations allowed.
UT: the uncertain data set already clustered.
$index(es)$: which are the indices of the attribute(s) to eliminate. Recall that this latter parameter is the belief feature selection method output. Indeed, contrary to

the incremental framework, here, the index(es) are directly used as parameter and no pre-processing step to eliminate their corresponding features is done.

Outputs:
The updated partition $P_{-} = \{C_1, C_2, .., C_k\}$ of UT^{-} objects.
UT^{-} the altered uncertain data set following its reduction (elimination of attributes having index(es) indices).

Our developed $DSBKM$ algorithm is summarized by the following pseudo-algorithm.

$DSBKM$ **algorithm** (UT, P, $noMaxIter$, $index(es)$)

Data: UT^{-}, P, $noMaxIter$, $index(es)$
Result: P_{-}, UT^{-}
begin
Initialization phase
1. Determine all reduced modes Q_l^{-} corresponding to each cluster $C_{l^{-}}$ after attribute(s) deletion having index(es) numbers.
2. $R = \emptyset$.
3. **for** *Each reduced object* X_i^{-} **do**
if *The Proposition 5.2 is checked* **then**
This object remains stable in its current cluster
else
This reduced observation must be revised: $R = R \cup \{X_i^{-}\}$.
4. Update the cluster modes after these allocations.
5. The objects of the revised set should be assigned to appropriate clusters according to their distances.
Updating phase
6. Once all objects have been allocated to clusters, retest the dissimilarity of objects against the current modes.
if *An object is found such that its nearest mode corresponds to another cluster rather than its current one* **then**
7. Reallocate the object to that cluster.
8. Update the modes of both clusters (departure and arrival group).
9. Reiterate (go to step 6) until no object has changed clusters after a full cycle test of the whole data set or the noMaxIter is performed.
end

Recall that the revised set R is a temporary group which is created and used during an intermediate step in order to assign to it the revised set objects.
The revised set concept is defined by the following formula within this decremental

environment:

$$R = \{X_i^- |\ Proposition\ 5.2\ is\ invalid\} \tag{5.4}$$

These proposed belief dynamic clustering approaches can be considered as probabilistic techniques when the attribute uncertainty is expressed via the bayesian belief function. Moreover, the standard certain database can be handled via certain bba's within $DSBKM$ framework. Thus, they offer more generalized approaches.

Remark:
The input UT^- parameter results from the reduction of the initial data set UT. Indeed, the index/indices of the eliminated attribute(s) is/are detected by the already developed feature selection approach when the expert does not specify which ones must be suppressed. Thus, a simpler eduction step must be performed.

5.5 Experimentations

A set of experiments has been conducted for testing these dynamic versions. The different results carried out from these tests will be presented and analyzed in order to evaluate our clustering methods in such uncertain and dynamic framework.

The experimental settings, which are detailed in Appendix A, will be considered to conduct these simulations and to advance an appropriate analysis.

The aim of conduct these experimentations consists in the evaluation of our dynamic proposals as well as their comparison study with the static version in terms of quality and time indicators under both certain and uncertain frameworks.

5.5.1 Experimental strategies

The evaluation of both incremental and decremental approaches developed, here, necessitates an initial partition as input parameter. This is the BKM output cluster collection. Respectively to each context, one strategy will be performed to conduct this evaluation task.

- For incremental procedure, initially BKM must be running by considering only $s-1$ attributes among all actual ones of the UCI data set to cluster. Indeed, this attribute is eliminated randomly. Thereafter, this non selected feature, above, will be provided to our incremental approach and the already attained partition must be updated consequently.
  The $ISBKM$ results will be compared to those of BKM static version by con-

sidering from the beginning all s attributes. No feature selection pre-processing step is done since only one known feature is incorporated.

- For the decremental method processes, differently, BKM generates, initially, k clusters characterized by $s+1$ attributes. Indeed, an attribute among s actual ones, choosing randomly, has been duplicated to attain $s+1$ features. Then, the decremental process alters this collection via the feature set decreasing by one attribute. This latter is the selected one by the feature selection method. Note that, it must be the already experimentally added attribute since it has, thus, two occurrences (redundant attribute).
  The $DSKBM$ outputs will be confronted to those of BKM by taking only s attributes at the beginning.

Indeed, we have simulated the dynamic aspect of these feature sets. Recall that s represents the actual number of features to increase or to decrease.

5.5.2 Empirical results and discussion

To evaluate the performance of our dynamic proposals, we compare firstly the obtained $ISBKM$ results with those of BKM classifier as static framework. Secondly, the $DSBKM$ versus BKM comparison will be made. To more illustrate this testing task, we will give for each category of evaluation criteria graphical presentations respectively to two dynamic frameworks.

$ISBKM$ simulations

1. Certain case:
   Testing certain databases is firstly done to evaluate our incremental proposal. The clustering quality indicator is the first evaluation criterion to compute. Thereafter, time criteria will be also reported.

   Clustering accuracy criterion The following table presents the clustering results in term of clustering accuracy from the application of both approaches (non-incremental method (BKM) and incremental one ($ISBKM$)).

Table 5.1: Clustering accuracy (%)(BKM vs $ISBKM$)(Certain case)

Databases	BKM	$ISBKM$
Breast Cancer Wisconsin (BCW)	85.27	90.86
Car evaluation (CE)	85.37	94.34
Congressional voting records (CVR)	85.96	92.57
Hayes-Roth (HR)	84.51	92.46
Lenses (LS)	84.22	89.66
Mushroom (MM)	83.81	89.00
Nursery (NY)	83.34	88.80
Spect heart (SH)	84.19	89.43
Tic-Tac-Toe Endgame (TE)	84.47	90.08
Zoo (ZO)	84.15	89.91

Time criteria Both the running time and the necessary number of iterations to reach the final stable solutions will be reported below by considering certain databases.

Table 5.2: Time performances (BKM vs $ISBKM$)(Certain case)

Databases	BKM		$ISBKM$	
	Time(s)	Iter	Time(s)	Iter
Breast Cancer Wisconsin	214	7	192	4
Car evaluation	284	9	206	7
Congressional voting records	231	7	167	4
Hayes-Roth	142	9	117	7
Lenses	56	5	49	3
Mushroom	322	8	296	5
Nursery	503	10	377	5
Spect heart	163	7	113	4
Tic-Tac-Toe Endgame	235	8	189	5
Zoo	154	5	109	4

Table 5.1 shows that the incremental proposal is more accurate than the static approach for all considered databases in such certain framework. As instance for Car evaluation database, the mean accuracy value goes from 85.37% with BKM method to 94.34% by applying our proposal $ISBKM$.

Table 5.2 shows further that the incrementally incorporation to new attribute, by using $ISBKM$ approach, is better than the static procedure namely BKM method in terms of time and iteration number. By considering the same base, namely Car evaluation, to reach the stability of the cluster collection, we need only 206 seconds with $ISBKM$, while this running time is 284 seconds for the static method. Moreover, a decrease in the iteration number is also noted by using our dynamic approach, so this number goes from 9 with BKM to only 7 with $ISBKM$ counter part.

2. Uncertain case:
   By considering four degrees of uncertainty level ($]0, 0.25]$, $]0.25, 0.5]$, $]0.5, 0.75]$ and $]0.75, 1]$), named as Level 1, Level 2, Level 3 and Level4, both the clustering accuracy and the time measures are computed and tabulated in the following.

   Clustering accuracy criterion We report the results relative to this uncertain case in Table 5.3. Indeed, the two last columns of this Table summarize all results and the mean accuracy for the different tested degrees of uncertainty is computed.

   Time criteria We expose, below, the obtained running time by seconds and the iteration number for both static and dynamic cases. In Table 5.4 and Table 5.5, we report the mean time indicators of the different degrees of uncertainty results.

Table 5.3: Clustering accuracy (%)(BKM vs $ISBKM$)(Uncertain case)

Databases	Level 1		Level 2		Level 3		Level 4		Mean	
	BKM	$ISBKM$	BKM	$ISBKM$	BKM	$ISBKM$	BKM	$ISBKM$	BKM	$ISBKM$
BCW	84.67	90.27	82.19	88.00	80.98	86.08	79.15	83.01	81.75	86.84
CE	84.03	91.03	82.72	90.56	81.22	88.64	79.38	85.55	81.84	88.95
CVR	84.89	91.29	83.26	89.45	81.64	87.55	80.44	84.28	82.56	88.14
HR	83.68	89.47	81.97	89.29	80.32	86.56	78.93	83.64	81.23	87.24
LS	83.41	88.33	81.41	87.06	79.17	85.36	77.11	82.70	80.28	85.86
MM	83.16	88.78	80.00	86.29	77.69	83.66	75.31	80.10	79.04	84.71
NY	82.94	88.19	79.75	86.83	77.72	83.72	76.22	80.87	79.16	84.90
SH	81.95	89.18	79.22	85.69	77.66	83.48	75.96	80.85	78.70	84.80
TE	81.31	88.98	78.19	86.93	76.32	85.15	75.05	81.21	77.72	85.57
ZO	83.66	89.22	81.83	87.19	79.87	85.53	78.03	82.78	80.85	86.18

Table 5.4: Running time(s)(BKM vs $ISBKM$)(Uncertain case)

Databases	Level 1		Level 2		Level 3		Level 4		Mean	
	BKM	$ISBKM$	BKM	$ISBKM$	BKM	$ISBKM$	BKM	$ISBKM$	BKM	$ISBKM$
BCW	312	194	327	201	361	233	433	307	358	234
CE	329	332	366	342	422	363	461	381	395	355
CVR	243	181	276	197	307	219	347	283	293	220
HR	170	152	181	164	196	171	223	198	193	171
LS	60	58	63	59	71	67	102	69	74	63
MM	342	304	391	312	487	393	513	431	433	360
NY	520	397	597	413	684	562	721	611	631	496
SH	154	127	163	138	199	152	228	178	186	149
TE	251	194	273	201	321	246	414	299	315	235
ZO	161	121	168	137	198	153	234	187	190	150

Table 5.5: Iteration number (BKM vs $ISBKM$)(Uncertain case)

Databases	Level 1		Level 2		Level 3		Level 4		Mean	
	BKM	$ISBKM$	BKM	$ISBKM$	BKM	$ISBKM$	BKM	$ISBKM$	BKM	$ISBKM$
BCW	7	4	7	4	9	5	12	7	9	5
CE	10	8	11	9	13	11	14	12	12	10
CVR	7	4	8	4	10	5	11	6	9	5
HR	10	8	11	9	12	10	13	11	12	10
LS	5	4	5	5	6	6	8	7	6	6
MM	8	5	8	6	9	7	11	9	9	7
NY	11	6	11	6	12	7	14	8	12	7
SH	8	5	9	5	9	6	10	7	9	6
TE	8	5	8	6	10	7	11	8	9	7
ZO	5	5	5	5	6	6	8	7	6	6

Based on the detailed results of Table 5.3 (from column 2 to column 9), we can remark that the clustering accuracy decreases when the uncertainty increases.

It is true for all considered databases.

It is interesting to note that the different mean values of $ISBKM$ accuracy remain high, between 84.71% and 88.95%, while those of the static method are between 77.72% and 82.56%. For example for Congressional voting records database, the accuracy goes from 81.75% using BKM to 86.84% with $ISBKM$ as tabulated in Table 5.3.
These results proves the efficiency of our developed approach in such uncertain and dynamic features environment.

In Table 5.4 and Table 5.5, the time indicators results are reported for this uncertain case as well as their mean values (the two last column of each table).
Remark that these time indicators increase when the uncertainty degree increases from Level 1 to Level 4 (from column 2 to column 9). Indeed, these experimental results prove the performance of $ISBKM$ method which is faster than BKM for almost all databases. For instance for Congressional voting records base, the time indicators are remarkably decreased. So, the running time goes from 293 seconds with BKM to 220 seconds by using $ISBKM$. On the other side, the number of iterations is 9 for the static procedure and only 4 for the dynamic one.

From these tables, the interesting results of our incremental proposal under uncertainty confirm that our method is most appropriate within such dynamic environment.

We illustrate the mean of all the accuracy results by the following figure which reports graphically the different results under uncertainty.

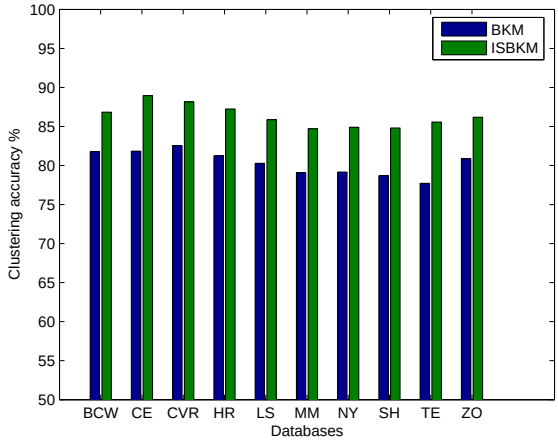
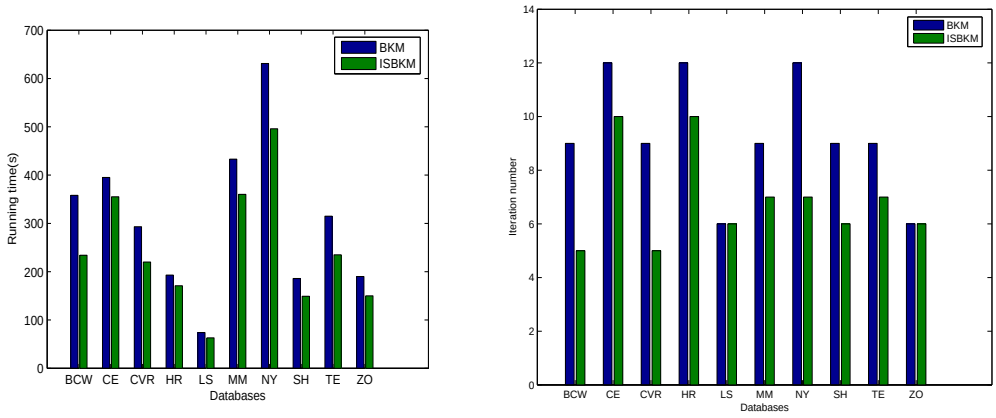


Figure 5.1: BKM vs $ISBKM$ clustering accuracy results under uncertainty (Mean)

To illustrate the obtained time results, Figure 5.2 gives a graphical presentation for the mean time performance, via the running time and the iteration number indicators, for both approaches relative to all used databases.



(a) Running time(s) (b) Iteration number

Figure 5.2: BKM vs $ISBKM$ Time indicators results under uncertainty (Mean)

As principal results of these experimentations, we observe that reclustering the whole data sets after the attribute set extension, using non-incremental BKM, is not a good practice since we need more time and a great number of iterations for finding the solution (stable partition) than the one obtained when we apply the $ISBKM$ approach. This latter updates an initial partition where some instances are already well classified in the appropriate clusters in spite of their extension.

$DSBKM$ simulations

In this part of our experimental study, both accuracy of clustering results and needed time to reach these solutions are computed to evaluate the quality of the obtained partitions. The standard certain case will be studied in first, then the uncertain counter part must be presented.

1. Certain case:
   This is the first studied case. Thus, the three criteria results will be detailed in the following tables.

   Clustering accuracy criterion The quality of resulting partitions is computed and reported in Table 5.6.

Table 5.6: Clustering accuracy (%)(BKM vs $DSBKM$)(%)(Certain case)

Databases	BKM	$DSBKM$
Breast Cancer Wisconsin	85.27	87.09
Car evaluation	85.37	91.62
Congressional voting records	85.96	89.90
Hayes-Roth	84.51	86.51
Lenses	84.22	87.56
Mushroom	83.81	86.12
Nursery	83.36	89.00
Spect heart	84.19	88.33
Tic-Tac-Toe Endgame	84.47	90.77
Zoo	84.15	88.78

   Time criteria Both running time and the number of iterations are presented in Table 5.7 as time performance criteria.

Table 5.7: Time performances (BKM vs $DSBKM$)(Certain case)

Databases	BKM		$DSBKM$	
	Time(s)	Iter	Time(s)	Iter
Breast Cancer Wisconsin	214	7	192	4
Car evaluation	284	9	206	9
Congressional voting records	231	7	167	4
Hayes-Roth	142	9	117	7
Lenses	56	5	49	3
Mushroom	322	8	266	5
Nursery	503	10	377	5
Spect heart	163	7	113	4
Tic-Tac-Toe Endgame	235	8	199	5
Zoo	154	5	109	4

   From the previous tables, we find that our proposal $DSBKM$ provides better results than BKM. Indeed, it is more accurate and faster to reach the final

solution. As example for Spect heart database, the accuracy value is 84.19% with BKM, while it is 88.33% using $DSBKM$. The running time of this same database goes from 163 seconds with static method to 113 seconds for the decremental method developed within this Chapter. The corresponding iteration number goes from 7 by using the static approach to 4 for $DSBKM$.

2. Uncertain case:
   All tested databases, within certain framework, will be used under uncertain framework to evaluate our decremental proposal.

Clustering accuracy criterion By the following Table 5.8, the detailed results respectively to accuracy index are reported by considering the four studied uncertainty levels. Besides, the mean values are also computed and presented.

Time criteria The clustering time performance of the developed $DSBKM$ together with that of BKM method on the real uncertain databases is tabulated in the following tables. In addition to time complexity, we take also the convergence speed measure, expressed by the iteration number needed to update our partition and obtain the stability. The results of different uncertainty levels, are summarized in Table 5.9 and Table 5.10.

Table 5.8: Clustering accuracy (%)(BKM vs $DSBKM$)(Uncertain case)

Databases	Level 1		Level 2		Level 3		Level 4		Mean	
	BKM	$DSBKM$	BKM	$DSBKM$	BKM	$DSBKM$	BKM	$DSBKM$	BKM	$DSBKM$
BCW	83.17	86.71	82.12	83.14	79.70	79.90	74.20	76.51	79.80	81.57
CE	85.33	87.98	84.71	85.33	83.20	83.90	79.47	80.13	83.18	84.34
CVR	84.28	88.51	83.17	84.37	82.56	82.39	78.33	79.99	82.09	83.82
HR	80.44	84.74	79.23	82.22	76.70	78.40	73.14	75.75	77.38	80.28
LS	79.19	83.78	78.19	81.92	77.88	80.75	71.23	76.14	76.62	80.65
MM	81.18	85.83	80.28	84.19	78.90	81.70	74.71	79.66	78.77	82.85
NY	82.05	85.05	81.97	84.66	81.25	83.12	77.84	78.01	80.78	82.71
SH	79.13	82.33	78.43	80.57	76.25	77.08	73.22	75.07	76.76	78.76
TE	78.40	87.77	76.88	82.13	72.56	78.90	69.27	73.19	74.28	80.50
ZO	77.30	84.70	76.67	81.44	71.80	74.50	69.41	70.81	73.80	77.86

Table 5.9: Running time(s)(BKM vs $DSBKM$)(Uncertain case)

Databases	Level 1		Level 2		Level 3		Level 4		Mean	
	BKM	$DSBKM$	BKM	$DSBKM$	BKM	$DSBKM$	BKM	$DSBKM$	BKM	$DSBKM$
BCW	296	261	313	294	361	327	402	394	343	319
CE	328	255	377	283	422	395	497	413	406	337
CVR	254	197	274	208	307	268	371	308	302	245
HR	149	131	154	138	196	183	227	211	182	166
LS	61	58	66	59	71	67	91	71	72	64
MM	381	271	401	291	487	322	599	407	467	323
NY	583	477	603	513	684	581	733	631	651	551
SH	168	127	174	141	199	152	277	197	205	154
TE	252	222	266	243	321	318	441	394	320	294
ZO	166	133	170	148	198	187	281	228	204	174

Table 5.10: Iteration number (BKM vs $DSBKM$)(Uncertain case)

Databases	Level 1		Level 2		Level 3		Level 4		Mean	
	BKM	$DSBKM$	BKM	$DSBKM$	BKM	$DSBKM$	BKM	$DSBKM$	BKM	$DSBKM$
BCW	7	6	7	7	9	9	12	11	9	8
CE	9	9	10	9	13	11	14	12	12	10
CVR	8	5	8	5	10	7	12	9	10	7
HR	9	7	10	8	12	9	13	10	11	9
LS	6	4	7	5	8	5	9	7	8	5
MM	8	5	9	6	10	6	11	9	10	7
NY	10	6	11	6	12	8	15	10	12	8
SH	8	5	9	6	9	7	11	8	9	7
TE	8	6	8	7	10	9	13	11	10	8
ZO	6	5	6	5	7	6	9	8	7	6

When the attribute set is reduced to reach the desired actual attribute number

for each database, the updating process of the decremental method generates better cluster partition with highest accuracy value for the majority of tested databases for all studied levels of uncertainty as shown in Table 5.8. Indeed, these clustering values decrease from Level 1 to Level 4 (from column 2 to column 9), in other word the uncertainty increasing. For instance, the Hayes-Roth mean accuracy goes from 77.38% with BKM to 80.28% with $DSBKM$ as presented in Table 5.8.

Besides, all previous time indicators results which are tabulated in Table 5.9 and Table 5.10 show that the dynamic belief clustering $DSBKM$ generates better results (faster solutions) than those returned by applying BKM when the attribute set is reduced. This remark is valid for all studied levels of uncertainty for almost all databases. For example, the mean value of running time for the same database, namely Hayes-Roth, goes from 182 seconds with BKM method to 166 seconds with $DSBKM$ one (see Table 5.9). It is the same for all tested databases. Hence, our decremental proposal has a good effect on reducing the time indicators both running time and number of iterations. Indeed, the average number of iterations for this base goes from 11 to 9 by using $DSBKM$ method instead of BKM one as reported in Table 5.10.

This confirms that our proposal is well appropriate within such dynamic uncertain features context. We observe, by the exposed iteration number, that reclustering the whole data objects after the decreasing of the attribute set cannot be considered as a good practice comparing to the decremental developed process due to the great number of iterations needed to find the final stable partitions compared to the ones obtained via $DSBKM$ approach.

Figure 5.3 shows the effect of the selection of the most irrelevant feature and its elimination on the clustering accuracy results. In fact, this improvement of the results proves that the developed decremental method provides interesting approach to cluster data set under such uncertain and dynamic context comparing to the standard static one consisting in a reclustering a whole data after this feature set reduction.

A summary of the time indicators results under uncertainty is represented graphically by Figure 5.4. Indeed, this latter proves the fact that $DSBKM$ is faster than BKM for all bases. These two last evaluation criteria are significantly decreased in average within our dynamic framework compared to static method.

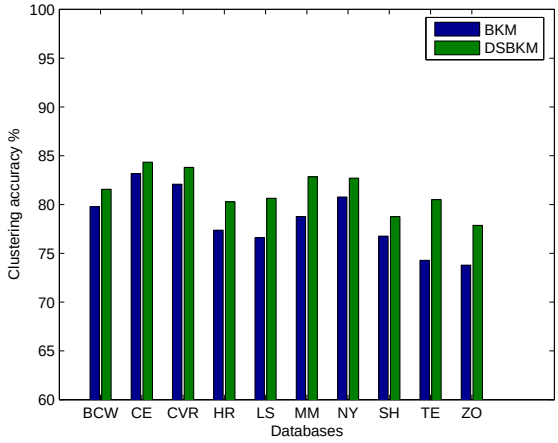
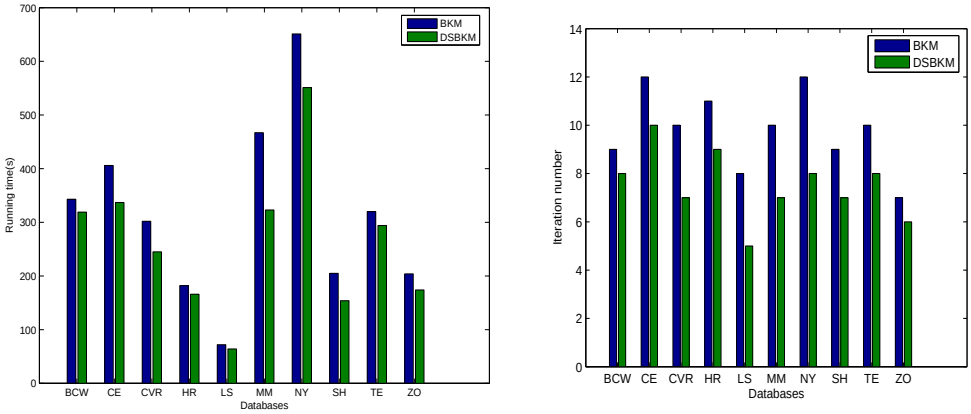


Figure 5.3: BKM vs $DSBKM$ clustering accuracy results under uncertainty (Mean)



(a) Running time((s)

(b) Iteration number

Figure 5.4: BKM vs $DSBKM$ Time indicators results under uncertainty (Mean)

We can conclude that by eliminating the most irrelevant attribute, which is selected via the proposed ranking-based selection method, the dynamic clustering process generates higher quality partitions with reduced time requirements and minimal iteration number.

5.5.3 Experimental summary

For each tested UCI data set, we run our incremental version a number of times equal to its attributes number. Indeed, each time, we eliminate initially one of these features and generate the cluster partition via BKM run. This eliminated feature will be added, thereafter, to evaluate this approach. The average results are the reported ones. For the decremental courter-part, at each execution one fixed attribute

is duplicated and considered during the initial clustering task. Then, the attained collection must be updated via its deletion.

From the experimental results relative to these dynamic proposals under uncertainty (both incremental and decremental methods), which are developed to handle dynamic feature set, we can summarize this study by some conclusions:

1. Best clustering quality is reached in most cases.
2. Quick partition update is ensured for almost all bases.

Note that these new dynamic clustering approaches which are developed under uncertainty are originally presented in (Ben-Hariz and Elouedi, 2008; Ben-Hariz and Elouedi, 2011).

5.6 Conclusion

New clustering approaches which dynamically update the cluster partition, after reducing or addition of feature(s), were proposed within this chapter. Thus, ISBKM and DSBKM methods were detailed to generate best and faster cluster partitions of uncertain data sets.

The uncertainty appears in attribute values which is represented through the belief functions. We have presented the major phases needed for the updating step under TBM framework. Especially, one feature selection approach as a pre-processing step is also developed under uncertainty.

These techniques, which are all based on similarity/dissimilarity concepts under uncertainty, are able to update the groups without a reclustering task.

In the next chapter, we suggest to develop another dynamic clustering family under uncertainty to deal with non-stationary data sets items. So, we will present the two procedures: incremental and decremental approaches for dynamic such data set items.

6

Clustering dynamic data items under uncertainty

6.1 Introduction

In the previous chapter, we have been interested by non-stationary feature set in an uncertain context. In this chapter, we will study non-stationary data set items as the second handled dynamic data set dimensionality. Indeed, during the recent years, non-static data sets have attracted considerable attention in different fields of machine learning. The continuous arrivals and movements of data items raise new challenges and research problems.

We address, here, the clustering problem in which, on one hand, data are gathered during time. On the other hand, these data may be reduced by eliminating one or more objects. By observing data at each time, a dynamic update of the clustering results to minimize one fitness measure between the obtained resulting model and the altered data is required. Thus, promising approaches will be developed in this chapter as possible solutions. These proposals permit an updating of clustering structure without reclustering these altered data sets at each time. It consists, simply, in a rearrangement process in order to absorb these new information. These approaches are developed under uncertainty where the attribute values of clustered items may be uncertain.

Recently, works in (Aranganayagi and Thangavel, 2010; Su et al., 2009) create the clusters by adding incrementally the data set objects. Furthermore, in (Boukharouba and Lecoeuche, 2008; Truta and Campan, 2007), incremental as well as decremental data sets procedures are provided to cluster non-static objects sets. These latter are based on SVM technique. Another methodology for dynamic data mining based on fuzzy clustering (Crespoa and Weberb, 2005) was developed. This method generates

fuzzy degrees memberships for the new arrivals. It does not deal with the decreasing of the data set which are known with certainty.
All these non-standard methods deal with numeric data and assume that their values are certain and thus could not deal with the challenge of uncertainty management.

After the problem formulation, we detail our proposed belief counter-parts. Nonetheless, basis of incremental and decremental procedures and the respective developed methods are presented to handle both the addition and the deletion of data set items. Before concluding this chapter, we report the different results obtained from uncertain databases in order to demonstrate the applicability and the effectiveness of our dynamic clustering framework based on belief function theory in comparison with static one.

6.2 Problem statement

Belief Clustering for Dynamic Data sets Items approaches ($BCDDI$) will be proposed for both an incoming flow of objects as well as a possible item reduction of the initial data set. This is the second subfamily of $BCDDD$ framework. This is ensured within an uncertain context. Thus, Incremental N Belief K-modes ($INBKM$) and Decremental N Belief K-modes ($DNBKM$) methods will be developed to deal with this problem besides the uncertainty parameter. Remark that I and D are respective to the process nature (incremental or decremental one), while N represents the initial number of instances to increase or to decrease.

The problem to be solved in this chapter could formulated as follows: Knowing a non-stationary data set and an already obtained partition, how to recursively update this partition to represent as possible the actual data distribution. The handled non-stationary data sets are imprecise ones meaning that the attribute values are uncertain.

According to some specified dynamic operations such as splitting and merging, which will be defined later, this partition must be adapted over time using the proposed dynamic clustering approaches under uncertainty. Here, update the partition means the cluster models rearrangement i.e. the recomputation of their modes following new object assignment(s) or the output of old object(s).

Indeed, there are several real-world applications of these phenomena. As example, let us mention dynamic customer segmentation to detect changes of customer behavior respectively to different periods. Indeed, customer transactions must be gathered

periodically. The deletion of some customers may also be encountered in some cases like the closing of one shop. Indeed, these customers may be characterized by uncertain attribute values. For traffic management, such dynamic manner to proceed identifies continuous changes in traffic states. As instance, the study of all Mondays respectively to each period and consequently update the generated classifier. It is the same, the characteristics may be known with uncertainty.

To ensure these updates, some issues could be raised, but the most important one is: Which cluster(s) should be updated and how?

For this purpose, we will present dynamic classifiers dedicated to both incremental and decremental cases developed to handle respectively an extended and a reduced data set.
The main parameters of our proposals are:

- Inputs:
  $P = \{C_1, C_2, .., C_l, .., C_k\}$.
  UT: A non-stationary data set which means that is may be extended as it can be reduced.
- Output: P_+ or P_-.

6.3 Basis of dynamic data clustering under uncertainty

6.3.1 Global view

If future behavior of the considered data set is not similar to past one, the use of the initial data mining outputs could lead to non-acceptable results as well as to non-appropriate decisions. Thus, a combination of traditional data mining with these dynamic aspects (future behavior) must be considered. For instance, let us mention the arrival of new customers. Therefore, if these latter are similar to already handled ones, then the already attained results remain valid. Otherwise, the update of these results must be done.
Three possible adopted strategies can be mentioned (Crespoa and Weberb, 2005):

- The negligence of these changes without any updating. The already generated classifiers remain stable.
- New results are developed using all the available data. It consists in a reclustering process.

- An update of the classifier is performed based on the initial results and the new data information. This is ensured via an updating process without a reclustering step.

With the first strategy which is judged as computationally cheap, the current data set tendencies could not be detected since these new information are not incorporated. The second one is quite costly, whereas the last one is more efficient, computationally, than the previous one and more faithful to the new data set structure. Thus, here, we will adopt this update-based methodology to develop our proposals.

We focus our approaches on belief clustering, more precisely BKM method since the handled categorical attribute values may be uncertain. These latter are represented and managed via belief function theory concepts. It is clearly that our problematic (uncertainty versus certainty) is partially different from the already handled ones in the exposed literature. It also addresses the decremental case.

The different situations will be described by the following scenarios:

- For incremental case:
  - The observation(s) to incorporate are known. Therefore, it will incorporated immediately.
  - Only the significant observations will be preserved. Thus, a pre-processing step must be conducted to select them.

- For decremental case:
  - The item(s) to eliminate are fixed. So, their elimination will be performed directly.
  - For an observation set, only the most irrelevant must be suppressed. This selection is the aim of the developed detection significant data method.

From these scenarios, it is shown that to conduct the dynamic process (incremental or decremental), we have to generate, respectively to the initial partition, the significant subset of data to incorporate or to eliminate.
Thus, a detection of these significant items step will be adopted firstly. For example, if a new instance to integrate is judged as an outlier, then it will be left out and not considered during the update phase. Indeed, this pre-processing step as well as the incremental and the decremental methods will be presented below.

6.3.2 Basic terminology

Suppose we have to add y new data points or to remove ℓ old ones simultaneously. Let us denote the resulting data sets as follows: $UT^{+} = UT \cup \{Y\}$ and $UT^{-} = UT \setminus$ Ł,

where $|Ł| = \ell$ and $|Y| = y$.
We define, thus, Y and Ł sets:

$$Y = \{X_i | i \in \{n+1, n^+\}\} \tag{6.1}$$

with $|Y| = y$, $n^+ = |UT^+|$, $n^+ = n + y$ and $y \geq 1$.

$$Ł = \{X_i | i \in \{1, n\}\} \tag{6.2}$$

with $|Ł| = \ell$, $n^- = |UT^-|$, $n^- = n - \ell$ and $n^- \geq k$.
This condition, namely $n^- \geq k$, is also necessary to guarantee that no empty clusters are generated after the updating step.

6.4 Dynamic data processing using belief clustering

As we have seen above, a pre-processing step may be needed in some cases. By the following, we detail the corresponding approach. Thereafter, the two dynamic clustering methods, $INBKM$ and $DNBKM$, under uncertainty are presented. For all these methods, the distance measure will be the core concept as detailed below.

6.4.1 The detection of significant data step

This is done in order to preserve only the items which are judged as significant enough and consequently must be considered during the dynamic process. The results of such pre-processing step are the most relevant observations or the the most irrelevant ones respectively to incremental and decremental cases.
The corresponding pseudo algorithm may be summarized as follows by considering these input parameters:

1. The already attained partition P of k clusters by applying BKM approach.
2. The subset Y to incorporate or Ł set to eliminate.
3. One threshold γ or δ.
   The first one is used within the incremental context, while the second one is considered within the decreasing framework. Indeed, γ and δ parameters must be defined to well specify Y' and Ł$'$ sets. Y' or Ł$'$, are the resulting sets by considering only the items having a significant impact (relevant versus irrelevant). Hereby, $|Y'| \leq |Y|$ and $|Ł'| \leq |Ł|$.

This method detects significant data and thus, preserve only some items from the input ones to add or to eliminate which are the most relevant ones within the incre-

mental context and the most irrelevant ones respectively to the decremental framework. The core concepts of this pre-processing step are γ and δ input parameters.

Detection of significant data algorithm (P, Y or Ł, γ or δ)

```
Data: P, Y or Ł, γ or δ
Result: Y' or Ł'
begin
    //Incremental case
    if Ł = ∅ then
        for each X_i ∈ Y do
            -Compute all its k distances respectively to k clusters of P
            if ∀ k distances > γ then
                Y' ← Y \ {X_i}
    // Decremental case
    else
        for each X_i ∈ Ł do
            if Its distance respectively to its current cluster ∉ δ then
                Ł' ← Ł \ {X_i}
end
```

Both threshold δ and γ must be specified to judge the meaning of one new item or the uninteresting old item. We present by the following how to define their values.

- γ definition:
  Let us consider the initial partition P of k clusters. For each of these clusters, we have to compute all distances of its items from the mode in order to find the maximal value. Indeed, this latter mentions the most far item from the mode within this cluster. It is defined thus: $Max_l = Max(D(X_u, Q_l))$, with $l \in [1, k]$, $u \in [1, n_l]$ and n_l is the cardinal of this cluster.
  Thereafter, the maximal distance within the partition must also be defined as follows: $Max_P = Max(Max_l)$, with $l \in [1, k]$. This is the distance of the most far item respectively to its cluster mode by considering all the partition.
  Indeed, we define as follows $\gamma = Max_P$.
  Then, for each object $X_i \in Y$: If $\forall D(X_i, Q_l) > \gamma$ then, $Y' = Y \backslash X_i$. Therefore, $|Y'| \leq |Y|$.

Indeed, within the incremental framework, if all distances of one considered instance exceed the γ value, then this item is determined as too dissimilar to all clusters. Therefore, it cannot be incorporated and consequently must be removed from Y. Thus, the updated set Y' is achieved.

- δ definition:
  To define this parameter, the same partition is considered to compute the same distances. Then, the minimum values will be the preserved ones instead of the maximum ones.
  Indeed, the minimal distance within all partition P is computed as follows: $Min_P = Min(Min_l)$, with $Min_l = Min(D(X_u, Q_l))$, with $l \in [1, k]$, $u \in [1, n_l]$ and n_l is the cardinal of this cluster.
  By considering also $MaxMin_P = Max(Min_l)$, then $\delta = [Min_P, MaxMin_P]$.
  For each object $X_i \in$ Ł: If $D(X_i, Q_r) \notin \delta$ then, Ł$' =$ Ł$\backslash X_i$, with Q_r is the mode of the current cluster C_r of the item X_i.

For the decremental context, if the distance of one object to eliminate respectively to its current group is lower to δ threshold, then this item is judged as very close to its cluster. Consequently, it cannot be eliminated and its removal from Ł must be performed. Hence, Ł$'$ is attained with $|$Ł$'| \leq |$Ł$|$.

6.4.2 Clustering online data within uncertainty

This section addresses dynamic clustering problem in which data may be gathered during a time interval. Indeed, a new data object may be observed each time and this requires a dynamic update of the clustering results.
The simplest idea to address this problem is to cluster the updated data set again at each time. This is judged as a very costly technique. However, more promising approach namely incremental clustering process, based on the optimization of an appropriate measure, will be developed as solution to cluster such online data.
Based on one initial cluster partition, and when new y instances are detected we have to consider them and rearrange our groups. Contrary to the previously developed approaches, in Chapters 4 and 5, which deal with the increase of only one cluster or one attribute. Here, our proposal permits to incorporate more than one object simultaneously since only one object handling will be a simple application of a new iteration of the static method namely BKM.
A first naive idea is to consider an already proposed dynamic approach, within this thesis work, to conduct this incremental process.
Indeed, all these new items may be grouped together in one new cluster at the beginning. Thereafter, one of $DKBKM$ approaches (see Chapter 4) will be performed to obtain from these $k + 1$ clusters only k ones. Note that this manner to proceed does not consider the similarity's degree of all these new arrivals which must be, naturally, the used criterion to perform this object gathering.
In the following, new dynamic approach will be developed to solve the above problem. Recall that, initially, the significance of these items is checked. Consequently,

these data may appear as stray data. Therefore, the dynamic phase is not conducted. Otherwise, the first test to perform is to check the cardinality of the new block of data to absorb. Two cases must be distinguished $y \ll n$ or not. Respectively to this condition, we will present an appropriate incremental approach to adopt.
For the first case, we consider in what follows incremental clustering as an exact on-line approach to construct recursively the solution by incorporating one point at a time. These new data will be absorbed in the old clusters. Indeed, a large number of absorbed data may change the density of clusters considerably, so that they should be split into two or more groups. Thus, a merge step may be performed to preserve only k clusters.
However, where the previous condition ($y \ll n$) is not valid, these y objects will be clustered in k groups. Then, a fusion of this resulting partition and the initial one will be done in order to attain k clusters which is the consensus solution. An appropriate combiner for solving this problem will be proposed under this uncertain context.
Whatever the considered case, the cluster memberships of the previous clustered data set items may change during each incremental step to keep all clusters in equilibrium and attain the stability of resulting partition. These two strategies for partition update used by the incremental algorithm can be described by the following pseudo algorithm.

$INBKM$ **algorithm** (UT^+, Y', P, $noMaxIter$, $Maximal\ cohesion$)

```
Data: UT+, Y', P, noMaxIter, Maximal cohesion
Result: P+
begin
    //This is the incremental process with y ≪ n.
    if y ≪ n then
        for Each X_i do
            1. Compute all its k distances respectively to k initial clusters.
            2. X_i is assigned to one existing cluster C_l, among the k initial ones, with
            the minimum of distance.
            3. Check the cohesion of this extended group.
            if Maximal cohesion is reached then
                1. This cluster is split into two.
                2. Merge step must be performed to preserve k clusters.
        Update the modes of the altered clusters.
        Retest the objects assignments.
    else
        BKM(Y', k, P')// P' is the resulting k groups by clustering Y' items.
        Partition combiner (P', P, P+)
end
```

Note that each merging or division of cluster(s) must be followed by an updating step of the corresponding mode(s) of these altered groups.

The *Maximal cohesion* parameter must be specified to decide the splitting of one extended cluster. We will consider for our study that $Maximal\ cohesion = 2{*}Initial\ cluster\ cohesion$, with $Initial\ cluster\ cohesion$ is the initial cohesion of the cluster before its extension.

The Y' input parameter may be the output of the detection of significant data algorithm (the new data to incorporate are not known a priori) or specified by one expert (a known subset of observations to integrate).

We will explain, by the following pseudo-algorithm, the partitions combiner procedure which will be able to merge both P and P' partitions of k groups each one and produce, finally, only one partition P_+ of k clusters.

1. Compute the $k * k$ inter-cluster dissimilarities matrix by using Equation 4.2.
2. Merge the two clusters with the minimum separation.
3. $k \leftarrow k - 1$.
4. Recompute the inter-cluster matrix. It consists in an update task (replace the two merged clusters by only one) and consequently recompute the corresponding inter-cluster distances.
5. reiterate from step 2 until the resulting clusters' number reaches k value.

Note that when the number of clusters attains k groups, we have to recompute all cluster modes and we have also to check the object assignments accordingly.

6.4.3 Clustering reduced data in the belief function context

Decremental clustering or unclustering training sample(s) may be also considered since non-stationary data includes also reduced data case. A decision to eliminate one or more objects, already considered during the clustering process, can be made. This decision, of one expert system or in order to improve the clusters' quality, must be handled. Indeed, the object(s) to eliminate can belong to one or several groups, so we have to check this condition besides their number.

For the first case, suppose that all objects to eliminate belong to one cluster C_l. Thus, this inequality must be checked $\ell < |C_l|$. This condition guarantees non empty

cluster following this reduction process, while the second case necessitates the checking of a similar condition but respectively to each considered cluster (the number of items to eliminate from one group and the cardinality of this latter).

Respectively to both cases, we have to test and improve the quality of the clustering solution by eliminating the objects which are judged as the most disturbing.

Starting from the full n-point solution, we will consider in the following the two cases. Thus, the respective summary algorithm will be detailed as follows.

$DNBKM$ **algorithm** (UT^{-},**Ł′**, P, $noMaxIter$, $Minimal\ cohesion$)

```
Data: UT⁻, Ł′, P, noMaxIter, Minimal cohesion
Result: P_
begin
    for Each cluster C_l do
        Eliminate among its items those belonging to Ł set.
        if |C_l| ≥ 1
        then
            Update its mode.
            Check its internal cohesion.
            if Minimal cohesion is reached then
                Merge step must be performed.
                Splitting step must be performed too to preserve always k groups.
        else
            C_l is eliminated.
            Splitting step must be performed to reach k clusters.
    Retest the objects assignements.
end
```

The *Minimal cohesion* must be defined in order to decide the merge of reduced clusters. It should be specified by the expert. We will define this input parameter in the following by: $Minimal\ cohesion = \frac{1}{2} * Initial\ cluster\ cohesion$. This latter is the cohesion of the cluster before its reduction.

The Ł′ input parameter may be the output of the detection of significant data algorithm (the old data to eliminate are not known a priori) or specified by one expert (a known subset of observations to delete).

6.5 Experimentations

Implementation of our dynamical proposals under the belief framework seems imperative to conduct their evaluation and validation tasks. Once the respective programs are implemented, we have performed several tests on real-world UCI databases for checking the feasibility and the efficiency of our approaches by computing their accuracies and the time indicators.

In this section, we will provide all necessary parts related to this evaluation task of our dynamic proposals besides the details of Appendix A. Then, we start by exposing our testing strategies. We further report our different attained results under certain and uncertain frameworks in comparison with the static approach ones. Finally, we conclude this experimental analysis.

6.5.1 Experimental strategies

The techniques for conducting both present dynamic methods are similar to the already exposed ones in the previous Chapters 4 and 5, since the used data sets, here, have a known number of items like in the previous dynamic frameworks (a known number of classes and a known number of features).
Indeed, we simulated the dynamic behavior of data set items by the following way. This latter consists of the below steps, respectively to the incremental method first and then the decremental approach:

- Incremental experimentation strategy:
  For each database, of n instances, we have to:
  - Perform BKM approach on a reduced set $UT \setminus Y$ and attain k clusters as result. Y set is chosen in a random manner. The number of its items, y, is randomly fixed in $]0\%, 30\%]$ of the original data set UT.
  - By considering these instances ($X_i \in Y$), we apply $INBKM$ method to absorb these new items and update the partition. The time and accuracy indicators must be computed.
  - Accept, now, the n actual items. Indeed, the static version must be used to cluster these latter in k groups. Obviously, the same evaluation criteria must be also calculated.
  - A comparative study must be conducted (static versus dynamic approach).
- Decremental experimentation strategy:
  Similarly, by considering one database, we have to:

- Duplicate few instances from UT set to create Ł subset. ℓ, the number of objects to delete, is also randomly fixed in $]0\%, 30\%]$.
- Consider both the initial n items as well as the duplicated ones. Then, BKM method will be carried out to attain k clusters and the performance indicators of this partition will be calculated.
- $DNBKM$ method will be applied to eliminate the subset of data which are incorporated following the duplication task. Likewise, the accuracy and time indicators must be computed.
- The static version, namely BKM algorithm, must be accomplished in order to build k groups of only n items now.
- The comparison of results of both static and dynamic processes will be done.

6.5.2 Empirical results and discussion

The aim of conducting these experimentations is to evaluate our dynamic methods by checking their effectiveness and efficiency.

In the following, we report the experimentations obtained from some UCI repository databases which are described by Appendix A. These experimental results are attained by applying all dynamic proposals which are developed, within this chapter, to deal with clustering non-static data. This is based on two kinds of evaluation criteria namely accuracy and time indicators. To more illustrate this, we give for each procedure (incremental and decremental) tables as well as graphical presentations relative to some databases results in both certain and uncertain frameworks.

Incremental procedure

In this part, we apply our incremental approach namely $INBKM$ on our selected UCI repository databases more precisely their belief versions. Tables 6.1 and 6.2 summarize the results based on the chosen evaluation criteria. Firstly, the certain case results are reported. Thereafter, those respective to the uncertain case will be presented.

- Certain case:
  We start by the first case representing the standard certain databases. Accuracy of attained cluster collections, the time requirements and the number of iterations are all computed and compared for both dynamic and static approaches. Indeed, this concerns the belief presentation of certain databases via certain belief functions.

Clustering accuracy criterion Clustering accuracy values of different tested databases are tabulated below.

Table 6.1: Clustering accuracy (%)(BKM vs $INBKM$)(Certain case)

Databases	BKM	$INBKM$
Breast Cancer Wisconsin (BCW)	85.43	85.99
Can evaluation (CE)	84.03	84.90
Congressional voting records (CVR)	82.89	83.04
Hayes-Roth (HR)	84.17	85.08
Lenses (LS)	83.88	84.91
Mushroom (MM)	83.04	82.99
Nursery (NY)	83.43	83.01
Spect heart (SH)	82.97	84.02
Tic-Tac-Toe Endgame (TE)	83.38	84.11
Zoo (ZO)	84.05	85.66

Time criteria $INBKM$ evaluation based on running time and the corresponding iteration number is tabulated in the following.

Table 6.2: Time performances (BKM vs $INBKM$)(Certain case)

Databases	BKM		$INBKM$	
	Time(s)	Iter	Time(s)	Iter
Breast Cancer Wisconsin (BCW)	308	7	242	6
Car evaluation (CE)	463	11	438	10
Congressional voting records (CVR)	298	8	273	7
Hayes-Roth (HR)	174	9	151	6
Lenses (LS)	53	6	41	5
Mushroom (MM)	441	9	449	10
Nursery (NY)	569	12	571	12
Spect heart (SH)	183	7	175	6
Tic-Tac-Toe Endgame (TE)	202	7	181	6
Zoo (ZO)	159	6	137	4

From Table 6.1, we can see that $INBKM$ in the certain case increases slightly the clustering accuracy for the most of the tested databases.
Let us mention as example the Spect heart database results, where the accuracy goes from 82.97% with BKM method to 84.02% using $INBKM$ one.

The same remark is induced from Table 6.2 which reports the different time indicators results respectively to each database. Let us continue with the same base, the running time is 183 seconds with the static approach and 175 seconds with the dynamic one. The corresponding number of iterations goes from 7 to 6, from static to dynamic framework.

From these tables, we find that the $INBKM$ is more accurate and fast than static version namely BKM to update cluster partition following the introduction of some new observations. This is reached for almost all databases.

- Uncertain case:
  We move to the second studied case representing uncertain databases within belief function framework. Three evaluation criteria are studied and reported by the following. Indeed, four levels of uncertainty will be considered to guarantee a further study and to monitor the behavior of both static and dynamic methods.

  Clustering accuracy criterion The quality of resulting partitions is judged based on the accuracy criterion. Detailed results are tabulated by the following in Table 6.3 respectively to the four uncertainty levels. Their summary is also reported in this table. Thus, the mean accuracy values are computed.

  Time criteria The same manner to test our approach is conducted hereby, from detailed results of the four studied uncertainty levels to their summary results. Hereby, time indicators are exposed in the following tables to decide which is the faster method, the dynamic or the static one.

Table 6.3: Clustering accuracy (%)(BKM vs $INBKM$)(Uncertain case)

Databases	Level 1		Level 2		Level 3		Level 4		Mean	
	BKM	$INBKM$	BKM	$INBKM$	BKM	$INBKM$	BKM	$INBKM$	BKM	$INBKM$
BCW	85.27	85.87	83.14	84.73	80.17	80.77	77.12	77.41	81.43	82.20
CE	83.99	84.71	82.98	83.87	79.90	80.13	76.98	77.28	80.96	81.50
CVR	82.70	82.88	81.26	81.99	79.13	79.98	76.97	77.88	80.02	80.68
HR	84.02	84.93	83.12	83.67	80.23	80.76	78.03	78.91	81.35	82.07
LS	82.81	84.04	81.13	82.23	78.83	79.27	75.44	76.77	79.55	80.58
MM	82.97	82.77	81.34	81.31	79.05	79.15	75.73	76.92	79.77	80.04
NY	83.13	83.00	82.22	82.24	78.89	79.13	74.91	76.38	79.79	80.21
SH	82.11	83.87	80.88	81.73	77.47	78.12	74.13	76.01	78.65	79.93
TE	83.08	83.66	82.51	82.72	78.33	78.75	75.44	77.12	79.84	80.56
ZO	83.77	84.14	81.05	82.17	77.67	78.23	75.03	77.28	79.38	80.52

Table 6.4: Running time(s)(BKM vs $INBKM$)(Uncertain case)

Databases	Level 1		Level 2		Level 3		Level 4		Mean	
	BKM	$INBKM$	BKM	$INBKM$	BKM	$INBKM$	BKM	$INBKM$	BKM	$INBKM$
BCW	327	271	352	294	389	344	427	403	374	328
CE	469	445	473	469	512	507	546	513	500	484
CVR	302	291	361	332	387	376	422	418	368	354
HR	192	171	250	218	289	274	334	328	266	248
LS	57	49	61	59	72	71	74	73	66	63
MM	462	463	509	491	572	543	633	607	544	526
NY	583	578	663	627	698	692	745	727	672	656
SH	194	191	231	224	302	299	381	341	275	264
TE	238	204	303	287	362	313	394	387	324	298
ZO	173	166	198	187	216	209	297	276	221	210

Table 6.5: Iteration number (BKM vs $INBKM$)(Uncertain case)

Databases	Level 1		Level 2		Level 3		Level 4		Mean	
	BKM	$INBKM$	BKM	$INBKM$	BKM	$INBKM$	BKM	$INBKM$	BKM	$INBKM$
BCW	8	7	9	8	10	9	12	11	10	9
CE	11	10	11	11	13	13	14	14	12	12
CVR	8	8	9	9	10	10	12	12	10	10
HR	10	7	11	9	12	10	14	12	12	10
LS	6	6	7	7	8	8	8	8	7	7
MM	10	10	11	12	13	13	15	15	15	15
NY	13	13	15	14	16	16	18	17	16	15
SH	8	7	9	8	11	10	12	11	10	9
TE	8	7	9	8	10	9	12	11	10	9
ZO	7	6	8	7	9	8	11	10	9	8

Table 6.3 shows that our dynamic belief clustering method give better clustering quality than BKM by considering the mean accuracy for all databases. For

example, the mean accuracy of generated cluster partition induced from Hayes-Roth database using BKM clustering technique is 81.35%, while it is 82.07% with $INBKM$ one.

From Table 6.4 and Table 6.5, we can conclude that $INBKM$ takes better place by considering time indicators as evaluation criteria. This latter gives better results than the static approach. Indeed, we find that our proposal decreases slightly both time indicators. For example from Hayes-Roth database, the mean running time(s) goes from 266 seconds with BKM to 248 seconds using $INBKM$. It is the same, a slight improvement of the iteration number is detected. As instance, Hayes-Roth database is clustered by using BKM in 12 iterations, while this is ensured via the dynamic method $INBKM$ in only 10 iterations.

These different $INBKM$ results are outlined in Figures 6.1 and 6.2. These latter give a graphical presentation of accuracy mean, time mean and the mean number of iterations under uncertainty.

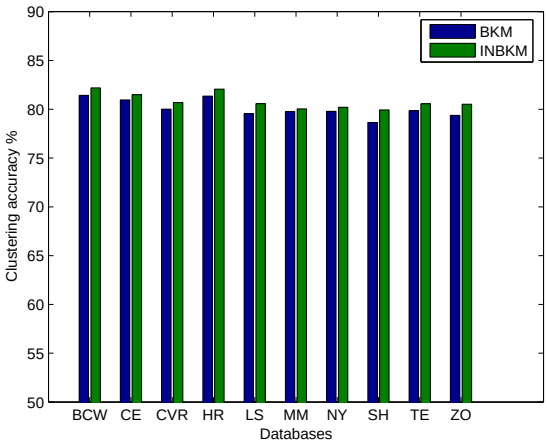
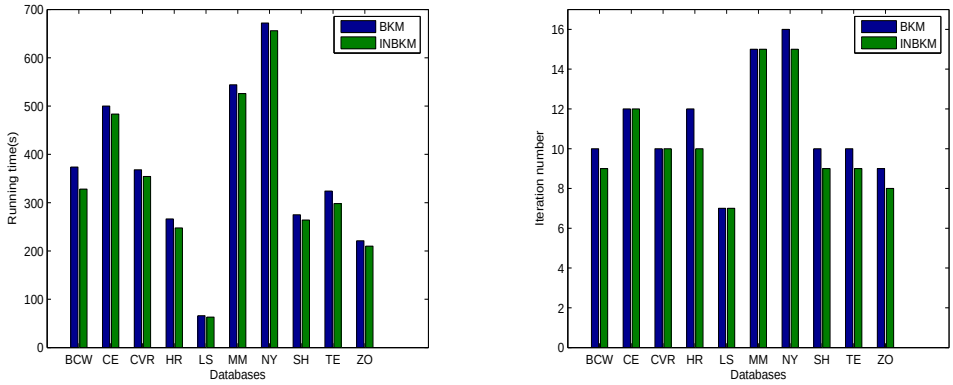


Figure 6.1: BKM vs $INBKM$ clustering accuracy results under uncertainty (Mean)



(a) Running time(s) (b) Iteration number

Figure 6.2: BKM vs $INBKM$ time indicators results under uncertainty (Mean)

In fact, as shown in previous tables and figures our incremental proposal surpasses the static one in most cases.

Decremental procedure

This subsection reports and interprets the decremental method experimentations results. The same belief real databases are used. Tables 6.6 and 6.7 show the different obtained results respectively to quality and time indicators by performing these experimentations for certain context. The uncertainty impact on these results will be studied later from Table 6.8 to Table 6.10.

- Certain case:
  The clustering accuracy, the time requirements and the number of iterations are all computed and compared for both dynamic and static approaches.

  Clustering accuracy criterion Clustering accuracy values of different tested databases are tabulated below.

Table 6.6: Clustering accuracy (%)(BKM vs $DNBKM$)(Certain case)

Databases	BKM	$DNBKM$
Breast Cancer Wisconsin (BCW)	85.43	89.14
Can evaluation (CE)	84.03	87.71
Congressional voting records (CVR)	82.89	86.97
Hayes-Roth (HR)	84.17	88.13
Lenses (LS)	83.88	86.53
Mushroom (MM)	83.04	87.17
Nursery (NY)	83.43	87.08
Spect heart (SH)	82.97	86.18
Tic-Tac-Toe Endgame (TE)	83.38	86.54
Zoo (ZO)	84.05	88.93

Time criteria $DNBKM$ evaluation based on running time and the corresponding iteration number is tabulated in the following table.

Table 6.7: Time performances (BKM vs $DNBKM$)(Certain case)

Databases	BKM		$DNBKM$	
	Time(s)	Iter	Time(s)	Iter
Breast Cancer Wisconsin (BCW)	308	7	291	6
Car evaluation (CE)	463	11	422	9
Congressional voting records (CVR)	298	8	268	7
Hayes-Roth (HR)	174	9	163	8
Lenses (LS)	53	6	49	5
Mushroom (MM)	441	9	437	9
Nursery (NY)	569	12	560	11
Spect heart (SH)	183	7	172	6
Tic-Tac-Toe Endgame (TE)	202	7	179	5
Zoo (ZO)	159	6	132	4

From Table 6.6 and Table 6.7, we find that $DNBKM$ generates better accuracies (increase) for almost all tested databases. It is the same, better time indices (decrease) are reported. As example for Car evaluation database, our decremental proposal result accuracy is 87.71%, while with BKM method is 84.03%. By considering the same base, $DNBKM$ reaches the stable solution in 422 seconds via only 9 iterations, whereas this time indicators go to 463 seconds and 11 iterations by using BKM static method.

Thus, $DNBKM$ is more accurate and fast than static version namely BKM to update cluster partition following the data set extension under certainty.

- Uncertain case:
  By the following, we will study the uncertainty impact on the different experimental results by considering the same evaluation criteria namely accuracy, time and iteration number.

Clustering accuracy criterion For each studied level of uncertainty, from Level 1 to Level 4, the corresponding results are tabulated in Table 6.8. Their mean values are computed and reported in this same Table.

Time criteria The same manner to test the time performances is conducted hereby. Time indicators are exposed in the following tables (Table 6.9 and Table 6.10) to decide which is the faster method, the dynamic or the static one.

Table 6.8: Clustering accuracy (%)(BKM vs $DNBKM$)(Uncertain case)

Databases	Level 1		Level 2		Level 3		Level 4		Mean	
	BKM	$DNBKM$	BKM	$DNBKM$	BKM	$DNBKM$	BKM	$DNBKM$	BKM	$DNBKM$
BCW	85.27	86.33	83.14	84.12	80.17	81.72	77.12	79.33	81.43	82.88
CE	83.99	85.17	82.98	84.25	79.90	81.44	76.98	79.64	80.96	82.63
CVR	82.70	85.08	81.26	83.71	79.13	80.49	76.97	77.93	80.02	81.80
HR	84.02	86.27	83.12	84.55	80.23	81.08	78.03	78.81	81.35	82.68
LS	82.81	84.68	81.13	82.48	78.83	79.92	75.44	77.42	79.55	81.13
MM	82.97	85.41	81.34	83.18	79.05	80.43	75.73	78.03	79.77	81.76
NY	83.13	85.92	82.22	83.27	78.89	81.07	74.91	77.53	79.79	81.95
SH	82.11	84.83	80.88	82.94	77.47	79.81	74.13	76.94	78.65	81.13
TE	83.08	84.12	82.51	83.01	78.33	79.99	75.44	77.14	79.84	81.07
ZO	83.77	86.56	81.05	84.26	77.67	81.04	75.03	78.88	79.38	82.69

Table 6.9: Running time(s)(BKM vs $DNBKM$)(Uncertain case)

Databases	Level 1		Level 2		Level 3		Level 4		Mean	
	BKM	$DNBKM$	BKM	$DNBKM$	BKM	$DNBKM$	BKM	$DNBKM$	BKM	$DNBKM$
BCW	327	299	352	321	389	374	427	412	374	352
CE	469	427	473	459	512	499	546	527	500	478
CVR	302	294	361	318	387	374	422	413	368	350
HR	192	181	250	215	289	279	334	321	266	249
LS	57	53	61	58	72	67	74	73	66	63
MM	462	451	509	499	572	547	633	601	544	525
NY	583	579	663	641	698	693	745	733	672	662
SH	194	189	231	222	302	294	381	351	277	264
TE	238	204	303	267	362	308	394	363	324	286
ZO	171	162	198	191	216	214	297	298	221	216

Table 6.10: Iteration number (BKM vs $DNBKM$)(Uncertain case)

Databases	Level 1		Level 2		Level 3		Level 4		Mean	
	BKM	$DNBKM$	BKM	$DNBKM$	BKM	$DNBKM$	BKM	$DNBKM$	BKM	$DNBKM$
BCW	8	6	9	7	10	8	12	10	10	8
CE	11	10	11	11	13	13	14	14	12	12
CVR	8	8	9	8	10	10	12	12	10	10
HR	10	9	11	11	12	13	14	14	12	12
LS	6	6	7	6	8	7	8	8	8	7
MM	10	10	11	11	14	12	15	13	13	12
NY	13	11	15	12	16	14	17	16	15	15
SH	8	7	9	8	11	9	12	11	10	9
TE	8	6	9	8	10	9	12	11	10	9
ZO	7	6	8	7	9	8	11	10	9	8

Tables 6.8, 6.9 and 6.10 summarize the experimental results of accuracy and

time indicators for both static and decremental clustering methods, where the mean values are presented.

Based on the previous tables, we find that $DNBKM$ produces more accurate solutions. For instance for Spect heart database, the mean clustering accuracy goes from 78.65% with BKM method to 81.13% with $DNBKM$.
The better time performance results are also attained for our decremental approach. Indeed, the same base namely Spect heart reaches the solution after 264 seconds and via 9 iterations, while the time indicators for the static method are 275 seconds and 10 iterations.
An improvement of the running time results for all tested databases by using our decremental method is mentioned, while the iteration number results, with $DNBKM$, are better or equal to those given by BKM for all bases.

The variability of all these studied evaluation criteria depends on the uncertainty level. Indeed, from Level 1 to Level 4, the accuracy decreases following this uncertainty increasing, whereas the running time and the iteration number, both have been increased. Therefore, results with $DNKBM$ are generally better than those given by BKM.

Thus, we conclude that $DNBKM$ is more accurate and faster than BKM. It is checked for almost all databases in such uncertain context.

To illustrate this experimental results, Figures 6.3 and 6.4 give a graphical presentation for the quality and time indicators attained by performing our decremental proposal.

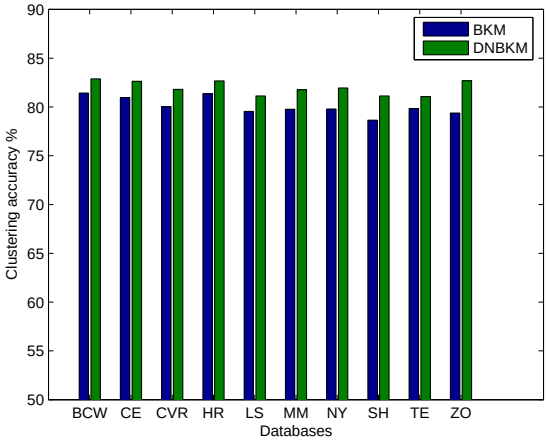
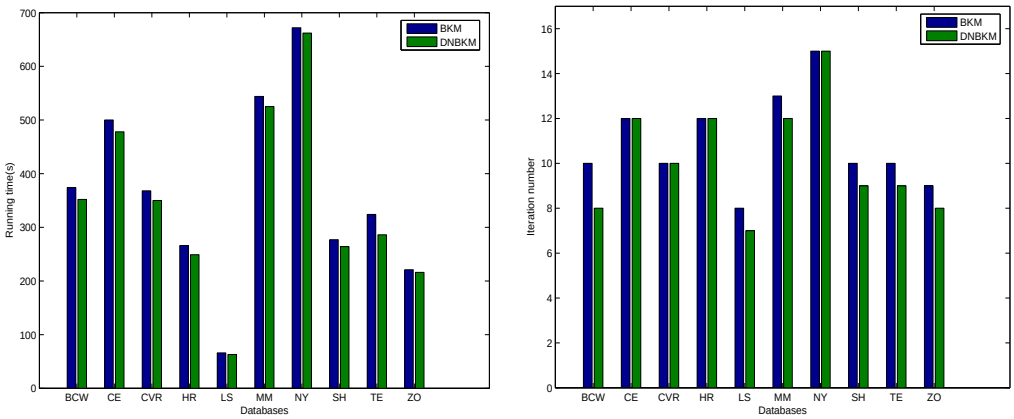


Figure 6.3: BKM vs $DNBKM$ clustering accuracy results under uncertainty (Mean)



(a) Running time(s) (b) Iteration number

Figure 6.4: BKM vs $DNBKM$ time indicators results under uncertainty (Mean)

We can conclude based on the above tables and figures that this decremental approach developed under uncertainty is well appropriate to update cluster partition when old item(s) are suppressed for almost all cases. The attribute values of these items are represented and managed via bba's concept under belief function framework.

6.5.3 Experimental summary

Within this chapter, belief counter parts of some real databases are used to test and evaluate our dynamic proposals. All studied evaluation criteria prove the ability of all developed methods to update the cluster partition under uncertainty when non stationary data are handled.

Indeed, to reach the stability, our approaches $INBKM$ as well as $DNBKM$ are faster than the static BKM one. Besides, the attained results are more accurate. In fact, our experimental results are better for almost all databases.

6.6 Conclusion

This chapter presented a class of algorithms for dynamic clustering. These algorithms are obtained by an extension of BKM for static uncertain clustering applications. A dynamic model namely $BCDDI$, encompassing both incremental and decremental evolution of the data set items, is considered.

The new data set objects to add to initial collection as well as the old instances to delete from original data set are both uncertain. The number of these objects constitutes the core concept of the two developed techniques. These proposals update the cluster partition of such dynamic data set without a whole re-clustering.

We attain interesting and positive results for all selected UCI databases regarding our new dynamic approaches comparing with those given by static counter part.

Conclusion

The main purpose of this thesis is to develop a new dynamic clustering framework under uncertainty based on both the K-modes method and the belief function theory to represent and manage the uncertainty in the attribute values. In fact, the Transferable Belief Model (TBM) is the adopted belief function theory interpretation by this research work.

We have first exposed the needed theoretical aspects namely an overview of both the belief function theory and clustering. Besides, we have detailed BKM (Ben-Hariz et al., 2006) approach and a new developed initialization method to deal with the randomness initialization problem. This new initial mode selection procedure, which is originally proposed in (Ben-Hariz et al., 2008) under static environment, is also useful to choose the first modes under our new dynamic belief clustering framework.

We have developed a family of dynamic clustering approaches under uncertainty. The uncertainty may appear in attribute values and is represented by bba's notion. The first sub-family called Belief Clustering for Dynamic Partition ($BCDP$) proposes incremental and decremental procedures to the increase and the decrease of the cluster partition cardinality. Another category of approaches is, thereafter, proposed but for dynamic feature set so-called Belief Clustering for Dynamic Data set Features ($BCDDF$). Next, we present the last sub-family developed for online data sets. This latter is named Belief Clustering for Dynamic Data set Items ($BCDDI$).
All our proposals aim at updating an initial cluster collection, without a whole reclustering again, when any change is detected. This is ensured based on the similarity/dissimilarity concepts.

In order to judge the performances of the BKM initialization approach as well as our dynamic belief clustering methods, we have implemented them with Matlab

V7.0. Thereafter, we have performed simulations on one selection of the UCI repository databases respectively to symbolic attributes. These latter bases are altered to introduce the uncertainty seen that the non-availability of real belief data sets. Different uncertainty degrees have been considered during this artificial creation task. Withal, the certain, probabilistic and possibilistic bases can be easily represented as particular cases of these belief data sets.

Clustering evaluation is based on clustering accuracy, running time and number of iterations. The first criterion is the most commonly used by the K-modes method works in literature. Then, we have reported the different results obtained from each dynamic approach and also compare them with those given by a static counter-part. Experiments have shown the feasibility and the quality of the different developments.

This thesis has thus addressed the challenges of representing and managing uncertain attribute values and clustering such imperfect data sets under dynamic environment. Thus, it has contributed to the development of dynamic clustering methods under uncertainty in belief function theory framework.

From the above, the dynamic environments investigated in this thesis, under uncertainty, are defined such that:

- The number of clusters changes, resulting in different modes, items migration between clusters and consequently their memberships update and change in the clusters' compactness.
- The number of clusters remains static with changing feature set. Then, the same consequences as the first dynamic environment kind (modes, compactness and items' membership changes) are also mentioned.
- The number of clusters remains stable with changing data set items.

Thus, the question of updating cluster partition in belief function theory in front of any change was handled.

However, there are still some open research issues worth future efforts. As mentioned before, three kinds of dynamic clustering have been advanced. A possible improvement will be related to their combination together and the development of an integrated dynamic system as a more general solution.

In addition to the uncertainty that pervades the attribute values, uncertainty may characterize the cluster memberships of data. Thus, soft clustering results are gener-

ated. Obviously, we suggest, as a future work, extensions of our proposals to manage this uncertainty form.

Other worthwhile future work includes the proposition of illustrative study of our proposals based on real case studies since the uncertainty and the dynamism characterize many real-world applications.

Finally, it would be interesting to extend our dynamic approaches of belief clustering to handle also the continuous attributes. Indeed, all over this thesis, we have dealt with only symbolic attributes. So, from real life data sets point of view, generally, these latter are characterized by a mixture of data types. Thus, the development of more general algorithms can be considered as another line of research. Therefore, we can develop the K-prototype approach under dynamic and uncertain context.

A

Experimental settings

A.1 Introduction

This appendix presents a description of the used databases to test our developed proposals is provided. Then, the uncertainty artificial creation procedure is detailed by exposing the all considered parameters. Finally, the evaluation criteria used to judge our methods, developed within this thesis work, are defined.

A.2 UCI data sets description

Once a clustering algorithm is developed, how it works should be tested by various data sets. In this sense, testing data sets plays an important role in the process of algorithm development. We have applied our methods to real databases obtained from the U.C.I repository of Machine Learning databases (Murphy and Aha, 1996).

Recall that we only deal with symbolic attribute values. Thus, we have just chosen the following bases. A brief description of these databases is given in the Table A.1, where #instances, #attributes, #classes denote respectively the total number of instances, the number of attributes and the number of classes.

Let us mention that this last column namely the class number will be considered during the evaluation clustering process as well as the clustering task. Indeed, to perform all developed K-modes-based dynamic clustering approaches under uncertainty, k parameter must be specified. This latter must be equal to the actual class number respectively to the considered database.

Table A.1: Description of UCI categorical databases

Databases	**#instances**	**#attributes**	**#classes**
Breast cancer Wisconsin (BCW)	699	9	2
Car evaluation (CE)	1728	6	4
Congressional voting records (CVR)	435	16	2
Hayes-Roth (HR)	160	5	3
Lenses (LS)	24	4	3
Mushroom (MM)	8124	22	2
Nursery (NY)	12960	8	5
Spect heart (SH)	267	22	2
Tic-Tac-Toe Endgame (TE)	958	9	2
Zoo (ZO)	101	17	7

Despite the great interest to the field of clustering under uncertainty recently, such real uncertain data, with uncertain attribute values represented through belief functions are not very available. Hence to experiment developed methods within such context, the creation of uncertainty in artificial way is done. Data sets of the U.C.I. repository are crisp data and do not contain neither uncertain attribute values nor uncertain class values. Then, these databases do not represent uncertainty. Thus, we have to incorporate this uncertainty by creating, artificially, belief attribute values relative to our dataset instances in order to generate their belief counter-parts as detailed in the next subsection.

A.3 Data sets in a belief function framework

Since there are not available real data sets containing uncertainty within the belief function framework, we have modified these databases by introducing uncertainty in the attribute values of their instances in order to obtain uncertain data sets. This uncertainty is presented through bba's concept given on the set of possible attribute values respectively to each instance. So, the question is how construct these bba's to obtain uncertain data sets?

A.3.1 Artificial bba's creation

The bba's are artificially created by taking into account these following basic parameters:

- The actual attribute values of the training instances as detailed in (Murphy and Aha, 1996).

- Degree of uncertainty p_j, where $1 < j \leq s$ (one for each attribute): it will vary in [0, 1] interval. The fixed value of p has a direct effect on the quality of results. It is an input uncertainty parameter which must be defined by one expert. In fact, for a large value of p_j, the clustering quality will decrease.
  We will consider these four different intervals of p_j for our simulations:

  - Level 1: we take $p_j \in]0, 0.25]$
  - Level 2: we take $p_j \in]0.25, 0.5]$
  - Level 3: we take $p_j \in]0.5, 0.75]$
  - Level 4: we take $p_j \in]0.75, 1]$

  Recall that for $p_j = 0$ the total certainty is considered (the certain belief function), while with $p_j = 1$ the total ignorance is handled (the vacuous belief function). Indeed, we will not consider this first case ($p_j = 0$) seen that our thesis work deals with clustering uncertain databases under dynamic environments.

- Percent of uncertain objects as well as the percent of uncertain attributes in a data set: the first one represents the percent of generated uncertain objects for a given data set, whereas the second mentions the percent of attributes with uncertain values. For these uncertainty parameter three degrees will be considered during the creation process, namely low middle and high, for the percent of uncertain objects and the percent of uncertain attribute.
  They are defined as: $[0, 35]$, $]35, 70]$ and $]70, 100]$, where all values are expressed as a percentage.

The idea is to assign to each attribute, a *bba* over the set of remaining attributes of this object, based on the set of their possible values .

After fixing the uncertainty degree relative to each attribute, we assign:

(1) $1 - f_j$, where f_j is a random number that must be defined in p_j interval (the specified uncertainty degree) as bbm to the actual attribute value.

(2) f_j value as bbm to the frame of discernment corresponding to this attribute.

Thus, each *bba* has only 2 focal elements as defined by the special belief function SSF (see Chapter 1), where the focus is the actual attribute value by taking $w_j = f_j$. These resulting bba's describe our belief about the value of the actual attribute values which the object has.

A.3.2 Lenses database as example: From certain to belief version

By Table A.2, we detail one example of a data set namely lenses data as presented in (Murphy and Aha, 1996). Let us denote by:

- i: the index of the object X_i, it varies from 1 to 24.
- A_j: the attribute where j varies from 1 to 4, the column corresponding to each A_j represents the A_j's value taken by each instance. To simplify our representation, in the following table, each A_j will be denoted simply by j, $1 \leq j \leq 4$. So 1 for A_1, 2 for A_2, 3 for A_3, and 4 for A_4.
- C_i: the class to assign to the object X_i . Its values are 1, 2 or 3.

This database for fitting contact lenses is complete: its examples are complete too (no missings) and noise free. Indeed, its classes information are as follows respectively to three real class values:

1. The patient should be fitted with hard contact lenses.
2. The patient should be fitted with soft contact lenses.
3. The patient should not be fitted with contact lenses.

In other hand, the attribute information are detailed as follows:

- Age of the patient: (1) young, (2) pre-presbyopic, (3) presbyopic.
- Spectacle prescription: (1) myope, (2) hypermetrope.
- Astigmatic: (1) no, (2) yes.
- Tear production rate: (1) reduced, (2) normal.

Thus, this database is defined as follows:

Table A.2: Lenses database

i	1	2	3	4	C_i
1	1	1	1	1	3
2	1	1	1	2	2
3	1	1	2	1	3
4	1	1	2	2	1
5	1	2	1	1	3
6	1	2	1	2	2
7	1	2	2	1	3
8	1	2	2	2	1
9	2	1	1	1	3
10	2	1	1	2	2
11	2	1	2	1	3
12	2	1	2	2	1
13	2	2	1	1	3
14	2	2	1	2	2
15	2	2	2	1	3
16	2	2	2	2	3
17	3	1	1	1	3
18	3	1	1	2	3
19	3	1	2	1	3
20	3	1	2	2	1
21	3	2	1	1	3
22	3	2	1	2	2
23	3	2	2	1	3
24	3	2	2	2	3

We can see though this table that all possible combinations of attribute value pairs are represented.

The choice of taking Lenses data base as a basic example is due to its size, which is the smallest one among all considered UCI ones.

In Table A.3, we will detail the uncertain Lenses version which results from the artificial generation method. Thus, the uncertainty parameter values have been fixed besides the actual attribute values of lenses data set as follows:

- $[0, 0, 0.5, 0.75]$: each value represents the degree of uncertainty respectively to one attribute. The first and the second attributes have certain values since their uncertainty degrees are zero. Thus, only the two other attributes are uncertain.
- Uncertainty percentage: here, it may be specified as low percent, middle percent and high one. They are defined respectively in $]0, 35]$, $]35, 70]$ and $]70, 100]$. For

example by considering low percent, only 35% of instances are uncertain at maximum.

For each instance, within the modified lenses database, we present bbm's on the different possible attribute values of the power set of the corresponding Θ.
Let Θ_1, Θ_2, Θ_3 and Θ_4, be the four frames of discernment of the four lenses attributes. They are defined as follows:
$\Theta_1 = \{1, 2, 3\}$, $\Theta_2 = \{1, 2\}$, $\Theta_3 = \{1, 2\}$ and $\Theta_4 = \{1, 2\}$.
So, the corresponding power sets are:
$2^{\Theta_1} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \Theta_1\}$, $2^{\Theta_2} = \{\emptyset, \{1\}, \{2\}, \Theta_2\}$,
$2^{\Theta_3} = \{\emptyset, \{1\}, \{2\}, \Theta_3\}$ and $2^{\Theta_4} = \{\emptyset, \{1\}, \{2\}, \Theta_4\}$.

Note that this uncertain database is reported, here, as an artificial belief database example. In fact, the age of the patient and the spectacle prescription are perfectly known, while the astigmatic characteristic and the tear production rate are uncertain attributes.
Let us remind that i represents the index of the object X_i, it varies from 1 to 24. So, only 12 instances must be represented via bba's concept respectively to the middle uncertainty percentage already fixed at 50%.

Table A.3: Belief Lenses database

i	1	2	3				4				C_i
			$\emptyset$	$\{1\}$	$\{2\}$	Θ_3	$\emptyset$	$\{1\}$	$\{2\}$	Θ_4	
1	1	1	0	0.77	0	0.23	0	0.4	0	0.6	3
2	1	1	0	0.52	0	0.48	0	0	0.55	0.45	2
3	1	1	0	0	0.99	0.01	0	0.56	0	0.44	3
4	1	1	0	0	0.83	0.17	0	0	0.39	0.61	1
5	1	2	0	0.6	0	0.4	0	0.17	0	0.73	3
6	1	2	0	0.95	0	0.05	0	0	0.59	0.41	2
7	1	2	0	0	0.65	0.35	0	0.87	0	0.13	3
8	1	2	0	0	0.8	0.2	0	0	0.4	0.6	1
9	2	1	0	0.81	0	0.19	0	0.74	0	0.26	3
10	2	1	0	0.73	0	0.27	0	0	0.56	0.44	2
11	2	1	0	0	0.81	0.19	0	0.54	0	0.46	3
12	2	1	0	0	0.97	0.03	0	0	0.48	0.52	1
13	2	2	0	1	0	0	0	1	0	0	3
14	2	2	0	1	0	0	0	0	1	0	2
15	2	2	0	0	1	0	0	1	0	0	3
16	2	2	0	0	1	0	0	0	1	0	3
17	3	1	0	1	0	0	0	1	0	0	3
18	3	1	0	1	0	0	0	0	1	0	3
19	3	1	0	0	1	0	0	1	0	0	3
20	3	1	0	0	1	0	0	0	1	0	1
21	3	2	0	1	0	0	0	1	0	0	3
22	3	2	0	1	0	0	0	0	1	0	2
23	3	2	0	0	1	0	0	1	0	0	3
24	3	2	0	0	1	0	0	0	1	0	3

This belief version is among the ones used to test our proposals. Obviously, for performing simulations of Chapters 3, 4, 5 and 6, we have developed all belief versions for used UCI databases.

A.4 Evaluation criteria

The selected criteria, used to judge the performance of our proposed dynamic clustering methods under uncertainty, can be categorized into two kinds as follows:

1. Quality clustering indicator:
   It must be maximized or minimized depending on the chosen function. Indeed, we will compute, to each developed approach, the *clustering accuracy value* (Equation A.1), as external evaluation criterion, to judge our proposal, since the actual class label of all data sets items are known a priori.

In fact, Huang (Huang, 1998) proposed this measure of clustering results evaluation so-called the clustering accuracy r and computed as follows:

$$r = \frac{\sum_{l=1}^{k} a_l}{n} * 100 \tag{A.1}$$

where n is the number of instances in the data set, k is the number of clusters, a_l is the number of instances occurring in both cluster l and its corresponding labeled class. This criterion is equivalent to the PCC expressing the percentage of the correctly classified instances used within supervised classification context. If r is equal to 100%, then the cluster partition is judged as excellent, whereas a 'null' or bad cluster arrangement has a clustering accuracy equal to 0%.
An equivalent criterion, also used in the literature, measures the percent of incorrectly classified instances. This is known as the error rate $e = 100 - r$.

Let us remember that the clustering accuracy criterion is an external evaluation index which is computed based on an external information namely the actual class labels, as stored in UCI databases, and the generated cluster collection.

This same criterion has been used for the purpose of comparison to choose the best scheme among the static and dynamic methods. Therefore, it is also considered as a relative clustering evaluation criterion.

Generally speaking, cluster validation approaches are classified into three categories to investigate the output clustering quality (Jain and Dubes, 1988): internal, external and relative criteria. Typically, the resulting partitions are evaluated using the external criterion, but if the true labels are available. This is the same assumption for our thesis work.

For the internal criteria, it usually measures the intra-cluster homogeneity, the inter-cluster separability or a combination of these two. The compactness and the separation of clusters belong to this category. Moreover, clustering evaluation methods that adhere to external criterion compare the results of the clustering algorithm against some external benchmark. Such benchmarks consist of a set of pre-classified items, and these sets are often created by experts. On the other hand, a relative cluster evaluation measure is an internal or an external evaluation measure that is used for the purpose of comparison. Thus, relative measures are not actually a separate type of cluster evaluation measure.

A number of such indices have been developed and proposed in the literature (Halkidi et al., 2000; Sharma, 1996; Rezaee et al., 1998; Theodoridis and Koutroubas, 1999; Xie and Beni, 1991).

2. Time indicators:
   - Running time which represents the time computation needed to obtain a final stable partition. So, this criterion is used to find the faster classifier among the dynamic proposed ones and the static method. It is expressed in seconds.
   - Iteration number which indicates how many iterations are performed until the partition stability.

A.5 Conclusion

In this appendix, we have presented the different experimental settings needed to test and evaluate our developed proposals under uncertainty. From data sets description to uncertain versions creation where the uncertainty is represented via belief function concepts. We have detailed the belief Lenses database as an illustrative example. Then, both time and quality criteria are detailed. This latter considers one external evaluation index. All these advanced concepts are necessary to check our solutions respectively to Chapters 3, 4, 5 and 6.

Bibliography

Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of ACM SIGMOD International Conference on Management Of Data (SIGMOD'98)*, pages 94–105. ACM Press.

Ammar, A. and Elouedi, Z. (2011). A new possibilistic clustering method: The possibilistic K-Modes. In *Proceedings of the 12th international conference on Artificial intelligence AI*IA*, pages 413–419. Springer-Verlag. LNCS 6934.

Ammar, A., Elouedi, Z., and Lingras, P. (2012). K-Modes clustering using possibilistic membership. In *Proceedings of the 14th International Conference on Information Processing and Managment of Uncertainty in knowldge-based systems (IPMU), Part III*, pages 596–605. Springer-Verlag. LNSC 299.

Anderberg, M. (1973). *Cluster Analysis for Applications*. Academic Press, Inc., New York, NY.

Aranganayagi, S. and Thangavel, K. (2009). *M-Squeezer algorithm to cluster the categorical data*. Narosa, Publishing House, New Delhi, India.

Aranganayagi, S. and Thangavel, K. (2010). Incremental algorithm to cluster the categorical data with frequency based similarity measure. *International Journal of Information and Mathematical Sciences*, 6(1):21–29.

Bai, L., Liang, J., Dang, C., and Cao, F. (2012). A cluster centers initialization method for clustering ctegorical data. *Expert Systems with Applications*, 39(9):8022–8029.

Barnett, J. A. (1991). Calculating Dempster-Shafer plausibility. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):599–602.

Bauer, M. (1993). Approximations for efficient computation in the theory of evidence. *Artificial Intelligence*, 61(2):315–329.

Ben-Hariz, S. and Elouedi, Z. (2008). An incremental clustering approach within belief function framework. In *Proceedings of the Twelfth IASTED International Conference on Artificial Intelligence and Soft Computing*, pages 98–103. ACTA Press.

Ben-Hariz, S. and Elouedi, Z. (2010a). DK-BKM: Decremental k belief k-modes method. In *Proceedings of the 4th International Conference on Scalable Uncertainty Management (SUM)*, pages 84–97. Springer-verlag. LNCS 6379.

Ben-Hariz, S. and Elouedi, Z. (2010b). IK-BKM: An incremental clustering approach based on intra-cluster distance. In *Proceedings of the 11th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE.

Ben-Hariz, S. and Elouedi, Z. (2010c). IK-BKM: Méthode K-modes crédibiliste incrémentale. In Édition, C., editor, *Actes des rencontres francophones sur la Logique Floue et ses Applications (LFA 2010)*, pages 168–175.

Ben-Hariz, S. and Elouedi, Z. (2011). Ranking-based feature selection method for dynamic belief clustering. In *Proceedings of the 2001 International Conference on Adaptive and Intelligent Systems (ICAIS11)*, pages 308–319. Springer-Verlag. LNAI 6943.

Ben-Hariz, S. and Elouedi, Z. (2014). New Dynamic Clustering Approaches within Belief Function Framework. *Intelligent Data Analysis*, 18(2). To appear.

Ben-Hariz, S., Elouedi, Z., and Mellouli, K. (2006). Clustering approach using belief function theory. In *Proceedings of the 12th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA)*, pages 162–171. Springer-Verlag. LNCS 4183.

Ben-Hariz, S., Elouedi, Z., and Mellouli, K. (2008). Selection initial modes for belief k-modes method. *International Journal of Applied Science, Engineering and Technology (IJASET)*, 4(4):233–242.

Beringer, J. and Hüllermeier, E. (2007). Adaptive optimization of the number of clusters in fuzzy clustering. In *Proceedings of IEEE International Conference on Fuzzy Systems (Fuzz-IEEE)*, pages 1–6. IEEE.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press.

Bingham, E. and Heikki, M. (2001). Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 245–250. ACM.

Bordogna, G., Pagani, M., and Pasi, G. (2006). A Dynamic hierarchical fuzzy clustering algorithm for information filtering. In *Studies in Fuzziness and Soft Computing*, pages 3–23. Springer. LNCS 206.

Boukharouba, K. and Lecoeuche, S. (2008). Online clustering of non-stationary data using incremental and decremental SVM. In *Proceedings of 18th International Conference of Artificial Neural Networks (ICANN)*, pages 336–345. Springer-Verlag. ISBN: 978-3-540-87558-1.

Bradley, P. S., Fayyad, U., and Reina, C. (1999). Scaling EM (Expectation-Maximization) clustering to large databases. Technical report, Microsoft Research, Seattle.

Bradley, P. S. and Fayyad, U. M. (1998). Refining initial points for k-means clustering. In *Proceedings of the Fifteenth International Conference on Machine Learning(ICML)*, pages 91–99. Morgan Kaufmann.

Câmpan, A. and Serban, G. (2006). Adaptive clustering algorithms. In *Proceedings of the 19th international conference on Advances in Artificial Intelligence: Canadian Society for Computational Studies of Intelligence*, pages 407–418. Springer-Verlag.

Cao, F., Liang, J., and Bai, L. (2009). A new initialization method for categorical data clustering. *Expert Systems with Applications*, 36(7):10223–10228.

Choi, S., Cha, S., and Tappert, C. C. (2010). A Survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48.

Crespoa, F. and Weberb, R. (2005). A methodology for dynamic data mining based on fuzzy clustering. *Fuzzy Sets and Systems*, 150(2):267–284.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38(2):325–339.

Denoeux, T. (1995). A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems Man and Cybernetics*, 25(5):804–813.

Denoeux, T. (2000). A neural network classifier based on Dempster-Shafer theory. *IEEE Transactions on Systems Man and Cybernetics - Part A*, 30(2):131–150.

Denoeux, T. and Masson, M. (2002). Clustering of proximity data using belief functions. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems of (IPMU)*, pages 609–616. ESIA-Université de Savoie.

Denoeux, T. and Masson, M. (2004). EVCLUS: Evidential Clustering of Proximity Data. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 34(1):95–109.

Dubois, D. and Prade, H. (1988). Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence*, 4(3):244–264.

Duran, B. and Odell, P. (1974). *Cluster Analysis: A Survey.* Springer-Verlag.

Elouedi, Z., Mellouli, K., and Smets, P. (2001). Belief decision trees: Theoretical foundations. *International Journal of Approximate Reasoning (IJAR)*, 28(2-3):91–124.

Elouedi, Z., Mellouli, K., and Smets, P. (2004). Assessing sensor reliability for multisensor data fusion within the transferable belief model. *IEEE Trans Syst Man Cybernetics, Part B*, 34(1):782–787.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). 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 (KDD)*, pages 226–231.

Fixen, D. and Mahler, R. P. S. (1997). The modified Dempster-Shafer approach to classification. *IEEE Transactions on Systems, Man, and Cybernetics - Part A*, 27(1):96–104.

Ghahramani, Z. and Jordan, M. (1994). Learning from incomplete data. Technical report, Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Green, P. D., Barker, J., Cooke, M., and Josifovski, L. (2001). Handling missing and unreliable information in speech recognition. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, pages 49–56.

Guha, S., Rastogi, R., and Shim, K. (1999). Rock: A robust clustering algorithm for categorical attributes. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 512–521. IEEE.

Halkidi, M., Vazirgiannis, M., and Batistakis, Y. (2000). Quality scheme assessment in the clustering process. In *Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, pages 265–276. Springer-Verlag. LNCS 1910.

He, J., Lano, M., Tan, C.-L., Sung, S.-Y., and Low, H.-B. (2004). Initialization of cluster refinement algorithms: a review and comparative study. In *Proceedings*

of International Joint Conference on Neural Networks (IJCNN), pages 297–302. IEEE.

He, Z. (2006). Farthest-Point Heuristic based Initialization Methods for K-Modes Clustering. *The Computing Research Repository (CoRR)*, abs/cs/0610043.

Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the First Pacific Asia Knowledge Discovery and Data Mining Conference*, pages 21–34. World Scientific.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining Knowledge Discovery*, 2(2):283–304.

Huang, Z. and Ng, M. (1999). A fuzzy K-modes algorithm for clustering categorical data. *IEEE Transaction on Fuzzy Systems*, 7(4):446–452.

Jain, A. and Dubes, R. (1988). *Algorithms for clustering data*. Prentice-Hall, Englewood cliffs, NJ.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323.

Janez, F. (1996). *Fusion de sources d'information définies sur des référentiels non exhaustifs différents*. PhD thesis, Université d'Angers, France.

John, G., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 121–129. Morgan Kaufmann.

Jousselme, A.-L., Grenier, D., and Bossé, E. (2001). A new distance between two bodies of evidence. *Information Fusion*, 2(2):91–101.

Kaufman, L. and Rousseeuw, P. (1990). *Finding groups in data: An Introduction to Cluster Analysis*. Wiley-Interscience.

Khan, S. and Kant, S. (2007). Computation of initial modes for k-modes clustering algorithm using evidence accumulation. In *Proceedings of the 20th international joint conference on Artificial intelligence (IJCAI)*, pages 2784–2789.

Krishnapuram, R. and Keller, J. (1993). A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(2):98–110.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. Berkeley, Calif.: University of California Press.

Masson, M. and Denoeux, T. (2008). ECM: An evidential version of the fuzzy c-means algorithm. *Pattern Recognition*, 41(4):1384–1397.

Mellouli, K. (1987). *On the propagation of beliefs in networks using the Dempster-Shafer theory of evidence.* PhD thesis, School of business, University of Kansas.

Murphy, K. C. (2000). Combining belief functions when evidence conflicts. *Decision Support Systems*, 29(1):1–9.

Murphy, P. and Aha, D. (1996). Uci repository databases. http://www.ics.uci.edu/m̃learn/.

Peters, G., Weber, R., and Crespo, F. (2010). Uncertainty modeling in dynamic clustering - A soft computing perspective. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ'IEEE)*, pages 1–6. IEEE.

Quost, B. and Denoeux, T. (2010). Clustering fuzzy data using the fuzzy EM algorithm. In A. Deshpande, A. H., editor, *Proceedings of the International Conference on Scalable Uncertainty Management (SUM)*, pages 333–346. Springer-Verlag. LNAI 6379.

Ralambondrainy, H. (1995). A conceptual version of the K-means algorithm. *Pattern Recognition Letters*, 15(11):1147–1157.

Rezaee, M., Lelieveldt, B., and Reiber, J. (1998). A new cluster validity index for the fuzzy c-mean. *Pattern Recognition Letters*, 19(3-4):237–246.

San, O. M., Huynh, V., and Nakamori, Y. (2004). An alternative extension of the k-means algorithm for clustering categorical data. *International Journal of Applied Mathematics and Computer Science*, 14(2):241–247.

Serban, G. and Câmpan, A. (2005). Incremental Clustering Using a Core-Based Approach. In *Proceedings of the 20th international conference on Computer and Information Sciences (ISCIS'05)*, pages 854–863. Springer-Verlag. LNAI 3733.

Shafer, G. (1976). *A mathematical theory of evidence.* Princeton Univ. Press.

Shafer, G. (1986). The combination of evidence. *International Journal of Intelligent Systems*, 1(3):155–179.

Sharma, S. (1996). *Applied multivariate techniques.* Wiley and Sons.

Smets, P. (1988). Belief functions. In *Non standard logics for automated reasoning*, pages 253–286. Academic Press.

Smets, P. (1990a). The combination of evidence in the transferable belief model. *IEEE-Pattern analysis and Machine Intelligence*, 12(5):447–458.

Smets, P. (1990b). The transferable belief model and other interpretations of Dempster-Shafer's model. In *Uncertainty in Artificial Intelligence 6 Annual Conference on Uncertainty in Artificial Intelligence (UAI-90)*, pages 375–384. Elsevier.

Smets, P. (1992a). The concept of distinct evidence. In *Proceedings of the International conference on Information Processing and Managment of Uncertainty in Knowldge-based Systems (IPMU'92)*, pages 789–794. Springer-Verlag. LNCS 682.

Smets, P. (1992b). The nature of the unnormalized beliefs encountered in the transferable belief model. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 292–297. Morgan Kaufmann.

Smets, P. (1993). Belief functions: The disjunctive rule of combination and the generalized bayesian theorem. *International Journal of Approximate Reasoning*, 9(1):1–35.

Smets, P. (1995). The canonical decomposition of a weighted belief. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1896–1901. Morgan Kaufmann.

Smets, P. (1998). The transferable belief model for quantified belief representation. In *Handbook of Defeasible Reasoning and Uncertainty management systems*, pages 267–301. Kluwer Academic Publishers.

Smets, P. (2002). Decision making in a context where uncertainty is represented by belief functions. In *Belief functions in business decisions*, pages 17–61. Physica-Verlag.

Smets, P. and Kennes, R. (1994). The transferable belief model. *Artificial Intelligence*, 66:191–234.

Smets, P. and Kruse, R. (1997). The transferable belief model for belief representation. In *Uncertainty in Information Systems: From needs to solutions*, pages 343–368. Kluwer Academic Publishers.

Su, X., Y. Lan, R. W., and Qin, Y. (2009). A fast incremental clustering algorithm. In *Proceedings of the International Symposium on Information Processing*, pages 175–178. Academy Publisher.

Sun, Y., Zhu, Q., and Chen, Z. (2002). An iterative initial-points refinement algorithm for categorical data clustering. *Pattern Recognition Letters*, 23(7):875–884.

Tessem, B. (1997). Approximation algorithms and decision making in the Dempster-Shafer theory of evidence-an empirical study. *International Journal of Approximate Reasoning*, 17(2-3):217–237.

Theodoridis, S. and Koutroubas, K. (1999). *Pattern Recognition*. Academic Press.

Trabelsi, S., Elouedi, Z., and Lingras, P. (2011). Classification systems based on rough sets under the belief function framework. *International Journal of Approximation Reasoning*, 52(9):1409–1432.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17:520–525.

Truta, T. and Campan, A. (2007). K-anonymization incremental maintenance and optimization techniques. In *Proceedings of the Symposium on Applied Computing*, pages 380–387. ACM Press.

Wagstaff, K. (2004). Clustering with missing values: No imputation required. In *Classification, Clustering, and Data Mining Applications (Proceedings of the Meeting of the International Federation of Classification Societies*, pages 649–658. Springer. LNAI 1431.

Walley, P. (1991). *Statistical reasoning with imprecise probabilities*. Chapman and Hall, London.

Wang, W., Yang, J., and Muntz, R. (1997). STING: A statistical information grid approach to spatial data mining. In *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB97)*, pages 186–195. Morgan Kaufmann.

Williams, W. and Lambert, J. (1966). Multivariate methods in plant ecology: V. similarity analyses and information-analysis. *Journal of Ecology*, 54(2):427–445.

Wu, S., Jiang, Q., and Huang, J. Z. (2007). A new initialization method for clustering categorical data. In *Proceedings of the 11th Pacific-Asia conference on Advances in knowledge discovery and data mining, PAKDD'07*, pages 972–980. Springer-Verlag. LNCS 7301.

Xie, X. L. and Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 13(8):841–847.

Yager, R. (1987). On the Dempster-Shafer framework and new combination rules. *Information Sciences*, 41:93–138.

Zouhal, L. and Denoeux, T. (1998). An evidence-theory k-NN rule with parameter optimization. *IEEE Trans.Syst.Man Cybern. C*, 28(2):263–271.