FUZZY REASONING BASED EVOLUTIONARY ALGORITHMS

APPLIED TO DATA MINING

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Min Chen

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Computer Science

May 2015

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Fuzzy Reasoning based Evolutionary Algorithms Applied to Data Mining

By

Min Chen

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Simone A. Ludwig
Chair

Dr. Saeed Salem

Dr. Changhui Yan

Dr. Maria Alfonseca-Cubero

Approved:

05/04/2015
Date

Dr. Brian M. Slator
Department Chair

ABSTRACT

Data mining and information retrieval are two difficult tasks for various reasons. First, as the volume of data increases tremendously, most of the data are complex, large, imprecise, uncertain or incomplete. Furthermore, information retrieval may be imprecise or subjective. Therefore, comprehensible and understandable results are required by the users during the process of data mining or knowledge discovery. Fuzzy logic has become an active research area because its capability of handling perceptual uncertainties, such as ambiguity or vagueness, and its excellent ability on describing nonlinear system.

The study of this dissertation is focused on two main paradigms. The first paradigm focuses on applying fuzzy inductive learning on classification problems. A fuzzy classifier based on discrete particle swarm optimization and a fuzzy decision tree classifier are implemented in this paradigm. The fuzzy classifier based on discrete particle swarm optimization includes a discrete particle swarm optimization classifier and a fuzzy discrete particle swarm optimization classifier. The discrete particle swarm optimization classifier is devised and applied to discrete data. Whereas, the fuzzy discrete particle swarm optimization classifier is an improved version that can handle both discrete and continuous data to manage uncertainty and imprecision. A fuzzy decision tree classifier with a feature selection method is proposed, which is based on the ideas of mutual information and genetic algorithms.

The second paradigm is fuzzy cluster analysis. The purpose is to provide efficient approaches to identify similar or dissimilar descriptions of data instances. The shapes of the clusters is either hyper-spherical or hyper-planed. A fuzzy c-means clustering approach based on particle swarm optimization, which clustering prototype is hyper-spherical, is proposed to automatically determine the optimal number of clusters. In addition, a fuzzy c-regression model, which has hyper-planed clusters, has received much attention in recent literature for nonlinear system identication and has been successfully employed in various areas. Thus, a fuzzy c-regression model clustering algorithm is applied for color image segmentation.

ACKNOWLEDGMENTS

I am very grateful to take this opportunity to thank Dr. Simone A. Ludwig, Dr. Saeed Salem, Dr. Changhui Yan and Dr. Maria A. Alfonseca-Cubero for serving on my graduate committee. First, I would like to express my greatest appreciation to my major advisor, Dr. Simone A. Ludwig. Not only did she guide me through the dissertation process, but also taught me the overall process of doing research. Everyone should be a fortunate as I was to have an advisor such as her. I cannot thank her enough for the supervision, encouragement and inspiration in developing the theory, conducting the experiment, publishing papers, and writing this dissertation. She has shown me the joys of teaching, rewards of research and importance of life, and for this I thank her.

I would like to also thank Dr. Saeed Salem for serving on my graduate committee. I have always enjoyed discussing many aspects of this research with him and look forward to future discussions. He have dedicated many hours on supporting my graduate study and providing helps on my professional developments.

I would like to thank Dr. Changhui Yan and Dr. Maria A. Alfonseca-Cubero for serving on my graduate committee. With all of the other commitments to the university, both dedicated time to take part in my dissertation research and provided valuable insight into the completion of my dissertation.

Finally, I would like to thank my family, friends for their support, understanding, and encouragement that I have the opportunity to pursue my Ph.D.. I certainly wasn't the easiest person to be around during the last year while I worked on this research and completed my dissertation. I want to thank Di Wang for his unwavering support allowed me the time to complete this dissertation. I am appreciative of my mother, Xiongjuan Wen, for her love, sacrifice, and support. Without her support, this work may never have been completed.

DEDICATION

To my beloved family for all your support along the way.

TABLE OF CONTENTS

ABSTRACT . . . iii

ACKNOWLEDGMENTS . . . iv

DEDICATION . . . v

LIST OF TABLES . . . x

LIST OF FIGURES . . . xii

1. INTRODUCTION . . . 1

1.1. Data Mining . . . 1

1.2. Swarm Intelligence . . . 2

1.3. Fuzzy Rule Classification . . . 3

1.3.1. Fuzzy Decision Tree . . . 4

1.4. Fuzzy Clustering . . . 5

1.5. Motivation and Problem Statement . . . 6

1.6. Contributions . . . 8

1.7. Dissertation Overview . . . 10

2. A FUZZY DISCRETE PARTICLE SWARM OPTIMIZATION CLASSIFIER FOR RULE CLASSIFICATION . . . 12

2.1. Related Work . . . 12

2.2. Proposed Approaches . . . 14

2.2.1. Discrete Particle Swarm Optimization with Local Strategy . . . 15

2.2.2. Definition of Overall Fitness . . . 17

2.2.3. Local Mutation Strategy . . . 18

2.2.4. DPSO-LS Classifier 18

2.2.5. FDPSO-LS Classifier 20

2.3. Experiments and Results 23

2.3.1. Experimental Setup 24

2.3.2. Results of the DPSO-LS Approach 24

2.4. Comparison and Results of DPSO-LS and FDPSO-LS for Discrete Data Sets 28

2.5. Results of FDPSO-LS Approach for Continuous Data Set 30

2.6. Summary 33

3. FUZZY DECISION TREE USING SOFT DISCRETIZATION AND A GENETIC ALGORITHM BASED FEATURE SELECTION METHOD 35

3.1. Related Work 35

3.2. Fuzzy Decision Tree Classifier 36

3.2.1. Preprocessing Phase 38

3.3. Experiments and Results 41

3.3.1. Experimental Setup 41

3.3.2. Experimental Results 42

3.4. Summary 46

4. PARTICLE SWARM OPTIMIZATION BASED FUZZY CLUSTERING APPROACH TO IDENTIFY OPTIMAL NUMBER OF CLUSTERS 48

4.1. Related Work 48

4.2. Fuzzy C-Means and Particle Swarm Optimization 50

4.2.1. Fuzzy C-means Clustering 50

4.2.2. Particle Swarm Optimization 51

4.3. Proposed Approach 52

4.3.1. Particle Encoding 52

4.3.2. Velocity Encoding 52

4.3.3. Decoding 53

4.3.4. Clustering Validation Techniques 53

4.4. Experiments and Results 57

4.4.1. Experimental Setup 57

4.4.2. Datasets 57

4.5. Experimental Study 58

4.5.1. Use of Synthetic Data 58

4.5.2. Use of Real-World Data 61

4.5.3. Visualization of Clustering Results 62

4.6. Summary 65

5. COLOR IMAGE SEGMENTATION USING FUZZY C-REGRESSION MODEL . . 67

5.1. Related Work 67

5.2. Proposed Approach 71

5.2.1. CIE-L*A*B* Color Space 71

5.2.2. Fuzzy C-Regression Model Clustering 71

5.3. Clustering Validation Techniques 74

5.4. Experiments and Results 75

5.5. Experimental Setup 75

5.6. Experimental Study 75

5.6.1. Comparison of cluster performance in *A*B space 75

5.6.2. Comparison using different validity indices 76

5.6.3. Comparison with MSE and PSNR 78

5.6.4. Comparison on segmentation results 88

5.7. Summary 88

6. CONCLUSION AND FUTURE WORK 90

6.1. Conclusions 90

6.2. Future Work 92

BIBLIOGRAPHY 95

LIST OF TABLES

Table Page

1. Parameters and their values of the DPSO and DPSO-LS algorithms. 25

2. Data sets used for the experiments. 25

3. Average accuracy of DPSO and DPSO-LS for 100 iterations. 26

4. Average rule size of all algorithms. 27

5. Average weighted TPRs (%) of all algorithms. 27

6. Average weighted FPRs (%) of all algorithms. 28

7. Average weighted precision (%) of all algorithms. 28

8. Average accuracy of DPSO-LS and FDPSO-LS for 100 iterations.. 28

9. Average rule size of DPSO-LS and FDPSO-LS for 10 runs. 29

10. Average weighted TPRs of DPSO-LS and FDPSO-LS for 10 runs. 30

11. Average weighted FPRs of DPSO-LS and FDPSO-LS for 10 runs. 30

12. Average weighted precision of DPSO-LS and FDPSO-LS for 10 runs. 31

13. Datasets used for the proposed fuzzy rule-based system using DPSO-LS. 31

14. Average accuracy of FURIA and FDPSO-LS for 100 iterations. 32

15. Average rule size of FDPSO-LS and FURIA. 33

16. Average weighted TPRs of FDPSO-LS and FURIA. 33

17. Average weighted FPRs of FDPSO-LS and FURIA. 34

18. Average weighted precision of FDPSO-LS and FURIA. 34

19. GA parameters of GA-based feature selection method. 42

20. Datasets used for experiments. 43

21. Average weighted TPR (%) of all algorithms. 44

22. Average weighted FPR (%) of all algorithms. 44

23. Average weighted precision (%) of all algorithms. 44

24. Parameters and their values of the proposed algorithm. 57

25. Datasets used for the experiments. 58

26. Index values with varying c using pinwheel data set. 60

27. Reconstruction error with varying c using transfusion data set. 64

28. Nine different indices using the proposed algorithm. 65

29. Mapping using the proposed algorithm. 66

30. Parameters and their values of the FCM, GK, and FCRM algorithms 76

31. FCM, GK, and FCRM using three different indices (PC, PE and MPC) 76

32. FCM, GK, and FCRM using three different indices (PC, PE and MPC) 77

33. Best cluster number of FCM, GK and FCRM using PC, PE and MPC 78

34. MSE ($\times 10^4$) using FCM with different cluster number 79

35. MSE ($\times 10^4$) using GK with different cluster number 80

36. MSE ($\times 10^4$) using FCRM with different cluster number 81

37. PSNR using FCM with different cluster number 82

38. PSNR using GK with different cluster number 83

39. PSNR using FCRM with different cluster number 84

LIST OF FIGURES

Figure Page

1. Processes of DPSO-LS based classifiers. 19

2. An example of fuzzy partition. 21

3. Fuzzy inference process. 23

4. Accuracy of all algorithms. 26

5. Accuracy comparison of DPSO-LS and FDPSO-LS. 29

6. Accuracy comparison of the proposed FDPSO-LS and FURIA. 32

7. A fuzzy discretization process. 38

8. AUC of J48 and J48-P. 45

9. AUC of REPTree and REPTree-P. 46

10. AUC of FURIA and FURIA-P. 46

11. AUC of FDT and FDT-P. 47

12. Synthetic data set 59

13. Kmeans using 3 different validity indices 59

14. K-medoid using 3 different validity indices 60

15. FCM using 9 different validity indices 61

16. GK using 9 different validity indices 62

17. FPSO using 9 different validity indices 63

18. Data sets using PCA projection, CSM and FSM, respectively. 64

19. Original image, FCM, GK, and FCRM with c = 3 in *a*b color space 76

20. Original image, grayscale image using FCM, GK, and FCRM are listed, respectively 85

1. INTRODUCTION

In this current information age, a tremendous expansion in the volume of data is seen that is being generated and stored. It is possible to collect, store, transfer and combine huge amounts of data at very low costs. However, only a small amount of the data has been used. It turns out to be very difficult to exploit information in the data in an intelligent way. The primary reasons includes the volumes of data are too large to manage, the data structures are too complicated to be analyzed and there is a lack of tools that can efficiently and effectively analyze and reveal valuable knowledge that is hidden.

The need to understand large, complex, information-rich data sets is common to all fields of studies. The objective of the field of knowledge discovery and data mining is the discovery of knowledge that is not only correct, but also comprehensible. This chapter briefly describes the background to the research topics investigated in this dissertation, brief descriptions of the background are introduced in Section 1.1-1.4. The motivation of the work is discussed in Section 1.5. The contributions of the work is listed in Section 1.6 and the structure of the dissertation is described in Section 1.7.

1.1. Data Mining

Data mining is called exploratory data analysis, among other things. It is an analytic process designed to explore data. Data mining aims to search for consistent patterns or systematic relationships between variables. It then validates the findings by applying the detected patterns to new subsets of data [1]. It is a statistical analysis process which can identify the clusters along with collection of data. Data mining can be achieved by classification, association, prediction, sequential pattern, similar time sequences and clustering [2].

The data mining tasks can be classified as unsupervised or supervised learning. Unsupervised learning focuses on finding patterns describing the data that can be interpreted. On the other hand, supervised learning involves using some features or fields of the data set to predict unknown or future values of interest.

The two primary goals of data mining can be classified as *prediction* and *description* [3]. *Prediction* involves using some features or fields of the data set to predict unknown or future values of interest, whereas *description* focuses on finding patterns describing the data that can be interpreted by humans. Several data mining techniques using prediction and description have emerged that include classification, clustering, regression, dependence modeling, etc.

The classification technique is used to discover a predictive learning function that classifies a data item into several predefined classes. It is also known as supervised classification, whereby given class labels are ordered to objects in the data collection. In general, classification approaches use a training set in which all objects are already associated with their corresponding class labels. The classification algorithm then learns from the training set data and builds a model. This model is then used to classify unseen data and to assign a class label to each data item.

1.2. Swarm Intelligence

Swarm Intelligence (SI) is a field of computer science, which is inspired by the behavior of real swarms, flocks, insect colonies to design and study efficient computational methods for solving problems [4]. Two main areas of swarm intelligence are Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). Ant colony optimization [5] is inspired by the real ants finding shortest paths from their nest to the food sources. Ants mark their paths to the food sources via a pheromone trail along their way. Other ants can be led to the food source by the pheromone traces.

The coordinated search of food for bird flocking can be modeled with simple rules for information sharing between individuals of the swarm. Based on the analogy of the behavior of flocks of birds, Kennedy and Eberhart [6] developed a method for function optimization referred to as *particle swarm optimization*. A particle swarm optimization algorithm includes a population of particles denoted as the swarm. Each individual is called a particle that represents a location in the problem space. Each particle starts at a random location with a velocity and searches for the optimum of a given objective function by moving through the search space. The movements of each particle depends on its velocity and the positions where good solutions have already been found by the particle, named as personal best, or by other particles in the swarm, called global best.

Particle swarm optimization is a population-based optimization tool which is mainly applied to solve various function optimization problems. Compare to Genetic Algorithm (GA) and Simulated Annealing (SA), the main strength of PSO is the fast convergence. Typically, each particle keeps track of the coordinates in the search space, which are associated with the personal best and the global best it has found so far. In each iteration, the velocity of the PSO algorithm is changed towards the personal and global best with some random component. Though the main use of PSO is for continuous function optimization, an increasing number of works have investigated the use of discrete PSO to be applied to more complex discrete problems.

1.3. Fuzzy Rule Classification

Rule discovery is an important classification method that has been attracting a significant amount of researchers in recent years. It uses a set of IF-THEN rules to classify a class or category in a natural way. A rule consists of antecedents and a consequent. The antecedents of the rule consist of a set of attribute values and the consequent of the rule is the class which is predicted by that rule.

One possible application of fuzzy logic in data mining is the induction of fuzzy rules to interpret the underlying data linguistically. Fuzzy logic can improve the classification system by using fuzzy sets to define overlapping class definitions [7]. The interpretability of the results can be improved and more insight into the classifier structure and decision making process is provided by the application of fuzzy IF-THEN rules [8]. Fuzzy rules are linguistic IF-THEN constructs that have the general form “IF A THEN C”, where A and C are collections of propositions and postpositions containing linguistic variables. A is called the antecedent, and C is the consequent of the rule. In effect, the tolerance for imprecision and uncertainty is exploited through granulation in soft data compression by using linguistic variables and fuzzy IF-THEN rules [8]. In this respect, fuzzy logic has the feature of mimicking the essential ability of the human mind to summarize data and focus on decision-relevant information. In a more explicit form, the i^{th} rule has the following form:

$$\text{IF } x_{i1} \in A_1^m \text{ AND .. AND } x_{ij} \in A_j^n \text{ THEN } c_i \in C_i^k \tag{1.1}$$

where x_{ij} denotes the j^{th} attribute of the i^{th} rule. A_j^m denotes the m^{th} antecedent value of the j^{th} attribute. c_i is the consequent of the i^{th} rule.

1.3.1. Fuzzy Decision Tree

Decision Tree (DT) mining is one of the frequently used classification methods that specify the sequences of decisions that need to be made accompanied by the resulting recommendation. DT mining typically uses a top-down strategy, and the measure of information gain is used as a "goodness" criterion. DTs are intrinsic multi-class learners that scale comparatively well, sometimes even outperforming other state-of-the-art methods especially when they are used as part of an ensemble method [9, 10]. DTs are comprehensible and interpretable and can handle different types of attributes (e.g., numerical and categorical) [11]. Popular methods of decision trees are ID3 [12], C4.5 [13] and CART [14], which generate a tree structure through recursively partitioning the attribute space until the whole decision space is completely partitioned into a set of non-overlapping subspaces [15], which is also called hard discretization. Soft discretization on the other hand is when the decision space is partitioned into a set of overlapping subspaces. The classical crisp discretization can cause low classification accuracy since it can not analyze noisy data using crisp cut points. Furthermore, crisp discretization can lead to misclassification of new objects, which are close to the separating boundary between decision classes [16].

Researchers have attempted to combine some elements of symbolic and sub-symbolic approaches to decision tree induction. The fuzzy approach is one of such extensions. Due to its ability of handling vagueness, ambiguity and reduction of complexity, fuzzy logic [7],[8] has been widely applied in dealing with problems of uncertainty, noise, and inexact data. A DT induction method using fuzzy set theory, in other words, Fuzzy Decision Tree (FDT), is becoming an increasingly popular method to solve classification problems. FDT, like classical DT, uses the top-down strategy. In order to find the best so called “cut-point”, FDT is based on soft discetization and follows the DT run recursively on each partition until the best cut point is found.

1.4. Fuzzy Clustering

Clustering analysis is one of the popular approaches and has been widely used in data mining. Clustering analysis is a process to identify groups or clusters based on some similarity measures. Most clustering algorithms can be categorized into two popular techniques known as hierarchical and partitional clustering. The output of the hierarchical clustering is a tree showing a sequence of clusters with each cluster being a partition of the data set. Hierarchical clustering does not specify the number of clusters, and the output is independent of the initial condition. However, the hierarchical clustering is static, i.e., the data points assigned to a cluster cannot be reassigned to another cluster. In addition, hierarchical clustering will fail to separate overlapping clusters due to the lack of information regarding the global shape or size of the clusters. On the other hand, partitioned clustering requires a fixed number of clusters to be specified a priori. Objective functions such as square error function are used as a criteria in the optimization process of the data partitioning. Partitioned clustering uses an iterative process to optimize the cluster centers, as well as the number of clusters. However, it is a challenge to find the “optimum” number of clusters since it always requires prior knowledge about the data. The advantages of hierarchical algorithms are the disadvantages of the partitional algorithms and vice versa.

The goal of clustering involves the task of dividing data points into homogeneous groups such that the data points in the same group are as similar as possible and data points in different groups are as dissimilar as possible [17, 2]. The importance of clustering is documented in pattern recognition [18], machine learning, image analysis [19], information retrieval, etc. Depending on whether a data point belongs to a single cluster or several clusters with different membership degrees, clustering methods can be categorized as either hard clustering [20, 21] or fuzzy clustering [22]. Each data point of the data set belongs to exactly one cluster in hard clustering. Fuzzy set theory which was proposed by Zadeh [7] in 1965 is used to describe the membership degrees in fuzzy cluster analysis. Therefore, each data point of the data set belongs to two or more clusters with a membership degree between 0 and 1. Due to the capacity of handling uncertainty and vagueness, the potential of fuzzy clustering to reveal the underlying structures in data with regard to similarities or dissimilarities among them can be exploited [23].

One of the widely used methods in fuzzy clustering is Fuzzy C-Means clustering (FCM) [24]. The FCM method attempts to partition a data set into a collection of c fuzzy groups. It finds a cluster center in each group such that the intra-distance within the group is minimized and the inter-distance between each group is maximized. All of the fuzzy clustering methods that have been applied recently mostly use an extension of the FCM algorithm. As we have discussed before, partitional clustering suffers from the following two drawbacks:

- The number of clusters needs to be specified in advance. Furthermore, it requires prior knowledge or ground truth of the data.

- In most cases, data points in overlapping areas can not be categorized correctly.

1.5. Motivation and Problem Statement

Fuzzy logic and fuzzy set theory which were proposed by Zadeh [7, 8], have been widely used in pattern recognition and fuzzy reasoning. Fuzzy logic, which works with reasoning rules, is very close to the human way of thinking. Unlike classical logic, fuzzy logic allows us to define values without specifying a precise value, which is not possible in classical logic.

In addition, fuzzy set theory makes it possible that an object can belong to one or more sets with a certain degree. The interpretations of membership degrees are in terms of similarity, preference, and uncertainty. In other words, the use of membership degrees can state how similar an object is to a prototypical one, indicate preferences between suboptimal solutions to a problem, and model uncertainty about the true situation. Generally, fuzzy reasoning is close to human reasoning. The solution obtained using fuzzy approaches are easy to understand and to apply.

Specially, fuzzy systems are the method of choice when linguistic, vague, or imprecise information has to be modeled.

As data is accumulated at an unimaginable rate from a very wide variety of sources, the difficulty of efficient analysis of the large amount of data is growing. Data mining, which is a part of the Knowledge Discovery in Databases, is one of the main solutions for this problem. Soft computing [25] techniques such as fuzzy theory, soft set, evolutionary algorithms and neural networks have been successfully applied to data mining. Soft computing uses hybridization of several computing paradigms such as fuzzy logic, neural networks and genetic algorithm. Soft computing is more suitable for real world problems due to its tolerance of imprecision and the ability of solving real world problems in reasonably less time. Fuzzy logic, which is one of the principal constituents of soft computing, provides a useful mechanism for data mining or knowledge discovery.

The main motivation of this research can be summarized as follows.

1. Fuzzy rule-based classification is one of the most popular approaches used for classification problems. The key motivation for capturing data behavior in the form of fuzzy rules is that fuzzy rules are easy to understand, verify, and extend. The fuzzy rule-based system is comprehensible because each fuzzy rule is linguistically interpretable. However, it is a challenge to automatically generate fuzzy rules from the data. In order to keep the resulting rule base small and comprehensible, both classification performance and interpretability are important. For this purpose, a PSO algorithm is used to develop a fuzzy classifier in an iterative approach.

2. Fuzzy Decision Trees enable the user to take into account imprecise description or heterogeneous values in data mining. The key feature of FDT is the interpretability. The rules obtained by FDT make it easier for the user to interact with the system. FDT have been extensively used in recent years. However, feature selection in FDT is very compu-

tationally expensive since joint entropy has to be calculated requiring the estimation of the joint probability distributions. In order to reduce the computational complexity, a variable selection based on genetic algorithm is proposed to address the combinatorial checking of the variables.

3. Contrary to fuzzy-ruled based classification and fuzzy decision tree methods, fuzzy clustering is an unsupervised learning technique. Unlike traditional clustering, a point is assigned to a single cluster. The use of fuzzy set theory allows a point to be assigned to two or more clusters. The Fuzzy C-means algorithm is one of the popular methods applied in fuzzy clustering. However, the number of clusters in fuzzy c-means needs to be specified in advance. A clustering approach based on Particle Swarm Optimization that automatically determines the optimal number of clusters is proposed.

4. Due to the variety and complexity of images, image segmentation is still a very challenging research topic. Various techniques have been introduced for object segmentation and feature extraction. Although fuzzy c-means can partition the fuzzy space efficiently, it does not take linearity of the divided data into consideration. In contrast, the fuzzy c-regression model clustering algorithm with hyperplane-shaped cluster prototypes returns results that have much more explanatory power, especially due to its multivariate nature. An unsupervised approach using the fuzzy c-regression model is applied and proposed for color image segmentation.

1.6. Contributions

This dissertation makes several contributions towards fuzzy logic and hybrid algorithms combined with Particle Swarm Optimization, and Fuzzy Decision Tree in solving data classification or clustering analysis problems. The contributions are:

1. A Particle Swarm Optimization based discrete classification implementation with a local search strategy (DPSO-LS) was devised and applied to discrete data. In addition, a fuzzy

DPSO-LS (FDPSO-LS) classifier is proposed for both discrete and continuous data in order to manage imprecision and uncertainty. Experimental results reveal that DPSO-LS and FDPSO-LS outperform other classification methods in most cases based on rule size, True Positive Rate (TPR), False Positive Rate (FPR), and precision, showing slightly improved results for FDPSO-LS.

2. A Fuzzy Decision Tree (FDT) classifier that is based on soft discretization was proposed and applied on feature selection. However, the data contains many redundant or irrelevant features. These features provide no useful information in any context. In order to improve the model interpretability and enhance the generalization, a Genetic Algorithm (GA) based feature selector is applied. The performance evaluation conducted has shown that our FDT classifier obtains in some cases higher values than other decision tree and fuzzy decision tree approaches based on measures such as true positive rate, false positive rate, precision and area under the curve.

3. A clustering approach based on Particle Swarm Optimization is proposed. This approach automatically determines the optimal number of clusters using a threshold vector that is added to the particle. The algorithm starts by partitioning the data set randomly within a preset maximum number of clusters in order to overcome the fuzzy c-means shortcoming of the predefined cluster count. A reconstruction criterion is applied to evaluate the performance of the clustering results of the proposed algorithm. The experiments conducted show that the proposed algorithm can automatically find the optimal number of clusters.

4. A Fuzzy C-Regression Model (FCRM) has been proposed whose prototype is hyper-planed and can either be linear or nonlinear allowing for better cluster partitioning. Thus, this chapter implements FCRM and applies the algorithm to color segmentation using Berkeley's segmentation database. The results show that FCRM obtains more accurate results compared to other fuzzy clustering algorithms.

1.7. Dissertation Overview

This dissertation is a paper-based version, where each chapter has been derived from papers published during the Ph.D. work. This is an overview of the remaining chapters of this dissertation:

In Chapter 2, a fuzzy discrete particle swarm optimization classifier for rule classification is discussed. This chapter is derived from the publications:

- Min Chen and Simone A. Ludwig, "A Fuzzy Discrete Particle Swarm Optimization Classifier for Rule Classification." *International Journal of Hybrid Intelligent Systems*: Special Issue on NaBIC 2012.

- Min Chen and Simone A. Ludwig, "Discrete Particle Swarm Optimization With Local Search Strategy for Rule Classification." *Proceedings of the Fourth World Congress on Nature and Biologically Inspired Computing (IEEE NaBIC'12)*, November 2012, Mexico City, Mexico.

In Chapter 3, a fuzzy decision tree using soft discretization is proposed and a Gentic Algorithm based feature selection method is discussed. This chapter is derived from the publication:

- Min Chen and Simone A. Ludwig, "Fuzzy Decision Tree using Soft Discretization and a Genetic Algorithm based Feature Selection Method." *Proceedings of the Fifth World Congress on Nature and Biologically Inspired Computing (IEEE NaBIC'13)*, August 2013, Fargo, ND, USA.

In Chapter 4, a novel fuzzy clustering using automatic particle swarm optimization is proposed and discussed. This chapter is derived from the publication:

- Min Chen and Simone Ludwig, "Particle Swarm Optimization based Fuzzy Clustering Approach to Identify Optimal Number of Clusters." *Journal of Artificial Intelligence and Soft Computing Research* , 2014.

In Chapter 5, fuzzy c-regression model clustering whose prototype is hyper-planed is applied in color image segmentation using Berkeley's segmentation database.

- Min Chen and Simone A. Ludwig, “Color Image Segmentation Using Fuzzy C-Regression Model.” Submitted to International Journal of Fuzzy Systems.

2. A FUZZY DISCRETE PARTICLE SWARM OPTIMIZATION CLASSIFIER FOR RULE CLASSIFICATION

The need to deduce interesting and valuable information from large, complex, information-rich data sets is common to many research fields. Rule discovery or rule mining uses a set of IF-THEN rules to classify a class or category in a comprehensible way. Besides the classical approaches, many rule mining approaches use biologically-inspired algorithms such as evolutionary algorithms and swarm intelligence approaches. In this chapter, a Particle Swarm Optimization based discrete classification implementation with a local search strategy (DPSO-LS) was devised and applied to discrete data sets. In addition, a fuzzy DPSO-LS (FDPSO-LS) classifier is proposed for both discrete and continuous data sets in order to manage imprecision and uncertainty. A Pittsburgh approach based particle swarm optimization is adopted and applied in classification rule mining.

The remainder of the chapter is arranged as follows. Section 2.1 describes related work. The proposed two approaches DPSO-LS and FDPSO-LS are introduced and described in Section 2.2. The experimental setup and results of the two approaches are demonstrated in Section 2.3. Finally, conclusions and future work are discussed in Section 2.6.

2.1. Related Work

Related work in classification rule mining using biology-inspired algorithms mainly include evolutionary algorithms and swarm intelligence approaches. Genetic algorithm based concept learning uses either the Pittsburgh approach or the Michigan approach [26]. For the Pittsburgh approach, every individual in the GA is a set of rules that represents a complete solution to the learning problem. For the Michigan approach, each individual represents a single rule that provides only a partial solution to the overall learning task.

Genetic algorithm based concept learning has been widely used for rule mining. In [26], a genetic algorithm based algorithm is proposed to discover comprehensive IF-THEN rules. It uses a flexible chromosome encoding where each chromosome corresponds to a classification rule. In addition, a hybrid decision tree/genetic algorithm is used to discover small disjunct rules in [27]. A decision-tree algorithm is used to classify examples belonging to large disjuncts, while a new genetic algorithm is designed for classifying examples belonging to small disjuncts.

Evolutionary approaches for automated discovery of censored production rules, augmented production rules and comprehensible decision rules are introduced in [28, 29, 30], respectively. The proposed GA-based approaches, similarly, use a flexible chromosome encoding, where each chromosome corresponds to an augmented production rule, a comprehensible decision rule or a censored production rule. An Evolutionary Multiobjective Optimization (EMO) algorithm is used to search for a large number of non-dominated fuzzy rule-based classifiers in [31].

With regards to swarm intelligence approaches, a classification algorithm called Ant-Miner, first introduced in [32], has been successfully applied to rule classification problems. PSO is another approach inspired by nature. However, most of the swarm intelligence algorithms for rule classification are based on the Michigan approach ([33, 34]).

Related work in fuzzy classification rule mining using the biology-inspired algorithms mainly include evolutionary algorithms and swarm intelligence approaches. GA is a popular evolutionary algorithm, which has been employed for the learning of fuzzy rules. GAs have been applied to learn both antecedent and consequent of fixed or varying number of fuzzy rules [35, 36, 37]. Also, GAs have been combined with other techniques like neural networks [38], Kalman filters [39], hill climbing [40], and fuzzy clustering [38]. EMO algorithms, which generate a family of equally valid solutions, have been introduced in [41].

Ant Colony Optimization (ACO), one of the swarm intelligence techniques, has been successfully used to extract rule based classification systems. In [42], ACO is used to extract fuzzy IF-THEN rules for the diagnosis of diabetes. A combination of ACO and fuzzy set theory, named FACO-Miner, is applied to learn a set of fuzzy rules from labeled data in a parallel manner in [43]. An improved ACO technique using fuzzy inference rules is applied to image classification and analysis in [44].

With respect to PSO, a Pittsburgh-based PSO fuzzy system for knowledge acquisition is introduced in [45]. A modified PSO, called Mutation PSO (MPSO), is built and used to obtain an optimal fuzzy rule-base. The algorithm generated a compact fuzzy rule base that works efficiently for medical diagnosis problems [46]. In [47], a case study of intrusion detection using a PSO approach for evolutionary fuzzy rule learning is proposed. It is capable of detecting known intrusive behavior in a computer network with an acceptable performance.

PSO has been proven to be able to achieve a faster convergence than the GA algorithm [45]. It has been experimentally shown that the PSO algorithm scales well and is not highly sensitive to the population size [45]. As far as the authors' knowledge is concerned, due to the lack of flexibility of the Pittsburgh approach [48], the Pittsburgh-based PSO algorithm on rule classification is rarely used in literature. On the other hand, in order to avoid premature convergence of particles, the Michigan approach usually requires some changes in the definition of the PSO algorithm to repel a particle from its neighbor [48]. In addition, the Michigan approach aims to optimize each rule's quality individually, and does not take the interaction between other rules into account [33]. In [45], the knowledge acquisition with a Pittsburgh-based swarm-intelligence approach is introduced. A learning strategy of a fuzzy-rule-based meta-scheduler is analyzed and compared with other scheduling strategies. In our study, similarly, we propose a Pittsburgh-based swarm-intelligence method, however, we improve the classification by applying a local strategy to address PSO's convergences problem. Furthermore, in order for the method to handle imprecision and vagueness in data sets fuzzy logic is employed.

2.2. Proposed Approaches

Two classifiers are proposed and investigated: a DPSO-LS classifier and a fuzzy DPSO-LS classifier (abbreviated as FDPSO-LS). The DPSO-LS classifier is designed to classify discrete data sets. As we have mentioned above, the use of linguistic variables and fuzzy IF-THEN rules exploits the tolerance for imprecision and uncertainty. In this respect, we extend the DPSO-LS classifier to a fuzzy DPSO-LS (FDPSO-LS) classifier, which can classify both discrete and continuous data sets. In this section, we first describe the DPSO-LS algorithm followed by a detailed description of the FDPSO-LS classifier.

2.2.1. Discrete Particle Swarm Optimization with Local Strategy

PSO was introduced by Eberhart and Kennedy [6] and is based on the analogy of the behavior of flocks of birds or schools of fish. Although the PSO algorithm was proposed for continuous space problems, however, many real-world data sets use categorical data, and therefore, we considered this within our classification task formulation. In classical PSO, swarm individuals are called particles, and the population is called the swarm. Each particle has its own position, velocity and historical information. The particles fly through the search space by using their own as well as their neighbors' historical information to steer toward the local or global optima.

In particular, a discrete PSO approach (DPSO-LS) for the classification rule mining problem is proposed. A Rule Base (RB) as a whole represents a 'particle'. Each RB is denoted as a matrix, where each row describes a classification rule. The rules are IF-THEN rules consisting of conjunctive antecedents and one consequent. Hence, the i^{th} particle is presented as follows:

$$P_i = \begin{pmatrix} a^i_{1,1} & a^i_{1,2} & \dots & a^i_{1,n} & c^i_1 \\ a^i_{2,1} & a^i_{2,2} & \dots & a^i_{2,n} & c^i_2 \\ \dots & \dots & \dots & \dots & \dots \\ a^i_{m,1} & a^i_{m,2} & \dots & a^i_{m,n} & c^i_m \end{pmatrix} \tag{2.1}$$

where a^i_{mn} represents the n^{th} antecedent in the m^{th} rule of the i^{th} particle. c^i_m is the m^{th} consequent of the i^{th} particle. m is the number of rules, and n is the number of antecedents. Thus, a particle consists of m rules, where each rule has n antecedents and 1 consequent.

The values of every antecedent are enumerated consecutively starting from 1. In this work, an antecedent has 3 discrete values, it will be enumerated as $\{1, 2, 3\}$. In this way, 0 means the antecedent is ignored. Thus, a rule with all its antecedents having a value of 0 is not allowed. In addition, the constraints of the swarm position updating process need to be considered since the particle might fly outside the solution space:

$$a^i_{j,k} \in [0, NF_{in}], j \in \{1, 2, ..., m\} \tag{2.2}$$

$$k \in \{1, 2, ..., n\} \tag{2.3}$$

$$c^i_j \in [1, NF_{out}] \tag{2.4}$$

where NF_{in} and NF_{out} represent the number of discrete values for an antecedent and a consequent, respectively. The i^{th} particle's velocity matrix is denoted as follows:

$$V_i = \begin{pmatrix} v^i_{1,1} & v^i_{1,2} & \cdots & v^i_{1,n} & v^i_{1,n+1} \\ v^i_{2,1} & v^i_{2,2} & \cdots & v^i_{2,n} & v^i_{2,n+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ v^i_{m,1} & v^i_{m,2} & \cdots & v^i_{m,n} & v^i_{m,n+1} \end{pmatrix} \tag{2.5}$$

where $v^i_{j,k} \in [V_{min}, V_{max}]$, $j \in \{1, 2, ..., m\}$, and the velocity matrix has the same dimension as the position matrix. V_{min} and V_{max} are the minimum and maximum values allowed for the velocity, respectively. More specifically, we use a change vector $\vec{V_i}$, which is the change vector for the i^{th} particle with the same dimension as the velocity matrix.

$$\vec{V_i} = \begin{pmatrix} \hat{v}^i_{1,1} & \hat{v}^i_{1,2} & \cdots & \hat{v}^i_{1,n} & \hat{v}^i_{1,n+1} \\ \hat{v}^i_{2,1} & v^i_{2,2} & \cdots & v^i_{2,n} & v^i_{2,n+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \hat{v}^i_{m,1} & \hat{v}^i_{m,2} & \cdots & \hat{v}^i_{m,n} & \hat{v}^i_{m,n+1} \end{pmatrix} \tag{2.6}$$

The values of $\vec{V_i}$ are randomly assigned to 1, 2 and 3, where 1, 2 and 3 are denoted as three directions. 1 is denoted as the direction of the particle's movement from the current position to the local best position ($Pbest$). 2 is denoted as the direction of the particle's movement from the current position to the global best position ($Gbest$). 3 is denoted as the direction of the particle's movement from the current position to another position at random within a specified range. The three directions are randomly assigned by following the ratios ω_1, ω_2, and ω_3 ($\omega_1 < \omega_2 < \omega_3$). As shown in Equation 2.7, the sum of the ratios should be equal to one. By adopting the concept of change vector, the velocity of the particle can be updated by considering the local best position, global best position and random changes. Precisely, as shown in Equation 2.8, for the i^{th} particle, $V_1(t)$ is the difference between the local best position and the current position, while $\vec{V_i}$ consist of 1s, and the rest of the values in the matrix are set to 0. Similarly, $V_2(t)$ is the difference between the global best position and the current position, while $\vec{V_i}$ consist of 2s. Values of $V_3(t)$ are randomly assigned within a specified range (see Equation 2.9), while values of $\vec{V_i}$ consist of 3s at the same positions. $\oplus$ denotes a matrix addition.

$$\omega_1 + \omega_2 + \omega_3 = 1 \tag{2.7}$$

$$V(t+1) = V_1(t) \oplus V_2(t) \oplus V_3(t) \tag{2.8}$$

$$V_3(t) \in [V_{min}, V_{max}] \tag{2.9}$$

After the velocity has been calculated, the particle's position can be computed as follows:

$$P(t+1) = P(t) \oplus V(t+1) \tag{2.10}$$

2.2.2. Definition of Overall Fitness

We propose a rule selection method where the number of classification rules included in each rule set is fixed to a predefined number. That is, each rule set with a specific number of rules (a rule base) is a particle. Thus, the overall fitness function of the rule set can be defined as follows:

$$F(S) = Accuracy(Coverage) = \frac{NCP(S)}{|S|} \tag{2.11}$$

where $NCP(S)$ is the number of instances that have been correctly classified in the data set S, and $|S|$ is the number of instances in the data set S.

2.2.3. Local Mutation Strategy

Since PSO, in general, can easily get stuck in local optima, a local strategy need to be devised that is run after a certain number of iterations has elapsed. In particular, the local strategy that was devised for DPSO-LS makes use of mutation. The proposed local strategy refines the worst rule of the best rule base, i.e., the global best position, in order to improve the overall performance every 20 iterations. Thus, for each selected worst rule, we mutate one value of the antecedent randomly within the constraints to see whether it improves the overall performance or not. If it improves the performance, we stop and replace the worst rule with the new rule. Otherwise, we continue mutating randomly until we have found a new rule or until we have mutated a maximum of 10 times.

The equation to measure the quality of every rule uses the Laplace-corrected precision [33] equation, which is given as:

$$f = \frac{1 + TP}{1 + TP + FP} \tag{2.12}$$

where TP is the number of True Positives, and FP is the number of False Positives. The equation is also used to prune the rules for which the f value is less than 0.1.

2.2.4. DPSO-LS Classifier

The proposed algorithm includes four main phases: data preprocessing phase, training phase, DPSO phase and testing phase. As shown in Figure 1, the DPSO-LS classifier includes all the solid rectangles and excludes the red dashed rectangles (these are only used for FDPSO-LS). The four phases are described respectively as follows.

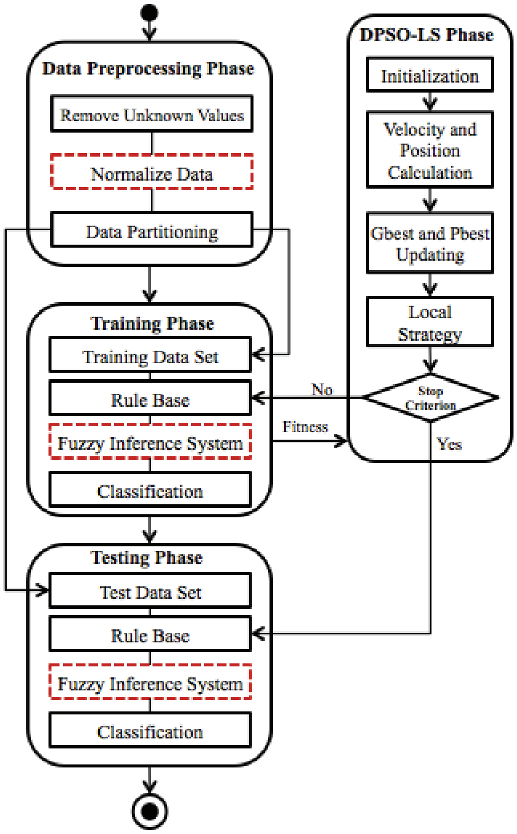

Figure 1. Processes of DPSO-LS based classifiers.

1. Data Preprocessing Phase

In this phase, firstly, we need to remove instances that have unknown values since the proposed system cannot handle these values. It is also known that the proposed system can only handle numerical data, if the class labels are non-numerical data, we convert it into numeric values. Then, the data set is randomly partitioned into 10 folds. 9 folds of the data is training data, which is used in the training phase, and 1 fold of the data is testing data, which is used in the testing phase.

2. DPSO-LS Phase

In this phase, the swarm is initialized. The velocity and position of each particle in the swarm are calculated. $GBest$ and $Pbest$ as described above are calculated, and their values are updated after the velocity and position have been updated accordingly. A local strategy is applied every 20 iterations. If the stopping criterion has not been met, $Gbest$ and $Pbest$ are forwarded to the training phase to calculate the overall fitness (see Equation 2.11), and individual fitness (see Equation 2.12). The DPSO-LS is stopped when the maximum number of iterations is met. The final $Gbest$ is forwarded to the testing phase.

3. Training Phase

   A rule base which is forwarded by the DPSO-LS phase is used to classify the training data set. The overall fitness and individual fitness are calculated accordingly and are forwarded to the DPSO-LS.

4. Testing Phase

   The final $Gbest$ forwarded by the DPSO-LS phase is used to classify the testing data set, and the experimental measures are calculated.

2.2.5. FDPSO-LS Classifier

A modified classifier, called fuzzy DPSO-LS classifier (FDPSO-LS), is implemented for both discrete and continuous data sets. A fuzzy partition with a simple fuzzy grid is adopted. Fuzzy set theory and the concept of linguistic variables, which were proposed by Zadeh [7, 8], have been widely used in pattern recognition and fuzzy reasoning. The use of the simple fuzzy partition method on classification rule discovery has been introduced in [37]. Applications on the fuzzy rule generation for control problems were proposed in [49]. Moreover, several fuzzy approaches for partitioning a pattern space were discussed in [50, 51].

Specially, an example of using the simple fuzzy partition method is showing in Figure 2, each attribute can be partitioned into three linguistic terms (denoted as L = low, M = medium, H = high). Triangular membership functions are used for the linguistic terms. In the proposed method, each linguistic term is viewed as a candidate 1-dimension fuzzy grid. Considering a two-class classification problem as in Figure 2, two antecedents with three membership functions can be partitioned into 9 grids on a 2-dimension plane. The closed circles and open circles denote the pattern in class 1 and class 2, respectively.

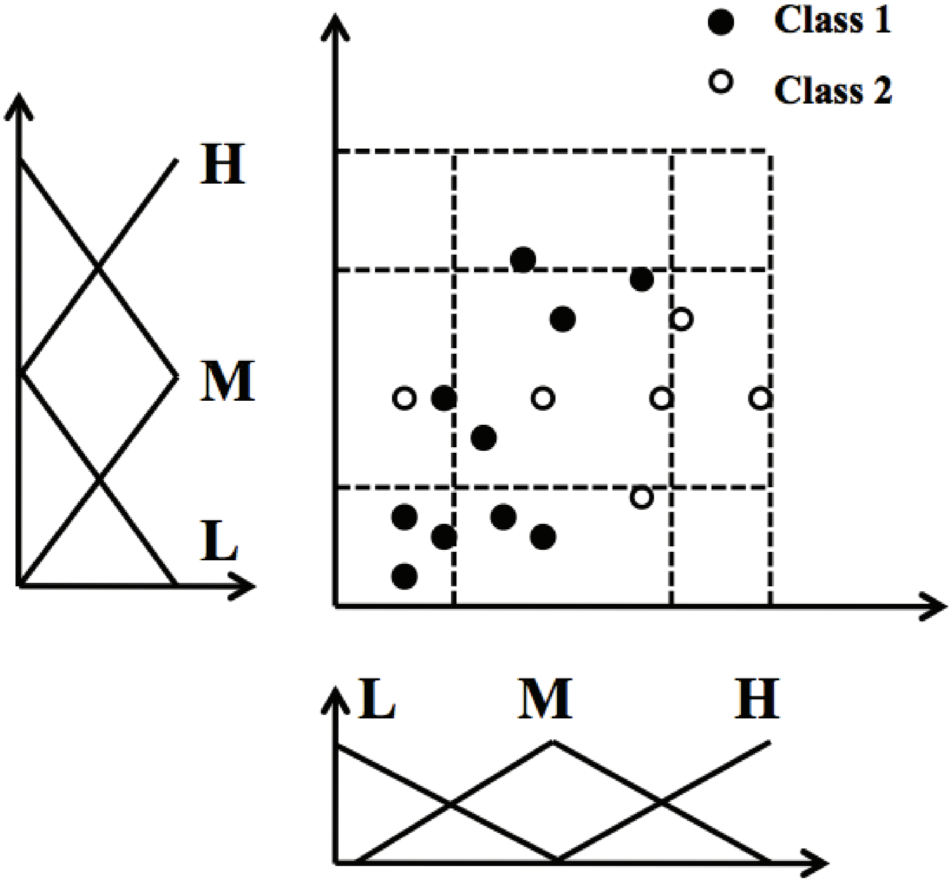

Figure 2. An example of fuzzy partition.

However, in the case of an n-dimensional classification problem, where each dimension has m linguistic terms, the possible number of rules is m^n. As the number of rules rises, an efficient algorithm that can automatically find the fuzzy rules is important and necessary.

Normally, several rules of the rule base are fired in the fuzzy rule classification system. The predicted class for a given instance is determined by the membership degree of the input variables.

Specifically, for each class k,

$$\beta_{\text{Class k}} = \arg\max_{k} \sum_{1 \leq i \leq n} \prod_{1 \leq j \leq m} \mu_{ij} \tag{2.13}$$

where μ_{ij} is the input membership degree of the i^{th} rule of the j^{th} antecedent. The class that has the largest β value is selected as the predicted class. Moreover, unlike the DPSO-LS classifier, the rule that has the smallest β value is chosen as the worst rule.

As shown in Figure 1, FDPSO-LS has similar processes as DPSO-LS, however, all the rectangles are used. The four main phases of data preprocessing, training, DPSO-LS and testing are similar to DPSO-LS. In the data preprocessing phase, besides the removal of unknown values and data partitioning processes, a data normalization process is used to normalize continuous data. Each column of the data set is normalized between 0 and 1 using Equation 2.14:

$$X_i = \frac{X_i - X_{min}}{X_{max} - X_{min}} \tag{2.14}$$

where X_i is the i^{th} value of the column. X_{min} is the minimum value of the column, and X_{max} is the maximum value of the column. The data set is partitioned into 10 folds. 9 folds of the data are used as the training data set, and the remainder is used as the test data set for the implementation.

The DPSO-LS phase is the same as for the DPSO-LS classifier. However, in the training and testing phases, a fuzzy inference process is added for the fuzzy reasoning process. The Fuzzy Inference System (FIS) is a popular computing system based on the concepts of fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning. It has been successfully applied to a wide variety of fields, such as automatic control, data classification, expert systems, decision analysis, etc. Due to its multidisciplinary nature, FIS is known by numerous other names. However, we only concentrate on the concept of the fuzzy IF-THEN rules.

The basic structure of a fuzzy inference process consists of three modules: fuzzification, fuzzy rule base and inference, and defuzzification. As shown in Figure 3, a crisp input is taken, the fuzzification module coverts the crisp input into a fuzzy input using the fuzzy set theory. In the second module, fuzzy rules are contained in a rule base and a reasoning mechanism that performs the inference procedure is included. Finally, a method of defuzzification to extract a crisp output that represents a fuzzy set is needed by the third module. Due to the way outputs are determined, there are two types of inference systems: Mamdani and Sugeno. Mamdani's fuzzy inference system was among the first control systems built using fuzzy set theory, which was proposed in 1975 by Ebrahim Mamdani [52]. Sugeno, or Takagi-Sugeno-Kang, was introduced in 1985 [53]. It is similar to the Mamdani method in many respects, however, the main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant. In this approach, only the Mamdani style of defuzzification is considered.

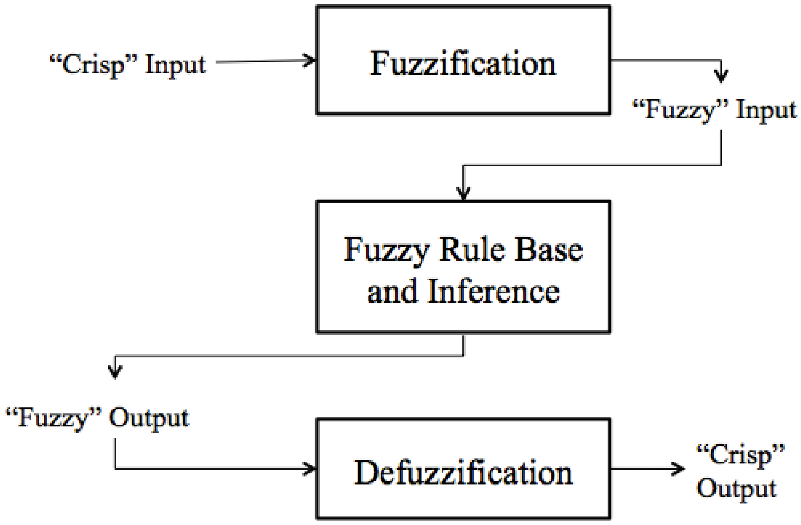

Figure 3. Fuzzy inference process.

2.3. Experiments and Results

As mentioned above, the experiments are conducted for three approaches: DPSO (without local strategy), DPSO-LS and FDPSO-LS. The experimental setup for both approaches are described in the following subsection followed by the description of the experimental results of both approaches. The results of DPSO-LS and FDPSO-LS are listed, respectively, followed by a comparison.

2.3.1. Experimental Setup

The experiments of the two approaches are conducted on a number of data sets taken from the UCI repository [54]. The experiments of the two approaches are evaluated on an ASUS desktop (Intel(R) Dual Core I3 CPU @3.07 GHz, 3.07 GHz) Matlab Version 7.13. All measurements of the two approaches are tested 10 times using 10-fold cross validation [60]. Each data set is divided into 10 random partitions. Nine partitions of the data set are used as the training data, and one partition is selected as the test data.

2.3.2. Results of the DPSO-LS Approach

As far as the performance evaluation is concerned for the proposed DPSO-LS algorithm, a comparison with other rule classification algorithms JRip, PART and decision tree algorithm J48 is performed. These three algorithms have been implemented in WEKA (Waikato Environment for Knowledge Analysis) [60]. The algorithms are summarized as follows:

- **JRip** is a RIPPER rule learning algorithm [55]. JRip is based on association rules with reduced error pruning (REP), and integrates reduced error pruning with a separate-and-conquer strategy. It is a very common and effective technique found in decision tree algorithms.

- **PART** is created by Frank and Witten [56] for a partial decision tree. PART combines the separate-and-conquer strategy of RIPPER with the decision tree. It works by building a rule and removing its cover until all the instances are covered.

- **J48** is a decision tree implementation induced by the C4.5 algorithm, which is developed by Quinlan [13]. It learns decision trees for the given data by constructing them in a top-down way.

Table 1 shows the parameters and their values used for our DPSO, DPSO-LS and FDPSO-LS algorithms. Usually, a large swarm size requires less iterations to reach convergence in PSO. In

Table 1. Parameters and their values of the DPSO and DPSO-LS algorithms.

Parameters	**Values**
Swarm Size	25
Maximum Iteration	100
$(\omega_1, \omega_2, \omega_3)$	(0.2, 0.3, 0.5)
$[V_{min}, V_{max}]$	[-1, 1]

our proposed algorithm, the swarm size is chosen as 25, and the maximum number of iterations for each run is set to 100. The description of the selected data sets used are summarized in terms of number of attributes, number of instances and number of classes as shown in Table 2. The 6 data sets are listed alphabetically, where data set *Breast-L* and *Breast-W* are abbreviations for *Ljubljana Breast Cancer* and *Wisconsin Breast Cancer*, respectively.

Table 2. Data sets used for the experiments.

Data Set	**Attributes**	**Instances**	**Classes**
Balance-scale	4	625	3
Breast-L	9	286	2
Breast-W	9	699	2
Car	6	1728	4
Lymphography	18	146	4
Tic-Tac-Toe	9	958	2

Measured are the rule size, the weighted average True Positive Rates (TPRs) and False Positive Rates (FPRs), as well as the precision.

As we mentioned before, the DPSO can easily get stuck in local optima. In order to see the performance improvements of the local strategy, we compare DPSO (without local strategy) with DPSO-LS (with local strategy) by running them 10 times for 100 iterations each. The average accuracy of the 10 runs is listed in Table 3. A corresponding two-tailed Student's t-test with a significance level of 5% is applied. The results show that the proposed DPSO-LS can achieve better accuracy in all cases. However, DPSO-LS only shows significant improvements in 3 of 6 cases.

Table 3. Average accuracy of DPSO and DPSO-LS for 100 iterations.

Data Set	DPSO (%)	DPSO-LS (%)	Significance
Balance-scale	77.27 ± 3.72	**83.39** ± 3.20	**Yes**
Breast-L	82.57 ± 2.63	**86.71** ± 1.07	**Yes**
Breast-W	91.43 ± 4.25	**94.20** ± 4.30	No
Car	94.92 ± 5.06	**97.30** ± 4.40	No
Lymphography	76.23 ± 3.51	**80.10** ± 3.60	**Yes**
Tic-Tac-Toe	100.00 ± 0.00	100.00 ± 0.00	No

In Figure 4, we see the accuracy of DPSO-LS compared to DPSO, JRip, PART and J48. Error bars are shown on the histograms of the DPSO-LS and DPSO (for the other algorithms, no variants were reported since they are not captured by WEKA). In most cases, the DPSO-LS algorithm has a higher accuracy. Although the Breast-W data set does not show better results, the values of the other four algorithms are very close.

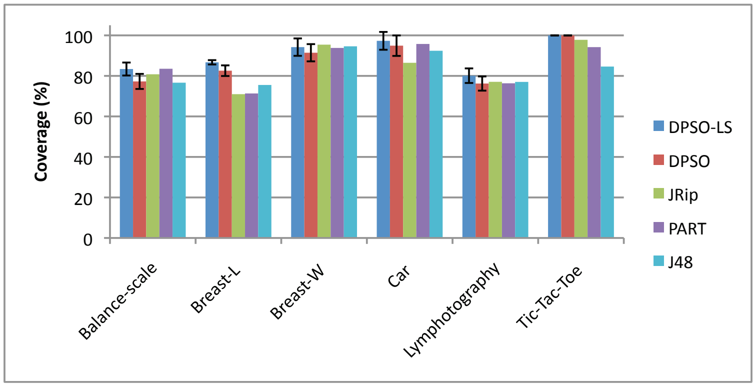

Figure 4. Accuracy of all algorithms.

For all rule mining algorithms it is necessary to test the average rule set size to indicate the complexity of the rule set produced by each algorithm. Table 4 lists the size of the rule set required for DPSO, DPSO-LS, JRip, PART, and J48. As shown in the table, the JRip algorithm always requires the least number of rules, while the PART algorithm requires the most number of rules. J48 uses by far the most number of rules with the exception of the Breast-L data set. The number of rules for the proposed DPSO-LS algorithm is less than the PART algorithm. Both DPSO-LS and DPSO show comparable results in terms of rule size.

Table 4. Average rule size of all algorithms.

Data Set	JRip	PART	J48	DPSO	DPSO-LS
Balance-scale	**12**	47	52	26.01±3.10	24.70±2.66
Breast-L	**3**	20	4	15.25±4.50	17.00±3.50
Breast-W	**6**	10	14	**6.05**±3.01	7.13±2.08
Car	49	68	131	**43.20**±5.20	44.18±4.17
Lymphography	**6**	13	21	11.15±2.50	9.40±3.06
Tic-Tac-Toe	**9**	49	95	35.3±3.76	38.80±1.70

Table 5 lists the average weighted TPR, which is also referred to as sensitivity. As shown in the table, the proposed algorithm, DPSO-LS, scores better than DPSO, JRip, PART and J48 in terms of sensitivity.

Table 5. Average weighted TPRs (%) of all algorithms.

Data Set	JRip	PART	J48	DPSO	DPSO-LS
Balance-scale	80.8	**87.5**	76.6	80.20±3.12	87.40±2.30
Breast-L	71.0	71.3	75.5	81.80±2.22	**89.50**±3.70
Breast-W	95.4	93.8	94.6	92.36±4.30	**97.27**±2.10
Car	86.5	95.8	92.4	93.50±3.10	**98.84**±1.33
Lymphography	77.7	76.4	77.0	73.30±3.26	**80.50**±4.40
Tic-Tac-Toe	97.8	94.3	84.6	**100.00**±0.00	**100.00**±0.00

The weighted average FRPs, which represent *1-Specificity*, are listed in Table 6. The FPRs of DPSO-LS are less than the other algorithms except for the Lymphography data set.

The weighted average precision values are compared in Table 7. The precision of the DPSO-LS is always better than DPSO, JRip, PART and J48, showing the largest improvement on the Breast-L data set.

Table 6. Average weighted FPRs (%) of all algorithms.

Data Set	JRip	PART	J48	DPSO	DPSO-LS
Balance-scale	16.4	9.7	17.3	15.01±3.25	**8.70**±2.20
Breast-L	48.9	54.2	52.4	25.80±4.30	**16.00**±7.20
Breast-W	4.4	8.0	6.4	1.20±0.18	**0.50**±0.01
Car	6.4	1.6	5.6	5.27±2.30	**1.04**±0.05
Lymphography	21.6	21	**18.7**	30.11±5.60	22.00±3.40
Tic-Tac-Toe	3.10	7.6	19.1	**0.00**±0.00	**0.00**±0.00

Table 7. Average weighted precision (%) of all algorithms.

Data Set	JRip	PART	J48	DPSO	DPSO-LS
Balance-scale	74.5	83.3	73.2	79.95±3.80	**85.40**±3.20
Breast-L	68.8	68.2	75.2	83.16±3.30	**89.50**±3.60
Breast-W	95.5	93.8	94.6	92.35±4.10	**96.59**±2.15
Car	88.1	95.9	92.4	93.76±3.30	**99.10**±1.20
Lymphography	76.5	76.6	77.6	71.31±5.12	**78.57**±5.80
Tic-Tac-Toe	97.8	94.2	84.6	**100.00**±0.00	**100.00**±0.00

2.4. Comparison and Results of DPSO-LS and FDPSO-LS for Discrete Data Sets

In order to compare the performance of DPSP-LS and FDPSO-LS on discrete data sets, a corresponding two-tailed Student's t-test with a significance level of 5% is applied. As shown in Table 8, only 2 of the 5 data sets show significant improvements.

Table 8. Average accuracy of DPSO-LS and FDPSO-LS for 100 iterations.

Data Set	DPSO-LS (%)	FDPSO-LS (%)	Significance
Balance-scale	77.27 ± 3.72	77.13 ± 2.50	No
Breast-L	82.57 ± 2.63	86.71 ± 1.07	Yes
Breast-W	91.43 ± 4.25	93.20 ± 2.30	No
Car	94.92 ± 5.06	97.30 ± 4.40	No
Lymphography	76.23 ± 3.51	80.10 ± 3.60	Yes

In Figure 5, we see the average accuracy of the DPSO-LS compared to FDPSO-LS. Error bars are shown on the histograms of both the proposed algorithms. In most cases, the proposed

FDPSO-LS algorithm has a higher accuracy. Besides the Balance-scale data set, FDPSO-LS achieves better results for the other four data sets.

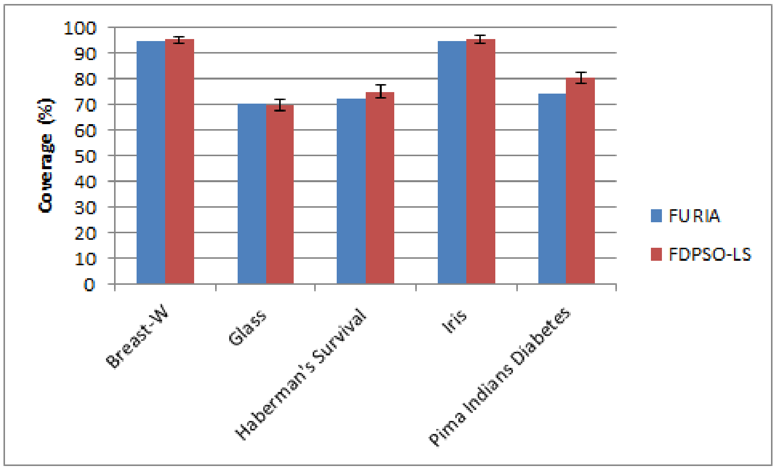

Figure 5. Accuracy comparison of DPSO-LS and FDPSO-LS.

In Table 9, the average rule size of DPSO-LS and FDPSO-LS is compared. FDPSO-LS requires less number of rules than DPSO-LS due to the usage of the linguistic variables.

Table 9. Average rule size of DPSO-LS and FDPSO-LS for 10 runs.

Data Set	DPSO-LS (%)	FDPSO-LS (%)
Balance-scale	24.70 ± 2.66	**7.12** ± 2.10
Breast-L	17.00 ± 3.50	**10.24** ± 3.07
Breast-W	7.13 ± 2.08	**7.08** ± 2.30
Car	44.18 ± 4.17	**14.12** ± 4.40
Lymphography	9.40 ± 3.06	**5.60** ± 3.60

Table 10 shows the average weighted TPR of DPSO and FDPSO-LS. FDPSO-LS does not show improvements compared to DPSO-LS for discrete data sets. FDPSO-LS scores slightly better on 2 out of 5 data sets.

As shown in Table 11, FDPSO-LS has a smaller FPRs in most cases except for the Car data set.

In terms of average weighted precision, FDPSO-LS does not show improvements compared to DPSO-LS on the discrete data sets except for Lymphography as shown in Table 12.

Table 10. Average weighted TPRs of DPSO-LS and FDPSO-LS for 10 runs.

Data Set	DPSO-LS (%)	FDPSO-LS (%)
Balance-scale	87.40 ± 2.30	**88.13** ± 3.12
Breast-L	**89.50** ± 3.70	86.71 ± 2.70
Breast-W	**97.27** ± 2.10	95.7 ± 2.30
Car	**98.84** ± 1.33	93.80 ± 3.45
Lymphography	80.50 ± 4.40	**83.80** ± 2.60

Table 11. Average weighted FPRs of DPSO-LS and FDPSO-LS for 10 runs.

Data Set	DPSO-LS (%)	FDPSO-LS (%)
Balance-scale	8.70 ± 2.20	**1.65** ± 2.80
Breast-L	16.00 ± 7.20	**7.42** ± 1.07
Breast-W	5.00 ± 1.20	**4.10** ± 2.23
Car	**1.04** ± 0.05	5.80 ± 2.18
Lymphography	22.00 ± 3.4	**16.50** ± 2.34

Overall, with respect to discrete data sets, FDPSO-LS does not significant good improvements in most cases. One reason is that it does not efficiently normalize discrete data sets using liguistic terms. Usually, it causes overfitting and decreases the accuracy. For example, for the Balance-scale data set each attribute has either 3 or 4 discrete values, and FDPSO-LS uses 3 membership functions. When we normalize the attribute values into 3 membership function, the data does not partition well for the attributes having small discrete values.

2.5. Results of FDPSO-LS Approach for Continuous Data Set

As far as the performance evaluation for the proposed FDPSO-LS is concerned, a comparison with other rule classification algorithm FURIA is performed. FURIA is short for Fuzzy Unordered Rule Induction Algorithm which extends the well-known RIPPER algorithm [57]. FURIA learns unordered fuzzy rule sets instead of rule lists. It includes a number of modifications and extensions to deal with uncovered examples.

Table 12. Average weighted precision of DPSO-LS and FDPSO-LS for 10 runs.

Data Set	DPSO-LS (%)	FDPSO-LS (%)
Balance-scale	**85.40** ± 3.20	82.10 ± 2.50
Breast-L	**89.50** ± 3.60	85.71 ± 3.32
Breast-W	96.59 ± 2.15	**96.70** ± 4.30
Car	**99.10** ± 1.20	97.30 ± 2.40
Lymphography	78.57 ± 5.80	**82.80** ± 3.41

The description of the selected data sets used are summarized in terms of number of attributes, number of instances, and number of classes as shown in Table 13. The 5 data sets are listed alphabetically.

Table 13. Datasets used for the proposed fuzzy rule-based system using DPSO-LS.

Data Set	**Attributes**	**Instances**	**Classes**
Breast-W	9	699	2
Glass	10	214	7
Haberman’s Survival	3	306	2
Iris	4	150	3
Pima Indians Diabetes	8	768	2

Measured are also the rule size evolved, the weighted average TPRs and FPRs, as well as the precision.

In order to observe the performance, we compared FURIA with FDPSO-LS by running both algorithms 10 times for 100 iterations each. The average accuracy of the 10 runs is listed in Table 14. The corresponding two-tailed Student’s t-test with a significance level of 5% was applied. The results show that the proposed FDPSO-LS can achieve better accuracy in most cases except for the glass data set. However, FDPSO-LS only shows significant improvements for 2 of the 5 data sets.

In Figure 6, we see the average accuracy of the proposed FDPSO-LS compared to FURIA. Error bars are shown on the histograms of the proposed FDPSO-LS. For most data sets, the proposed FDPSO-LS algorithm has a higher accuracy. Besides, for the glass data set FDPSO-LS obtains higher accuracy for the other data sets.

Table 14. Average accuracy of FURIA and FDPSO-LS for 100 iterations.

Data Set	FURIA (%)	FDPSO-LS (%)	Significance
Breast-W	94.71	**95.20**±1.30	No
Glass	70.56	69.70±2.20	No
Haberman's Survival	72.55	**75.02**±2.40	**Yes**
Iris	94.67	**95.56**±1.70	No
Pima Indians Diabetes	74.48	**80.60**±2.30	**Yes**

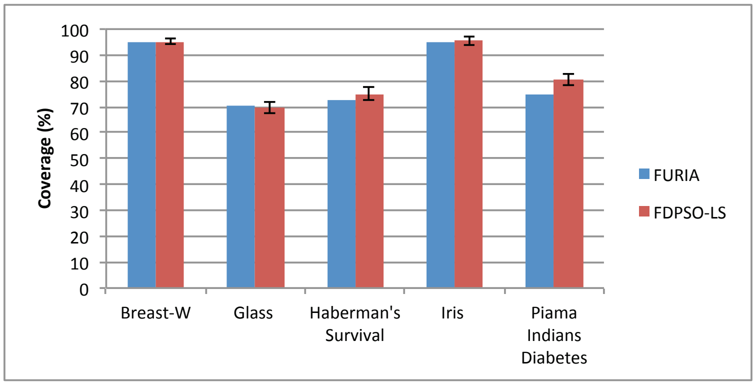

Figure 6. Accuracy comparison of the proposed FDPSO-LS and FURIA.

Table 15 lists the size of the rule set required for FDPSO-LS and FURIA. As shown in the table, the number of rules for the proposed FDPSO-LS is less than for the FURIA algorithm for most data sets. The reason is that the proposed FDPSO-LS reduces the rule size since it uses the local strategy. The values after ± are standard deviations of the corresponding results.

Table 16 lists the average weighted True Positive Rates (TPRs), which are also referred to as sensitivity. As shown in the table, the proposed algorithm, FDPSO-LS, scores better than FURIA for most data sets in terms of sensitivity except for the Glass data set.

The weighted average FPRs, which represent *1-Specificity*, are listed in Table 17. The FPRs of the proposed FDPSO-LS are less than FURIA, which indicates that FURIA has a higher false positive rate.

Table 15. Average rule size of FDPSO-LS and FURIA.

Data Set	FURIA (%)	FDPSO-LS (%)
Breast-W	15	**7.12**±2.10
Glass	16	**9.40**±3.20
Haberman's Survival	**4**	7.20±2.40
Iris	5	**4.00**±1.70
Pima Indians Diabetes	**5**	7.70±2.30

Table 16. Average weighted TPRs of FDPSO-LS and FURIA.

Data Set	FURIA (%)	FDPSO-LS (%)
Breast-W	94.7	**95.20**±2.13
Glass	**70.6**	69.50±3.21
Haberman's Survival	72.5	**78.10**±2.72
Iris	94.7	**94.74**±2.33
Pima Indians Diabetes	74.5	**81.20**±3.11

The weighted average precision values are compared in Table 18. The precision of FDPSO-LS is always better than FURIA, showing the largest improvement on the Haberman's Survival and Pima Indians Diabetes data sets.

2.6. Summary

In this study, we have proposed two classifiers: DPSO-LS and FDPSO-LS. Both classifiers are based on the proposed DPSO-LS algorithm, which uses a rule base to represent a 'particle' that evolves the rule base over time. DPSO-LS is implemented as a matrix of rules, representing IF-THEN classification rules, that have conjunctive antecedents and one consequent. In addition, a local mutation search strategy was incorporated in order to take care of the premature convergence of PSO. The DPSO-LS classifier was applied on discrete data sets based on the IF-THEN classification rules, while the FDPSO-LS is based on the concept of fuzzy IF-THEN rules and is applied to both discrete and continuous data sets.

Experiments were conducted using 6 discrete data sets and 5 continuous data sets that are taken from the UCI repository. Our DPSO-LS algorithm was compared against DPSO, JRip,

Table 17. Average weighted FPRs of FDPSO-LS and FURIA.

Data Set	FURIA (%)	FDPSO-LS (%)
Breast-W	6.7	**4.70**±3.23
Glass	13.1	**7.80**±3.81
Haberman's Survival	57.3	**33.80**±5.20
Iris	2.7	**2.58**±1.20
Pima Indians Diabetes	36.7	**23.30**±3.70

Table 18. Average weighted precision of FDPSO-LS and FURIA.

Data Set	FURIA (%)	FDPSO-LS (%)
Breast-W	94.7	**96.70**±2.70
Glass	**70.5**	70.10±3.20
Haberman's Survival	69 .0	**77.80**±3.13
Iris	94.7	**95.10**±2.40
Pima Indians Diabetes	73.7	**80.23**±3.61

PART and J48. In addition, FDPSO-LS was compared against FURIA. Measures used were rule size, TPRs, FPRs, and precision. The experimental results revealed that DPSO-LS achieves better performance for most data sets than FPSO-LS applied to discrete data sets. On the other hand, FDPSO-LS obtains better performance when applied to continuous data sets compared to FURIA.

3. FUZZY DECISION TREE USING SOFT DISCRETIZATION AND A GENETIC ALGORITHM BASED FEATURE SELECTION METHOD

In data mining, decision tree learning is an approach that uses a decision tree as a predictive model mapping observations to conclusions. The fuzzy extension of decision tree learning adopts the definition of soft discretization. Many studies have shown that decision tree learning can benefit from the soft discretization method leading to improved predictive accuracy. This chapter implements a Fuzzy Decision Tree (FDT) classifier that is based on soft discretization by identifying the best "cut-point". The selection of important features of a data set is a very important preprocessing task in order to obtain higher accuracy of the classifier as well as to speed up the learning task. Therefore, we are applying a feature selection method that is based on the ideas of mutual information and genetic algorithms. The performance evaluation conducted has shown that our FDT classifier obtains in some cases higher values than other decision tree and fuzzy decision tree approaches based on measures such as true positive rate, false positive rate, precision and area under the curve.

The contribution of this chapter is arranged as follows. Section 3.1 describes related work. The proposed approach is introduced and described in Section 3.2. The experimental setup and results are demonstrated in Section 3.3. Finally, conclusions and future work are discussed in Section 3.4.

3.1. Related Work

Related work regarding the classification task in the area of data mining include neural networks, naive Bayes classification, decision tree, genetic algorithm, etc. [58]. Neural networks have become equally popular to decision trees due to its relative ease of application and abilities to provide gradual improvements [59]. Neural networks are seen as data driven self-adaptive methods, which can adjust themselves to the data without any explicit specification of the underlying model [60]. However, neural networks lack similar levels of comprehensibility as decision trees, which is a problem when users want to understand or justify the decisions [59].

Naive Bayes learning is one particular strategy belonging to the category of learning methods. It is a statistical method for classification, which is based on applying the Bayes' theorem with the naive independence assumption [61]. Naive Bayes learning has been deployed in numerous classification tasks due to its simplicity, effectiveness and incremental training ability. Naive Bayes classifiers have widespread deployment in medical diagnosis [62], email filtering [63], and recommender systems [64, 65]. Due to the independence assumption of Naive Bayes, a large amount of research has been conducted on relaxing the Naive Bayes independence assumption in machine learning. However, learning the tree structure is not trivial especially in the area of text classification [66].

With respect to fuzzy decision trees applied to classification tasks, fuzzy decision trees have been applied in the medical and financial fields [59], and have been used for ranking tasks [11], etc. Fuzzy decision tree induction follows the same steps as that of when building a classical decision tree. [67] proposed a novel criterion on measurement of cognitive uncertainty, and [68] proposed an alternative criterion based on fuzzy mutual entropy in the possibility domain.

Related work related to feature selection has shown that many search approaches have been proposed. [69] aggressively reduce the document vocabulary in a naive Bayes model and a decision tree approach using an information measure. A normalized mutual information feature selection (NMIFS) [70] is proposed as a measure of redundancy among features. Two feature evaluation metrics for the naive Bayes classifier have been applied on multi-class text data sets in [71]. Three new approaches to fuzzy-rough feature selection based on fuzzy similarity relation have been proposed in [72] to provide robust solutions and advanced tools for data analysis.

In general, feature selection can improve the scalability, efficiency and accuracy of classifiers. Therefore, our FDT approach makes use of a feature selection technique.

3.2. Fuzzy Decision Tree Classifier

The main difference between classical DT and FDT is using crisp or soft discretization respectively. The classical DT uses crisp discretization while fuzzy decision tree is based on soft discretization. The decision space is partitioned into a set of non-overlapping subspaces using the crisp discretization method. For soft discretiztion, the decision space is partitioned into a set of overlapping subspaces. For both classical and fuzzy decision trees, each path from the root node to a leaf node represents a classification rule. In a more explicit form, the i^{th} branch has the following form:

$$\text{IF } x_{i1} \in A_1^m \text{ AND .. AND } x_{ij} \in A_j^n \text{ THEN } c_i \in C_i^k \tag{3.1}$$

where x_{ij} denotes the j^{th} attribute of the i^{th} branch. A_j^m denotes the m^{th} antecedent value of the j^{th} attribute. c_i is the consequent of the i^{th} rule.

The fuzzy decision tree has been extended in the possibility domain based on fuzzy set theory [70]. A fuzzy set F is characterized by a membership function $F(a) : U \to [0, 1]$. $F(a)$ is the membership degree of F taking a value $a \in U$. Let $V = \{F_1, F_2, ..., F_m\}$ be a family of fuzzy sets of U. Then

$$\sum_{i=1}^{m} F_i(a) = 1, \forall a \in U \tag{3.2}$$

The cut-point is determined by the fuzzy set pair A_1 and A_2 such that $A_1(a) + A_2(a) = 1$. The fuzzy class entropy in a data set S is:

$$E(S) = \sum_{j=1}^{k} p(c_j, S) \log p(c_j, S) \tag{3.3}$$

where $p(c_j, S) = \sum_{a_i \in c_j} (A_1(a_i) + A_s(a_i))$ is the fuzzy proportion of examples in S. The class information entropy is calculated by the probability of fuzzy partition as follows:

$$E(S) = \frac{N^{S_1}}{N^S} E(S_1) + \frac{N^{S_2}}{N^S} E(S_2) \tag{3.4}$$

$$E(S_i) = -\sum_{j=1}^{k} p(c_j, S_i) \log p(c_j, S_i), i = 1, 2 \tag{3.5}$$

$$p(c_j, S_i) = \frac{N^{S_i c_j}}{N^{S_i}}, i = 1, 2 \tag{3.6}$$

where $N^S = \sum_{n=1}^{|S|} \sum_{i=1}^{2} A_i(a_n)$, $N^{S_i} = \sum_{n=1}^{|S|} A_i(a_n)$, $i = 1, 2$.

A fuzzy discretization process mainly includes four phases (seen in Figure 7): sorting, evaluation, splitting and stopping. Since we are also considering feature selection as a preprocessing step, it is the step to be performed before the other four phases are started.

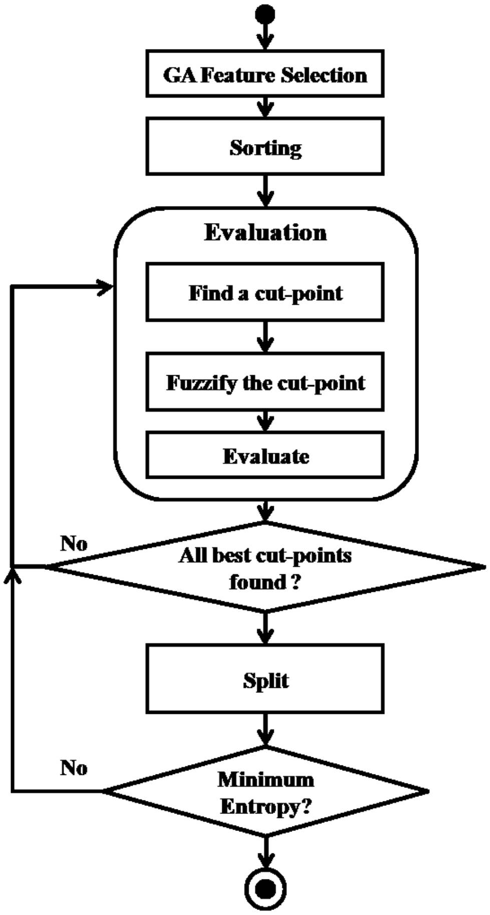

Figure 7. A fuzzy discretization process.

3.2.1. Preprocessing Phase

Feature selection is a common technique in data mining in order to reduce the overall feature set that is provided to the algorithm choosing the most important features to be used for the training of the classifier. However, not only does the reduction of features contribute to a faster learning process, but it usually also improves the classification accuracy. For the feature selection task, methods from information theory are frequently used. Feature selection involves the maximization of the mutual information between features and the class label. However, this procedure is very computationally expensive since the joint entropy has to be calculated requiring the estimation of the joint probability distributions. In order to reduce the computational complexity, a variable selection based on the principle of minimum-redundancy/maximum-relevance, which maximizes the mutual information indirectly was proposed in [73]. However, since all possible combinations of variables need to be checked, there is still a large computational portion involved, thus, a simple method of incremental search, that obtains sub-optimal solutions has been proposed by previous work [74]. The use of a genetic algorithm was proposed to address the combinatorial checking of the variables, which our FTD classifier has adopted.

Algorithm 1 GA-based Feature Selection Method

```
Input: number of features
Input: feature vector
Input: class vector
Output: selected feature vector
$gen_{max}$: maximum number of generations
$N_{pop}$: population size
calculate entropy of each feature
calculate output entropy
calculate mutual information between feature and output
calculate mutual information between features
random initialization of population
for $gen = 1 : gen_{max}$ do
    for $index = 1 : N_{pop}$ do
        calculate maximum relevance
        calculate minimum redundancy
        calculate fitness by subtracting max. relevance from min. redundancy
    end for
    rank population according to their fitness
    perform crossover
    remove repeated features and features with $entropy = 0$
end for
```

Algorithm 1 shows the steps involved in the feature selection process. The inputs are the number of features of the data set, feature vector, and class vector. The output is a vector of selected features. The first steps of the algorithm are the calculation of the entropy of each feature vector and the class vector, as well as the calculation of the mutual information between the feature and class vectors and between the features. Once these values are calculated the GA process can start by setting up a population of randomly initialized chromosomes. The first generation can begin. While iterating over the population, the maximum relevance, minimum redundancy and fitness value are calculated for each chromosome. Afterwards the population is ranked, crossover is performed, and repeated features and features with an entropy of 0 are removed and another generation is started. This process proceeds until the maximum number of generations is reached. The feature vector found is the one used for the next steps in our proposed FDT approach.

1. Sorting Phase

   The continuous values of a feature are sorted in either ascending or descending order. This task can be computationally expensive if care is not taken when considering the sorting algorithm. Quick-sort is one efficient sorting algorithm, which has a time complexity of $O(NlogN)$ [75].

2. Evaluation Phase

   The next step after sorting is to find the best “cut-point”, which can split a range of continuous values into two parts. In the proposed algorithm, the evaluation function used is as given by Equation 3.4.

3. Splitting Phase

   The intervals are split in a top-down strategy, which requires to evaluate “cut-points”. In order to choose the best one and split the range of continuous values into two partitions, the algorithm runs recursively for each part until a stopping criterion is satisfied.

4. Stopping Phase

A stopping criterion specifies when the discretization process is stopped. Specifically, a threshold value $\theta \in [0.1, 0.2]$ is predefined. If the truth level of a branch $\frac{N^{S_1}}{N^S}$ is greater than θ, then the truth level of the branch belonging to the j^{th} class is calculated as follows:

$$\delta_{i,j} = \frac{\sum_{a_k \in c_j} A_i(a_k)}{N^{S_i}}, i = 1, 2 \tag{3.7}$$

Another predefined maximum value of δ called $\mu \in [0.8, 0.9]$ is used as the stopping criterion. If the maximum δ value is greater than μ, the corresponding branch search is terminated.

Generally, a FDT classifier starts by sorting the continuous values of a feature. It then generates a possible candidate “cut-point”, and fuzzifies the “cut-point”. It uses an entropy evaluation function to check whether the candidate’s “cut-point” is satisfied or not. It recursively keeps checking until the best “cut-point” is found, and repeats to generate the soft discretization for the other attributes. When all attributes have been soft discretized, the attribute of minimum value will be selected to generate two child branches and nodes. This process repeats until the stopping criterion is met.

3.3. Experiments and Results

In order to investigate the performance of our FDT approach, experiments are conducted comparing the effect of using all features of five chosen data sets, or using the preprocessing step that reduces the feature set with the GA-based feature selection method as described earlier. The experimental setup is described in the following subsection followed by the experimental results.

3.3.1. Experimental Setup

The experiments of all algorithms are conducted on a number of data sets taken from the UCI repository [54]. The experiments of FDT are run on an ASUS desktop (Intel(R) Dual Core I3

CPU @3.07 GHz, 3.07 GHz) of Java Version 1.6.0.25. A few data mining algorithms are used for comparison provided by the Weka software (version 3.7.8). All experiments use the 10-fold cross validation [60] technique. Each data set is divided into 10 partitions. Nine partitions of the data set are used as training data and one partition is selected as test data.

3.3.2. Experimental Results

In order to compare our FDT classifier, two DT classifiers J48 and REPTree, and a fuzzy rule classification algorithm FURIA were chosen. The algorithms are summarized as follows:

- **J48** is a decision tree implementation induced by the C4.5 algorithm, which is developed by Quinlan [13]. It learns decision trees for the given data by constructing them in a top-down way.

- **REPTree** stands for Reduced Error Pruning Tree [60], which is a fast decision tree implementation that builds a decision tree using information gain as the splitting criterion. It adopts a reduced-error pruning using top-down strategy. It uses the C4.5 method to deal with missing values and only sorts values of numeric attributes once.

- **FURIA** is short for Fuzzy Unordered Rule Induction Algorithm, which extends the well-known RIPPER algorithm [57]. FURIA learns unordered fuzzy rule sets instead of rule lists. It includes a number of modifications and extensions to deal with uncovered examples.

Table 19. GA parameters of GA-based feature selection method.

Parameter	**Values**
Population size	$200 \times \#$ of selected features
Maximum iteration	80
Selection	Elitism
Crossover rate	1

Table 19 shows the parameters and their values used for FDT with the GA-based feature selection. For the proposed algorithm, the population size is chosen as the product of 200 and number of selected features, and the maximum number of iterations is set to 80. An elitist selection strategy is selected and the crossover rate is set to 1.

The description of the selected data sets used are summarized in terms of number of attributes, number of instances and number of classes as shown in Table 20. The 5 data sets are listed alphabetically. The values in brackets under the column *Features* is the reduced number of features after the selection process is applied.

Table 20. Datasets used for experiments.

Data Set	Features	Instances	Classes
Diabetes	8 (4)	768	2
Glass	10 (8)	214	7
Ionosphere	34 (18)	351	2
Pendigits	16 (9)	10992	10
Vehicle	18 (7)	946	4

Measured are the weighted average true positive rate (FPR) and the false positive rate (TPR), as well as the precision. Experiments were run using the data sets as listed above on all algorithms, first without the feature selection stage meaning that all features were used, and the second time using the reduced feature set as determined by the GA-based feature selection method. All results reported in the tables are reported by a number indicating all features were used from the data sets, and the second value in brackets are results when the algorithms are run with the reduced feature set. The values in bold are the best values comparing the results for with/without the GA-based feature selection method.

In Table 21, the average weighted true positive rates (TPR) of all algorithms are measured. As shown in the table, FURIA and FDT using soft discretization always score better than the classical DT techniques, J48 and REPTree that use hard discretization. In addition, FDT with/without the GA-based feature selection method scores slightly better than FURIA on most data sets.

Table 21. Average weighted TPR (%) of all algorithms.

Data Set	J48	REPTree	FURIA	FDT
Diabetes	73.8 (*74.9*)	75.3 (*72.8*)	74.5 (*75.1*)	**76.2** (***76.8***)
Glass	66.8 (*71.5*)	66.4 (*69.2*)	**70.6** (*74.8*)	**70.6** (***75.1***)
Ionosphere	91.5 (*91.7*)	89.5 (*90.6*)	91.2 (*89.5*)	**91.6** (***91.9***)
Pendigits	96.6 (*93.9*)	95.6 (*92.9*)	**98.0** (***94.5***)	96.8 (*93.4*)
Vehicle	72.5 (*68.3*)	72.3 (*66.7*)	70.6 (*70.2*)	**72.6** (***72.1***)

The average weighted false positive rate (FPR) is tabulated in Table 22. FURIA and FDT achieve better results (smaller values) than J48 and REPTree. Furthermore, FDT scores slightly better than FURIA in most of the cases.

Table 22. Average weighted FPR (%) of all algorithms.

Data Set	J48	REPTree	FURIA	FDT
Diabetes	32.7 (*32.3*)	32.8 (*34.8*)	35.7 (*36.2*)	**31.9** (***31.3***)
Glass	13.0 (*10.4*)	13.8 (*12.3*)	13.1 (*11.1*)	**12.4** (***10.2***)
Ionosphere	12.5 (*11.6*)	13.2 (*11.9*)	12.3 (*14.3*)	**11.9** (***11.2***)
Pendigits	0.4 (*0.7*)	0.5 (*0.9*)	**0.2** (***0.6***)	0.4 (*0.8*)
Vehicle	9.3 (*10.7*)	9.3 (*10.2*)	9.8 (*11.1*)	**9.2** (***10.1***)

With respect to the average weighted precision (see Table 23), FURIA and FDT obtain better results than J48 and REPTree. In most cases, FDT scored slightly better than FURIA.

Table 23. Average weighted precision (%) of all algorithms.

Data Set	J48	REPTree	FURIA	FDT
Diabetes	73.5 (*74.4*)	74.7 (*72.3*)	73.7 (*74.4*)	**75.7** (***76.5***)
Glass	67.0 (*71.5*)	65.8 (*69.0*)	**70.5** (***72.1***)	68.5 (*70.3*)
Ionosphere	91.5 (***91.8***)	89.4 (*90.6*)	91.2 (*89.4*)	91.6 (***91.8***)
Pendigits	96.6 (*93.9*)	95.6 (*92.3*)	**98.0** (***94.8***)	96.8 (*94.1*)
Vehicle	72.2 (*67.9*)	71.1 (*68.3*)	68.8 (*63.9*)	**73.8** (***71.2***)

AUC is the area under the ROC curve. ROC stands for "Receiver Operation Characteristic" which is part of a field called "Signal Detection Theory" developed during World War II for the analysis of radar images [76]. Two different methods are used to calculate the AUC. J48, REPTree and FURIA use a parametric method using a maximum likelihood estimation to fit a smooth curve to the data points since these algorithms are part of the WEKA software. Our FDT classifier uses a non-parametric method based on the construction of a trapezoid under the curve as an approximation of the area. In order to compare the performance of using the GA-based feature selection method, the AUC of all the algorithms with/without the feature selection are measured. J48-P, REPTree-P, FURIA-P and FDT-P are abbreviations for the algorithms using the feature selection preprocessing method.

In Figure 8, J48 when the GA-based feature selection method was applied achieved the same or even better AUC values for 4 out of 5 data sets.

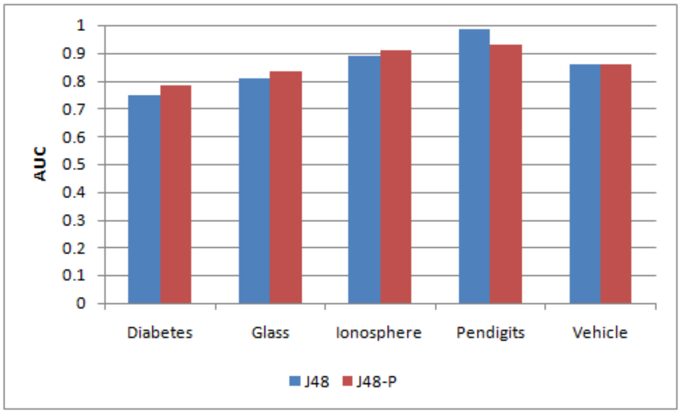

Figure 8. AUC of J48 and J48-P.

The AUC values when using REPTree with the reduced feature set are either the same or even better than using REPTree when all features are used, the score is 3 out of 5.

Evaluating AUC on the FURIA classifier shows that 4 out of 5 data sets have less AUC values when the GA-based feature selection method is used. It seems that FURIA suffers from over-fitting when using the GA-based preprocessing methods. The results are shown in Figure 10.

As shown in Figure 11, the AUC values of 4 out of 5 data sets show slight improvement when using FDT with the GA-based feature selection method.

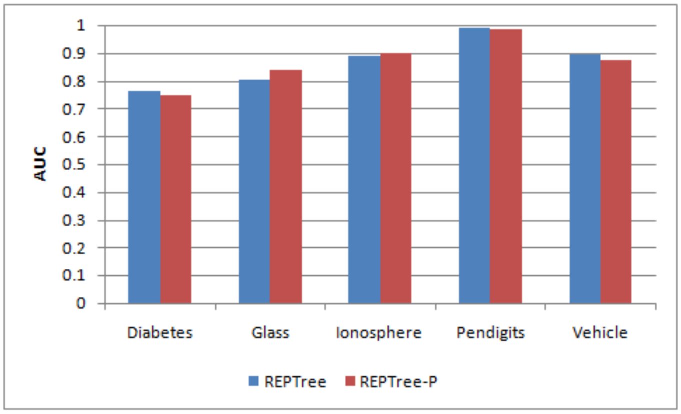

Figure 9. AUC of REPTree and REPTree-P.

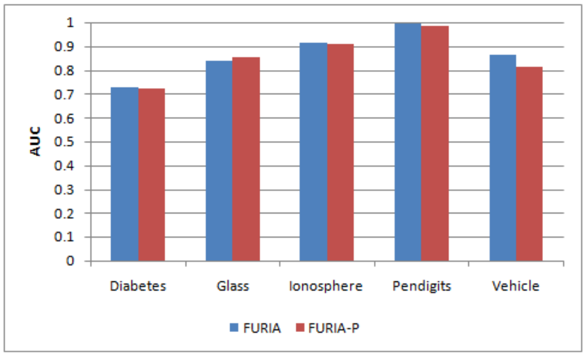

Figure 10. AUC of FURIA and FURIA-P.

Generally, most algorithms, except FURIA, achieve slightly higher AUC values when the GA-based feature selection is used as the data preprocessing method.

3.4. Summary

In this chapter, we presented a fuzzy decision tree (FDT) approach using a GA-based feature selection method. The FDT approach uses soft-discretization searching for the best cut-point in order to improve the predictive accuracy. The soft-discretization works by partitioning the decision space into a set of overlapping subspaces instead of using crisp discretization partitioning. Futhermore, since the reduction of the feature space has shown to improve the accuracy of classifiers in general, we investigated a GA-based feature selection method combined with our FDT approach.

Our FDT classifier was compared to J48, REPTree, and FURIA both with and without using the GA-based feature selection method. Five continuous-valued data sets taken from the UCI

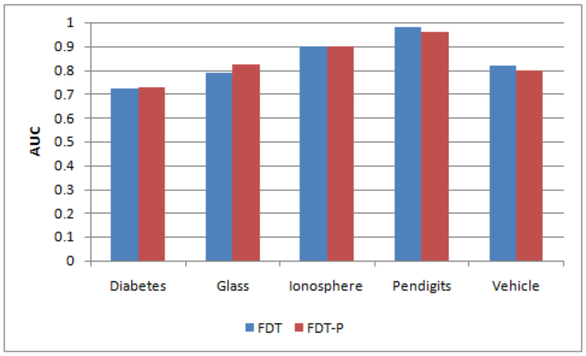

Figure 11. AUC of FDT and FDT-P.

repository were used. Overall, the results revealed that the approaches using soft discretization rather than hard discretization, such as FURIA and our FDT classifier, obtained better predictive classification accuracy in terms of TPR, FPR, precision and AUC. Furthermore, our proposed classifier achieved slightly better results than FURIA in most cases.

4. PARTICLE SWARM OPTIMIZATION BASED FUZZY CLUSTERING APPROACH TO IDENTIFY OPTIMAL NUMBER OF CLUSTERS

Fuzzy clustering is a popular unsupervised learning method used in cluster analysis which allows a data point to belong to two or more clusters. Fuzzy c-means is one of the most well-known and used methods, however, the number of clusters need to be defined in advance. This chapter proposes a clustering approach based on Particle Swarm Optimization. This approach automatically determines the optimal number of clusters using a threshold vector that is added to the particle. The algorithm starts by partitioning the data set randomly within a preset maximum number of clusters in order to overcome the fuzzy c-means shortcoming of the predefined cluster count. A reconstruction criterion is applied to evaluate the performance of the clustering results of the proposed algorithm. The experiments conducted show that the proposed algorithm can automatically find the optimal number of clusters.

The rest of this chapter is organized as follows. In Section 4.1, fuzzy c-means and PSO are introduced. The proposed algorithm is described in Section 4.2. A list of validity indices is given as well. The experimental results and analysis is described in Section 4.4. We finally conclude this chapter in Section 4.6.

4.1. Related Work

FCM was first developed by [77] in 1973, and was extended by [24] in 1981. Since then, FCM is one of the best fuzzy clustering methods. Many different variants of FCM have been introduced. For example, the Gustafson-Kessel (GK) algorithm [21] is a fuzzy clustering technique which can estimate local covariance to partition data into subsets, which can be well fitted with linear sub-models. However, since considering a general structure of the covariance matrix can have a substantial effect on the modeling approach, the Gath-Geva algorithm [79] was proposed to overcome this shortcoming. Another algorithm, called Fuzzy C-Varieties (FCV) [78] clustering algorithm, is a fuzzy clustering method for which the prototype of each cluster is represented as a multi-dimensional linear vector. The approach is similar to cluster analysis, however, it uses the statistical method of principal component analysis for the clustering task. Another algorithm, referred to as generalized FCM algorithm, is presented in [80], in which setting of the algorithm parameters is being done automatically.

Related work lists many evolutionary computation methods that have been applied for clustering. For example, a hybrid technique based on combining the k-means algorithm and Nelder-Mead simplex search was applied for cluster analysis in [81]. Another algorithm based on the combination of Genetic Algorithm (GA), k-means and logarithmic regression expectation maximization was introduced in [82]. In [83], a k-means algorithm that performs correct clustering without preassigning the exact number of clusters was proposed. A genetic k-means algorithm for cluster analysis was introduced in [84]. In [85], a GA based method to solve the clustering problem and experiment on synthetic and real life data sets to evaluate the performance was proposed. A GA algorithm that exchanges neighboring centers for k-means clustering has been introduced in [86]. A combination of evolutionary algorithm with an ant colony algorithm for the clustering problem was presented in [86, 87]. A clonal selection based method has been combined with FCM in [88].

PSO has also been applied to data clustering. In particular, two methods called PSO-V and PSOU are introduced in [89], whereby a reformulated objective function of fuzzy c-means is minimized by the PSO algorithm for the cluster analysis task. Another PSO-based fuzzy clustering algorithm is introduced to overcome the shortcomings of FCM in [90]. An ant colony clustering algorithm is applied for solving the clustering task in [91]. The algorithm uses the global pheromone update and heuristic information to find clustering solutions. In [92], a genetic fuzzy K-modes algorithm for clustering categorical data is proposed, which uses a genetic algorithm to obtain the global optimal clustering solution. A hybrid data clustering algorithm that uses the merits of PSO and K-harmonic means is proposed in [93]. The hybrid algorithm helps to escape from local optima, and thus overcomes the problem of slow convergence of the PSO algorithm. A hybrid evolutionary algorithm, called FAPSO-ACO-K, is introduced in [94]. The hybrid algorithm combines PSO, ACO and k-means applied to cluster analysis. Another method for dynamic parameter adaptation in PSO is proposed in [95]. The proposed algorithm uses fuzzy logic to improve the convergence and diversity of the swarm in PSO.

The high computational cost and the slow convergence rate severely limit the use of PSO on clustering analysis. For these reasons, a chaotic map PSO with an accelerated convergence rate strategy was introduced in [96]. The algorithm works by adopting chaotic maps and adaptive action to avoid local minima. In [97], a hybrid fuzzy clustering method based on FCM and FPSO is proposed to overcome the shortcomings of PSO. Another modified version of PSO, known as Multi-Elitist PSO (MEPSO), is proposed in [98]. This approach solves the hard clustering problem by automatically determining the optimal number of clusters. This approach shows that PSO is guaranteed to solve clustering problems automatically.

This chapter addresses the shortcoming of the FCM algorithm , which is the predefined cluster count. A clustering approach based on PSO [99] is proposed whose aim it is to automatically determine the optimal number of clusters using a threshold vector. The algorithm partitions the data set randomly (within a preset maximum number of clusters) and uses a reconstruction criterion to evaluate the performance of the clustering results.

4.2. Fuzzy C-Means and Particle Swarm Optimization

4.2.1. Fuzzy C-means Clustering

Fuzzy clustering is a method of clustering which allows one piece of data to belong to two or more clusters. The FCM algorithm is an iterative partition clustering technique which was first introduced by Dunn [77] and was extended by Bezdek [24]. FCM is a pretty standard least squared error model that generalizes an earlier and very popular non-fuzzy c-means model that produces hard clusters of the data. An optimal c partition is produced iteratively by minimizing the weighted within group sum of squared error objective function:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c} (u_{ij})^m d^2(y_i, c_j) \tag{4.1}$$

where $Y = [y_1, y_2, ..., y_n]$ is the data set in a d-dimensional vector space. n is the number of data items. c is the number of clusters which is defined by the user where $2 \leq c \leq n$. u_{ij} is the degree of membership of y_i in the j^{th} cluster. m is a weighted exponent on each fuzzy membership. c_j is the center of cluster j. $d^2(x_i, c_j)$ is a square distance measure between object y_i and cluster c_j. An optimal solution with c partitions can be obtained via an iterative process which is as follows:

1. Input(c, m, ϵ, data)

2. Initialize the fuzzy partition matrix $U = [u_{ij}]$

3. Iteration starts and set t=1

4. Calculate the c cluster centers with U^t:

$$c_i = \frac{\sum_{i=1}^{n}(u_{ij})^m y_i}{\sum_{i=1}^{n}(u_{ij})^m} \tag{4.2}$$

5. Calculate the membership U^{t+1} using:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}(\frac{d_{ij}}{d_{kj}})^{\frac{2}{(m-1)}}} \tag{4.3}$$

6. If the stopping criteria is not met, $t = t + 1$ and go to Step 4.

4.2.2. Particle Swarm Optimization

PSO was originally designed and introduced by Eberhart and Kennedy [6]. The PSO is a population search algorithm which intends to simulate the choreography of a bird folk. Each individual, called particle, within the swarm is represented by a vector in a multidimensional search space. A velocity vector is assigned to each particle to determine the next movement of the particle. Each particle updates its velocity based on the current velocity, best personal position it has explored so far and the global best position explored by the swarm:

The velocity and position of the particle at next iteration is updated as:

$$V_i(t+1) = w \cdot V_i(t) + c_1 \cdot r_1 \cdot (X_i^l(t) - X_i(t)) + c_2 \cdot r_2 \cdot (X^g - X_i(t)) \tag{4.4}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{4.5}$$

for the i^{th} particle, where w is the inertia weight, $V_i(t)$ is the previous velocity in iteration t of i^{th} particle. c_1 and c_2 are coefficients. Generally, r_1 and r_2 are random numbers between 0 and 1. $(X_i^l(t) - X_i(t))$ is the difference between local best X_i^l of the i^{th} particle and previous position $X_i(t)$. Similarly, $(X^g - X_i(t))$ is the difference between global best X^g and previous position $X_i(t)$.

4.3. Proposed Approach

The proposed algorithm is based on PSO and FCM. The particle encoding, velocity encoding, decoding and clustering validation is described separately. The procedures of the proposed algorithm are presented at the end of the section.

4.3.1. Particle Encoding

A particle is a $2 \times k$ matrix, where k is the maximum number of clusters that is predefined. The first row represents the centers. Each value in the second row controls the activation of each center in the first row.

$$X_i = \begin{pmatrix} x_{1,1}^i & x_{1,2}^i & ... & x_{1,k}^i \\ t_{2,1}^i & t_{2,2}^i & ... & t_{2,k}^i \end{pmatrix} \tag{4.6}$$

where $x_{1,k}^i$ represents the i^{th} particle's position in cluster k. $x_{1,k}^i$ should be in the range of $[x_{min}, x_{max}]$. $t_{2,k}^i$ is the i^{th} particle's threshold value in the range of $[0, 1]$. If the threshold value is greater than 0.5, the center is activated. Otherwise, it is deactivated.

4.3.2. Velocity Encoding

The velocity matrix should have the same dimension as the position matrix with a range. Suppose we set the range as $[v_{min}, v_{max}]$, all values of the velocity matrix should be between v_{min}

and v_{max}. Thus, the i^{th} velocity is denoted as:

$$V_i = \begin{pmatrix} v^i_{x1,1} & v^i_{x1,2} & \cdots & v^i_{x1,k} \\ v^i_{t2,1} & v^i_{t2,2} & \cdots & v^i_{t2,k} \end{pmatrix} \tag{4.7}$$

Similarly, k is the maximum number of clusters. The first row is the velocity of the centers, and the second row is the velocity of the threshold values.

4.3.3. Decoding

$Y = (y_1, y_2, .., y_n)$ is the data set with d dimensions. The cluster centers can be decoded as $C = (c_1, c_2, ...c_k)$ using Equation 4.2.

4.3.4. Clustering Validation Techniques

The aim of clustering validation is to evaluate the clustering results by finding the best partition that fits the underlying data. Thus, cluster validity is used to quantitatively evaluate the results of clustering algorithms. Compactness and separation two widely considered criteria for measuring the quality of the partitioning of a data set into different numbers of clusters. Conventional approaches use an iterative approach by choosing different input values, and they select the best validity measure to determine the “optimum” number of clusters. A list of validity indices for fuzzy clustering is listed below.

1. Dunn’s Index (DI): the Dunn’s Index is proposed to identify the compactness and separation of the clusters. The function that uses to calculate the result of the clustering is as follow:

   $$DI = \min\{\min_{j\in c, i\neq j}\{\frac{\min_{x\in C_i, y\in C_j} d(x,y)}{\max_{k\in c}\{\max_{x,y\in C} d(x,y)\}}\}\} \tag{4.8}$$

   where $d(x, y)$ is the distance of the two cluster centers. DI takes its minimum value when the cluster structure is optimal.

2. Weighted Inter-Intra (Wint) Index: the weighted inter-intra (Wint) measure is introduced by Strehl [106] in 2002. It compares the compactness of the data to its separation.

$$Wint = (1 - \frac{2c}{n}) \cdot (1 - \frac{\sum_i \frac{1}{n-|c_i|} \sum_{j \neq i} inter(c_i, c_j)}{\sum_i \frac{2}{|c_i|-1} intra(c_i)}) \tag{4.9}$$

   where $intra(c_i)$ is the average intra-distance within cluster i. $inter(c_i, c_j)$ is the average inter-distance between cluster i and cluster j. Wint obtains its maximum value when the cluster structure is optimal.

3. Least Squared Error (SE) Index: the weighted within cluster sum of squared error function is used [100]:

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m ||y_i - c_j||^2 \tag{4.10}$$

   where y_i is the i^{th} data point with d dimensions. c_j is the value of the j^{th} cluster, and $||y_i - c_j||$ is the Euclidean distance between y_i and c_j. J_m takes its minimum value when the cluster structure is best.

4. Partition Coefficient (PC) Index: the partition coefficient (PC) is defined as [24]:

$$PC = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^2 \tag{4.11}$$

   PC obtains its maximum value when the cluster structure is optimal.

5. Partition Entropy (PE) Index: the partition entropy was defined as [78]:

$$PE = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij} log_b(u_{ij}) \tag{4.12}$$

   where b is the logarithmic base. PE gets its minimum value when the cluster structure is optimal.

6. Modified Partition Coefficient (MPC) Index: modification of the PC index, which can reduce the monotonic tendency, is proposed by Dave in 1996 [101].

$$MPC = 1 - \frac{c}{c-1}(1 - PC) \tag{4.13}$$

   where c is the number of cluster. An optimal cluster number is found by maximizing MPC to produce a best clustering performance for a data set.

7. Fukuyama and Sugeno (FS) Index: Fukuyama and Sugeno proposed a validity function in 1989 [102]. It is defined as:

$$FS = \sum_{i=1}^{n}\sum_{j=1}^{c}\mu_{ij}^{m}||x_i - c_j|| - \sum_{i=1}^{n}\sum_{j=1}^{c}\mu_{ij}^{m}||c_j - \bar{c}|| \tag{4.14}$$

   where $\bar{c} = \sum_{j=1}^{c} c_j/c$. It measures the separation. The first term equals to J_m which is the least squared error. It measures the compactness. The best clustering performance for a data set is found by maximizing the value of FS.

8. Xie-Beni (XB) Index: Xie and Beni proposed a validity function in 1991 [103], and later it was modified by Bezdek in 1995 [104].

$$XB = \frac{J_m}{n \times \min_{i \neq j} ||z_i - z_j||^2} \tag{4.15}$$

   XB reaches its minimum value when the cluster structure is optimal.

9. Partition Coefficient and Exponential Separation (PCAES) Index: the partition coefficient and exponential separation (PCAES) index [105] is defined as:

$$PCAES = \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{(u_{ij})^2}{u_M} - \sum_{k=1}^{c} exp(-\min_{k \neq i} ||z_i - z_k||^2/\beta_T) \tag{4.16}$$

where $u_M = \min_{1 \le j \le c}\{\sum_{i=1}^{n} u_{ij}^2\}$ and $\beta_T = (\sum_{j=1}^{c} ||z_j - \bar{z}||^2)/c$. $\bar{z} = \sum_{i=1}^{n}(y_i/n)$. PCAES takes its maximum value when the cluster structure is optimal.

The procedure of the proposed algorithm is as follows:

Input: data set $Y = [y_1, y_2, ..., y_n]$, number of cluster c, fuzzification coefficient m.

Output: a $n \times c$ partition matrix U and corresponding centers.

1. Randomly initialize a swarm

2. Iteration starts and set t=1

3. Update the velocity of each particle using Equation 4.4

4. Update the position of each particle using Equation 4.5

5. Update the personal best and global best

6. Calculate the partition matrix U

7. If the stopping criterion is not met, $t = t + 1$ and go back to Step 3)

8. The partition matrix U of the global best is used to reconstruct the original data

9. Calculate the reconstruction error. In order to use a consistent method to evaluate the eight different indices, the reconstruction criterion (RC) [107] is used. The reconstruction criterion uses the cluster prototypes and partition matrix to "reconstruct" the original data vectors. The reconstructed version of the original data vectors, $\hat{Y} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_n]$, is calculated as:

$$\hat{y}_i = \frac{\sum_{j=1}^{c} u_{ij}^m c_j}{\sum_{j=1}^{c} u_{ij}^m} \tag{4.17}$$

once the reconstruction has been finished, the squared error of the reconstruction vectors and original vectors are evaluated using Equation 4.18.

$$E = \sum_{i=1}^{n} ||\hat{y}_i - y_i||^2 \tag{4.18}$$

10. Select the partition matrix and centers corresponding to the minimum reconstruction error.

4.4. Experiments and Results

In this section, the experimental setup, datasets and experimental study are described respectively.

4.4.1. Experimental Setup

The experiments are implemented and evaluated on an ASUS desktop (Intel(R) Dual Core I3 CPU @3.07 GHz, 3.07 GHz) Matlab Version 7.13. All measurements of the proposed algorithm are executed 30 times and the average is taken. The parameters required for the proposed algorithm are listed in Table 24.

Table 24. Parameters and their values of the proposed algorithm.

Parameter	Value
Maximum number of cluster	10
Maximum iteration	50
Swarm size	25
Maximum run	30
Fuzzification coefficient (m)	2

4.4.2. Datasets

The experiments are conducted on a number of datasets taken from the UCI repository [54], and synthetic data sets were generated using Matlab. The datasets are described in Table 25.

Table 25. Datasets used for the experiments.

Data Set	Dimensions	Instances	Classes
Pinwheel	2	1000	2
Transfusion	4	748	2
Haberman	3	306	2
Breast-W	9	699	2
Jain	2	373	2
Thyroid	5	215	2
Iris	4	150	3
DIM032	32	320	5
DIM064	64	320	5
DIM128	128	320	5
DIM256	256	320	5

4.5. Experimental Study

4.5.1. Use of Synthetic Data

In order to investigate the clustering performance with different numbers of clusters, we use a synthetic data set, named pinwheel, to test the clustering performance using K-means [109], K-medoid [110], FCM, Gustafson-Kessel (GK) and our proposed algorithm (FPSO). K-means is one of the unsupervised learning methods that uses an iterative refinement technique. The number of desired cluster, k, is defined in advance. K-medoid is another unsupervised learning method related to the K-means algorithm. Similarly, the K-medoid classifies the data set into k clusters. However, K-medoid is more robust to noise and outliers as compared to K-means. Unlike K-means, the medoid is defined as the data point whose average dissimilarity within the cluster is minimal.

K-means and K-medoid are traditional hard clustering techniques, while FCM, GK and FPSO are soft clustering techniques. The nine validity indices listed in Equations 4.8-4.16 are used.

The cluster performance of the pinwheel data set is displayed in Figure 12. The first figure is the original data set. As can be seen, the cluster centers are different using Kmeans, K-medoid, FCM, GK and FPSO, respectively.

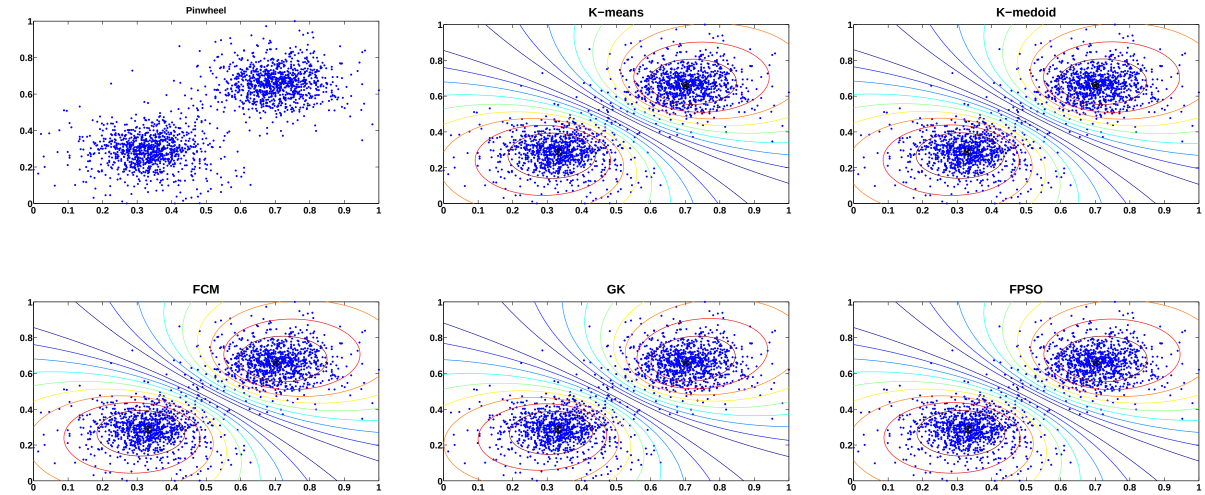

Figure 12. Synthetic data set

Figure 13 shows the performance of the pinwheel data set using the K-means algorithm. Figure 14 shows the performance of the pinwheel data set using the K-medoid algorithm. Since K-means and K-medoid are algorithms using hard partitioning, the DI index, Wint index and SE index are used for validation.

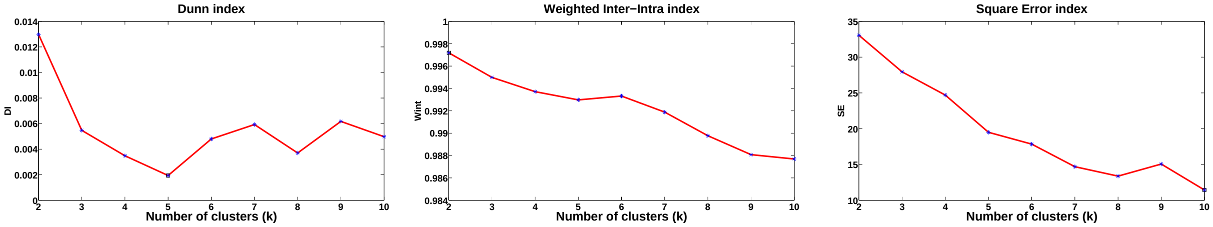

Figure 13. Kmeans using 3 different validity indices

In Figure 15, the performance of the pinwheel data set using FCM is given. Figure 16 shows the performance of the pinwheel data set using the GK algorithm. The performance of the proposed algorithm (FPSO) is displayed in Figure 17. The correct number of clusters found for the nine indices are listed in Table 26. The correct cluster number for the pinwheel data set is 2. The correct number of clusters found measuring DI using the five different algorithms are 5, 5, 7,

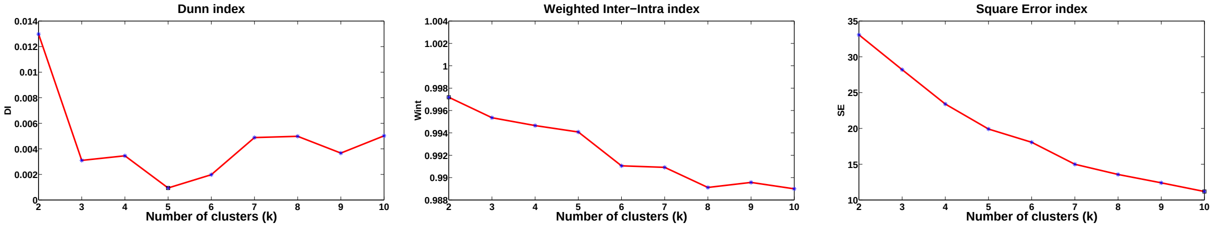

Figure 14. K-medoid using 3 different validity indices

Table 26. Index values with varying c using pinwheel data set.

Indices	Kmeans	K-medoid	FCM	GK	FPSO
DI	5	5	7	10	7
Wint	2	2	2	2	2
SE	10	10	10	10	10
PC	-	-	2	2	2
PE	-	-	2	2	2
MPC	-	-	2	2	2
FS	-	-	10	10	10
XB	-	-	10	7	10
PCAES	-	-	4	10	8

10, and 7, respectively. The correct number of clusters found applying Wint are consistent with the correct cluster number. The correct number of clusters found by SE is consistent with 10. As the number of clusters increases, the SE values decrease. PC, PE and MPC using the FCM, GK and FPSO algorithms find the correct cluster number. Measuring FS using the FCM, GK and FPSO algorithm are similar to SE. As the number of clusters increases, the FS values decrease. The correct number of clusters found applying XB using the FCM, GK and FPSO algorithms are 10, 7, and 10, respectively. The correct number of clusters found measuring PCAES using the FCM, GK and FPSO algorithms are 4, 10, and 8, respectively. Overall, the Wint, PC, PE and MPC indices, which outperform the other indices, find the correct number of clusters.

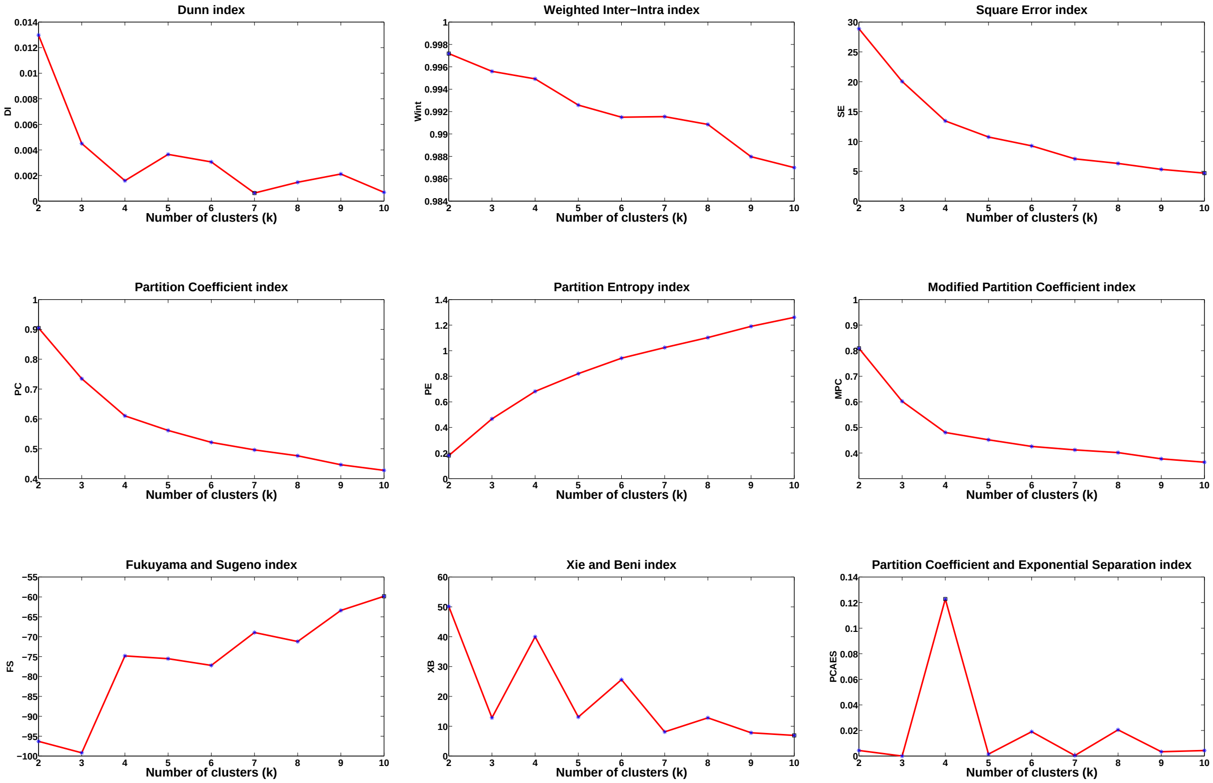

Figure 15. FCM using 9 different validity indices

4.5.2. Use of Real-World Data

In this section, we investigate the behavior of the clustering results using nine different validity measures listed before.

In Table 27, the reconstruction errors of the transfusion data set, where c ranges from 2 to 9, have been calculated using the proposed algorithm by applying Equations 4.8-4.16. The reconstruction errors can simply the results for comparison purpose. As shown by the results, the values in bold identifying the minimum reconstruction errors with different cluster numbers for each measure. 6 out of 8 cases show that $c = 2$ is the correct number of clusters. This indicates that the proposed FPSO can find the best number of clusters automatically.

Due to the stochastic nature of our proposed algorithm, we tested the proposed algorithm on 30 runs and calculated the average number of clusters as listed in Table 33. The standard deviation

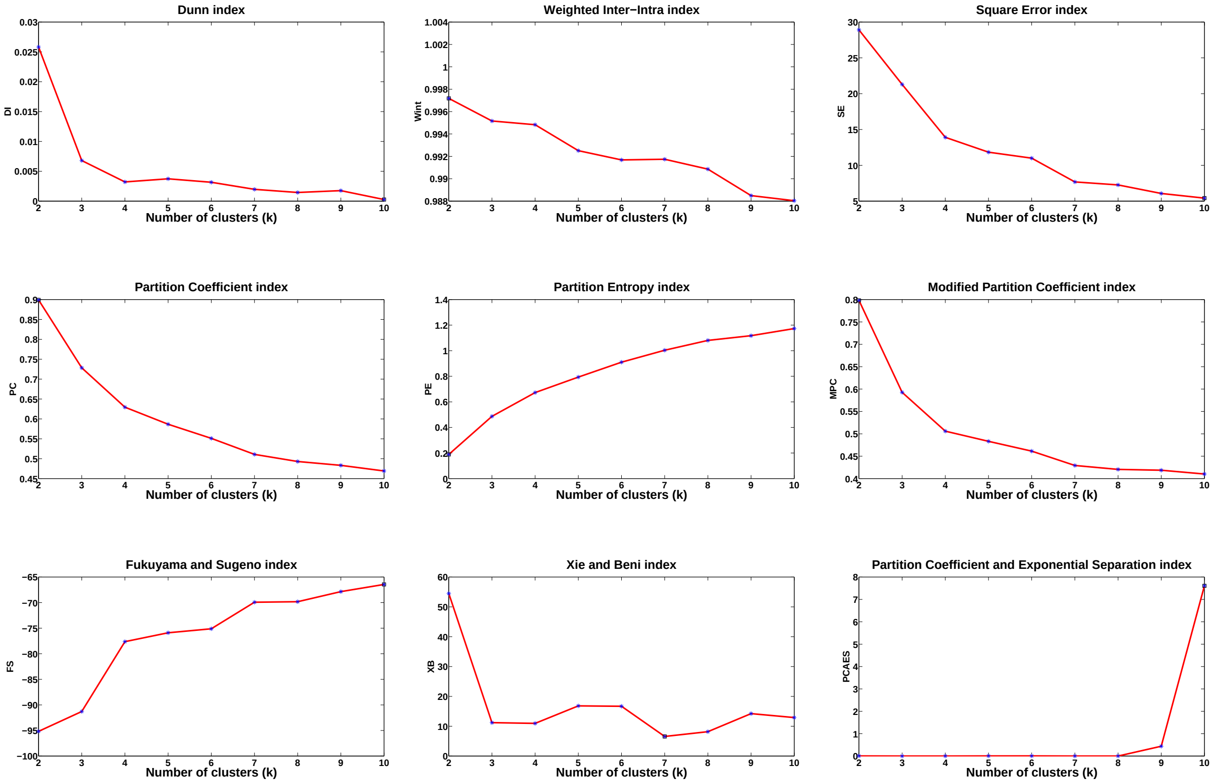

Figure 16. GK using 9 different validity indices

values are given as well. The correct numbers of clusters using different validity measures are tabulated, respectively. In all the cases, the number of clusters predicted by FPSO is close to the correct number of clusters. DI, SE, XB and PCAES do not find the correct number of clusters. Wint can identify the correct number of clusters but only for low-dimensional datasets. MPC returns the correct number of clusters, but with larger standard deviation values. PC and PE find the correct number of clusters consistently, however, as the number of dimension increases, the accuracy decreases.

4.5.3. Visualization of Clustering Results

Since the validity measures reduce the overall evaluation to a certain number, therefore there is some loss of information. In order to better analyze the results, a low-dimensional graphical representation of the clusters is adopted. A toolbox implemented by [108] is used to visualize the

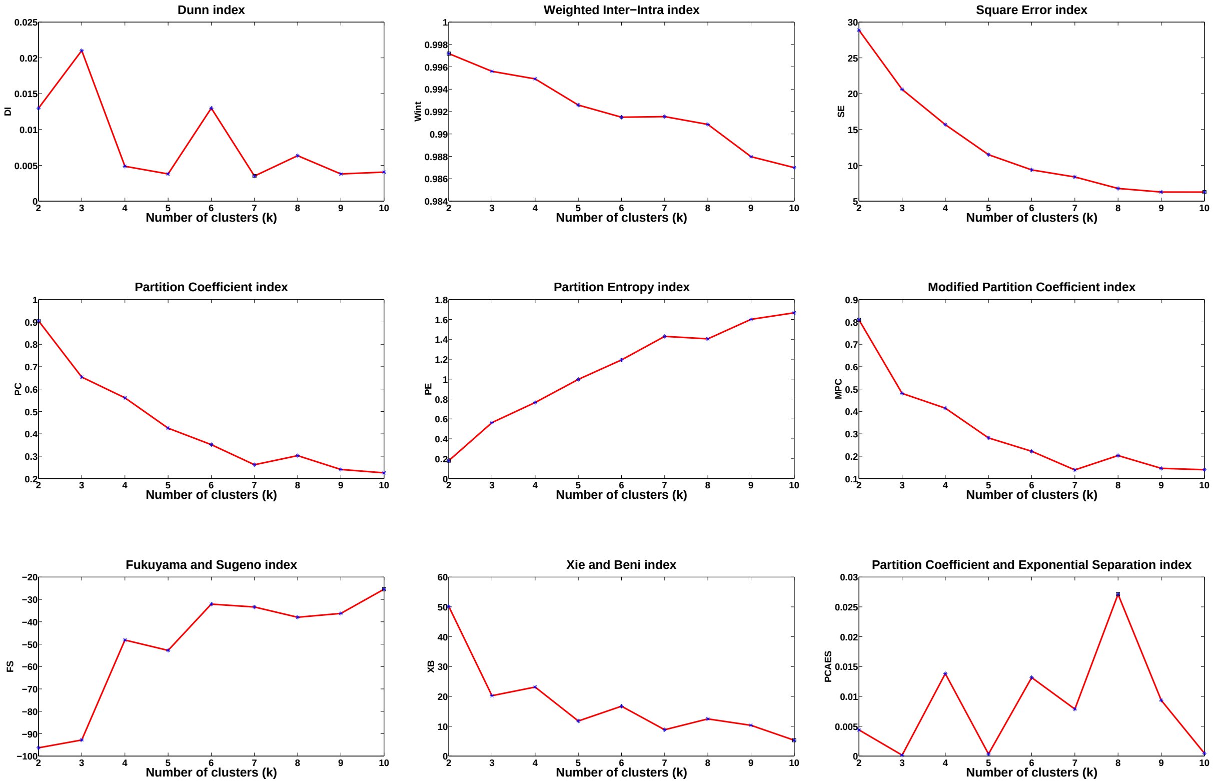

Figure 17. FPSO using 9 different validity indices

cluster results using the proposed algorithm. Principal Component Analysis (PCA) projection, Conventional Sammon Mapping (CSM), and Fuzzy Sammon Mapping (FSM) are used. The dimensions of DIM032, DIM064, DIM128 and DIM256 are 32, 64, 128 and 256, respectively. The correct number of clusters is 5. Figure 18 lists the performance of PCA, CSM and FSM using the four data sets. The black cross represents the identified cluster centers. Obviously, the correct number of clusters can be seen by looking at the figure. As the number of dimension increases, the performance of the PCA, CSM and FSM decrease.

Table 27. Reconstruction error with varying c using transfusion data set.

	c=2	3	4	5	6	7	8	9
DI	31.2	30.4	**24.5**	61.3	31.3	**24.5**	59.8	25.0
Wint	**23.6**	29.3	25.4	47.4	51.5	68.8	42.5	36.5
SE	**23.6**	25.6	26.5	23.6	28.7	33.7	24.3	25.9
PC	**16.3**	29.6	34.2	81.8	22.7	75.1	31.1	21.4
PE	26.3	85.0	26.0	57.0	24.9	39.3	**16.9**	30.3
MPC	17.0	44.8	24.6	60.4	93.3	**12.2**	166.7	12.5
FS	**23.6**	**23.6**	**23.6**	25.6	29.5	24.9	33.6	23.7
XB	**23.6**	43.0	25.1	36.1	27.1	61.3	68.8	63.1
PCAES	**38.9**	81.7	82.5	48.1	53.8	57.5	97.2	93.4

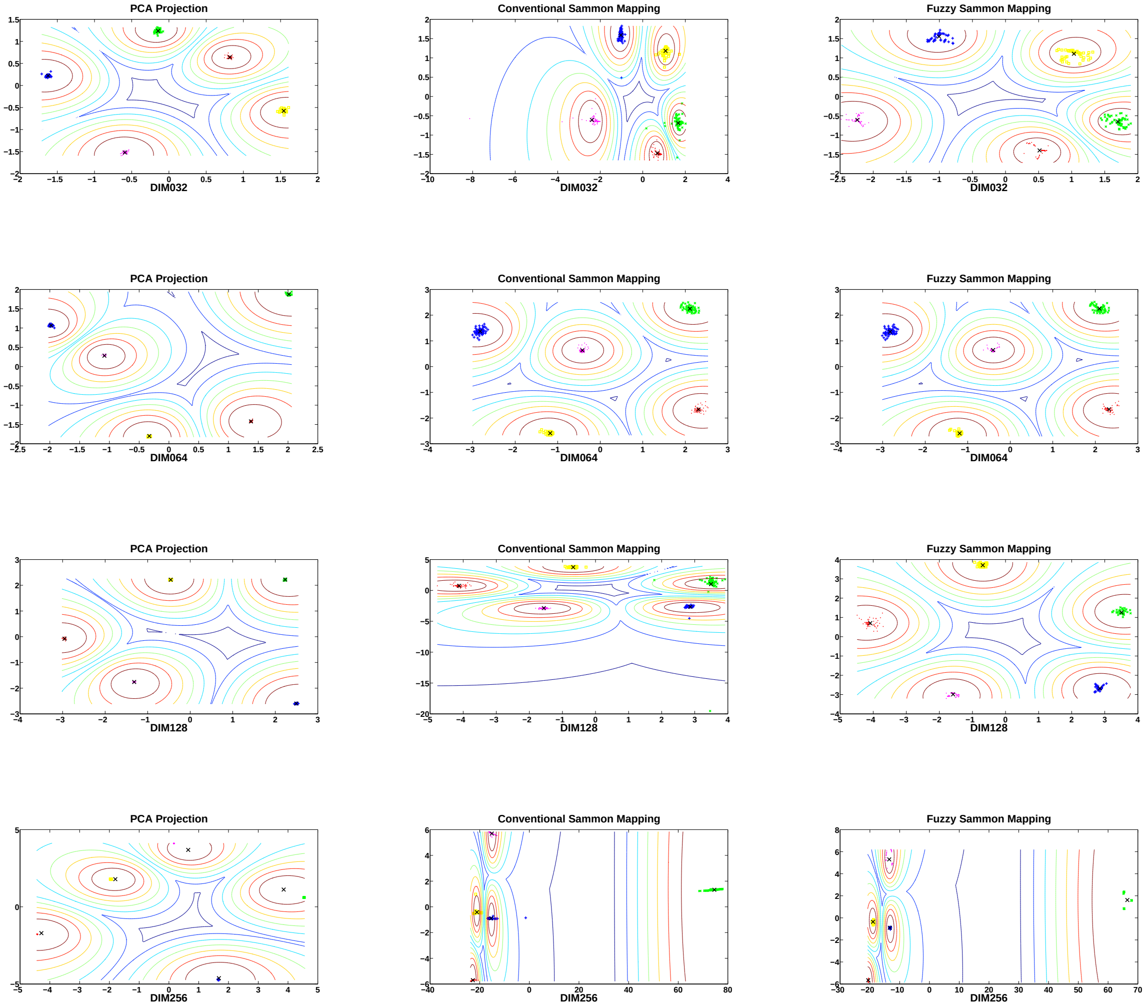

Figure 18. Data sets using PCA projection, CSM and FSM, respectively.

Table 28. Nine different indices using the proposed algorithm.

	DI	Wint	SE	PC	PE	MPC	FS	XB	PCAES	FPSO	k
Transfusion	5.70	2.63	9.43	2.17	2.13	3.67	2.73	9.63	6.87	2.42	2
std.	1.84	0.49	0.86	0.38	0.35	2.15	1.05	0.56	2.29	0.58	
Haberman	5.90	2.93	9.97	2.00	2.00	6.50	2.07	9.83	8.80	2.24	2
std.	0.55	0.25	0.18	0.00	0.00	1.50	0.25	0.75	2.51	0.15	
Breast	6.13	3.00	8.43	3.27	2.17	5.30	2.13	9.77	6.83	2.64	2
std.	1.96	0.00	1.43	1.72	0.38	2.45	0.35	0.63	2.09	0.62	
Jain	7.30	2.97	9.57	2.87	2.10	3.60	3.37	7.90	6.47	2.25	2
std.	1.64	0.56	0.68	0.82	0.31	1.52	2.75	2.20	2.22	0.15	
Thyroid	4.27	2.83	9.90	2.10	2.03	4.27	2.00	9.53	8.10	2.24	2
std.	1.87	0.38	0.31	0.40	0.18	2.21	0.00	0.68	1.99	0.24	
Iris	3.83	2.93	9.00	2.53	2.30	4.13	2.40	9.57	6.30	2.55	2
std.	1.58	0.52	1.62	0.94	0.47	2.06	0.89	0.57	2.58	0.7	
DIM032	6.33	2.73	7.90	4.60	4.00	4.90	3.73	7.40	6.40	5.34	5
std.	1.07	0.64	1.83	1.43	1.91	1.44	1.10	1.30	2.04	2.08	
DIM064	7.17	3.20	7.77	5.47	5.50	5.53	2.00	9.43	7.40	5.94	5
std.	2.63	1.92	1.83	1.33	1.41	1.46	0.00	1.01	1.92	1.50	
DIM128	6.90	2.57	8.27	5.63	5.57	5.77	2.00	8.37	8.03	5.90	5
std.	2.02	0.57	1.55	1.25	1.38	1.04	0.00	1.97	1.35	1.24	
DIM256	8.57	2.57	8.57	6.43	6.13	6.70	2.00	9.13	8.57	6.52	5
std.	1.19	0.50	1.43	1.65	1.72	1.56	0.00	1.07	1.33	1.16	

The performance of the mapping are listed in Table 29. The mean square error of the re-calculated membership values (P), two different original and re-calculated validity measures (F and F*), and the Sammon stress coefficient (S) are listed in the parenthesis. As the number of dimension increases, the FSM is better than PCA and CSM in terms of smaller P, F* and S values. The performance of PCA, CSM and FSM are the same in terms of F values.

4.6. Summary

In this study, we proposed an algorithm to overcome the drawbacks of traditional partition clustering, which is that the number of clusters needs to be predefined. The proposed algorithm uses using PSO and FCM with a threshold vector to control and identify the optimal number of clusters. The algorithm solves the clustering problem via an iterative fuzzy partition process.

Table 29. Mapping using the proposed algorithm.

Datasets	PCA	CSM	FSM
DIM032	(0.0030 0.983 0.9975 0.082)	(0.0104 0.983 0.950 0.295)	(0.0023 0.983 0.974 0.078)
DIM064	(0.0011 0.994 0.9995 0.133)	(0.0005 0.994 0.993 0.055)	(0.0005 0.994 0.992 0.056)
DIM128	(0.0008 0.996 0.9998 0.125)	(0.0050 0.996 0.977 1.560)	(0.0004 0.996 0.994 0.050)
DIM256	(0.0046 0.200 0.2010 0.114)	(0.0262 0.200 0.208 34.28)	(0.0044 0.200 0.200 0.048)

For the evaluation of our algorithm, we generated a synthetic dataset as well as used 6 datasets from the UCI repository. We compared our algorithm with hard clustering approaches such as Kmeans and K-medoid as well as with fuzzy clustering algorithms such as FCM and GK. Nine different validity indices were used to evaluate the performance. Furthermore, measures such principal component analysis projection, conventional sammon mapping, and fuzzy sammon mapping were used to visualize the clustering results.

Overall, the results show that the proposed algorithm can identify the correct number of clusters on all the data set tested. However, due to the slow convergence and the stochastic nature of the PSO algorithm, the prediction results of a single run vary and thus make it difficult to predict the correct number of clusters. Unlike K-means and FCM, the proposed algorithm needs to be executed repeatedly in order to find the correct solution. In addition, the maximum number of clusters has to be predefined, and the iterative process to identify the optimal number of clusters is computationally expensive by comparing to K-means and FCM.

5. COLOR IMAGE SEGMENTATION USING FUZZY C-REGRESSION MODEL

Image segmentation is one important process in image analysis and computer vision, and is a valuable tool that can be applied in fields of image processing, health care, remote sensing, and traffic image detection. Given the lack of prior knowledge of the ground truth, unsupervised learning techniques like clustering have been largely adopted. Fuzzy clustering has been widely studied and successfully applied in image segmentation. In situations such as limited spatial resolution, poor contrast, overlapping intensities, noise and intensity inhomogeneities, fuzzy clustering can retain much more information than the hard clustering technique. Most fuzzy clustering algorithms have originated from Fuzzy C-Means (FCM) and have been successfully applied in image segmentation. However, the cluster prototype of the FCM method is hyper-spherical or hyper-ellipsoidal. FCM may not provide the accurate partition in situations where data consists of arbitrary shapes. Therefore, in this chapter, a Fuzzy C-Regression Model (FCRM) has been proposed whose prototype is hyper-planed and can either be linear or nonlinear allowing for better cluster partitioning. Thus, this chapter implements fuzzy c-regression model clustering algorithm and applies the algorithm to color segmentation using Berkeley's segmentation database. The results show that FCRM obtains more accurate results compared to other fuzzy clustering algorithms.

The remainder of this chapter is organized as follows. Section 5.1 lists the related work regarding fuzzy image partitioning. Section 5.2 describes the fuzzy c-regression model and the proposed approach applied to color image segmentation. Experimental results are presented in Section 5.4, and conclusions are drawn in Section 5.7.

5.1. Related Work

Related work with regards to the use of fuzzy theory in image segmentation include rule-based methods, fuzzy-geometrical methods, information theoretical methods, Type II thresholding methods, and fuzzy clustering methods [111].

Past research related to rule-based methods use fuzzy rules to determine a threshold value in image segmentation. Images are considered as typical nonstationary signals. Fuzzy rule-based image processing techniques are applied to noise removal and edge extraction. A novel approach for enhancing the results of fuzzy clustering for solving image segmentation problems is introduced in [112]. A Sugeno-type rule-based system is developed to interact with the clustering result obtained by the FCM algorithm. In [113], an approach which combines an associative restoration algorithm with a fuzzy image enhancement technique is presented and is applied in electronic portal images in radiotherapy. However, fuzzy rule based segmentation is sensitive to both the structure of the membership functions and parameter value selections. Thus, a generic fuzzy rule-based segmentation technique that tries to solve the problem of manual selection of the parameters of the fuzzy membership is introduced in [114]. This proposed technique is application-independent and incorporates spatial relationships between pixels. Fuzzy Rules for Image Segmentation incorporating Texture features (FRIST) is proposed in [115]. The fractal dimension and contrast features of texture are incorporated in FRIST by considering image domain specific information.

Fuzzy-geometrical methods [116], which focus on local image information, minimize or maximize fuzzy geometrical measures, such as compactness [117]. In [118], a new approach to multidimensional data clustering is described. The approach developed a “Radar” diagram shape matching methodology to accomplish the fuzzy geometric features technique for man-machine expert systems. A new quantitative index for image segmentation using the concept of homogeneity within regions is defined in [120]. The proposed index shows that the fuzzy geometry based thresholding algorithms produced a single stable threshold for a wide range of membership variations. A semi-supervised FCM technique called GG-FCM is used to add geometrical information during clustering [119]. The approach is not only based on spectral information obtained by FCM, but also takes into consideration the geometrical relationship between neighboring pixels.

Related work on information theoretical methods uses measurements such as fuzzy entropy, index of fuzziness, and fuzzy divergence to minimize or maximize fuzzy information. In [121], a new measure called divergence between two fuzzy sets is introduced and a tailored version of the probability measure of a fuzzy event is also used for image segmentation. A complete method can be viewed as a weighted moving average technique, greyness ambiguity being the weights is introduced in [122]. An image thresholding approach based on the index of nonfuzziness maximization of the 2-D grayscale histogram is introduced in [123], and has shown that the approach is more robust when applied to noisy images.

Type II thresholding methods interpret image information as Type II fuzzy sets. These methods use information-theoretical measures to locate a global threshold [111]. In [124], an evolving fuzzy classifier approach that is able to adapt and evolve at an on-line machine vision system is introduced. In [125], a new modified thresholding measures for MRI brain images using type-1 and type-2 fuzzy sets is presented. An Interval Type 2 (IT2) fuzzy entropy based approach is used to compute optimum thresholds for multistage gray scale image segmentation in [126]. An automatic leukocyte segmentation using intuitionistic fuzzy and interval Type II fuzzy set theory in pathological blood cell images is presented in [127]. The use of intuitionistic fuzzy set and interval Type II fuzzy set can consider more uncertainties and different types of uncertainty as compared to basic fuzzy set theory.

Fuzzy clustering methods classify all image pixels into different segments. Up to now, FCM is one of the most commonly used methods in image segmentation, and there have been many variants of fuzzy clustering algorithms that originated from FCM. A modified fuzzy c-means clustering algorithm for MR brain image segmentation is introduced in [128]. The proposed algorithm extracts a scalar feature value from the neighborhood of each pixel. It converges faster than standard FCM in the case of mixed noise. An improved FCM algorithm for image segmentation, which introduces a tradeoff weighted fuzzy factor and a kernel metric is introduced in [129]. The proposed algorithm using a tradeoff weighted fuzzy factor can accurately estimate the damping extent of neighboring pixels. FCM is sensitive to noise in the image since it ignores the spatial information contained in the pixels. A novel fuzzy clustering algorithm with non-local adaptive spatial constraints is presented in [130]. The approach uses an adaptive spatial parameter for each pixel to guide the noisy image segmentation process. Reference [131] proposes the weighted image patch-based FCM algorithm for image segmentation. The algorithm improves its robustness to noise by incorporating local spatial information embedded within the segmentation process. In color image segmentation, it is difficult to analyze the image on all of its colors. Soft computing techniques namely FCM, possibilistic fuzzy c-means, and competitive neural networks have been used to group likely colors [132]. A novel initialization scheme to determine the cluster number and obtain the initial cluster centers for the FCM algorithm to segment color images is introduced in [135]. The initialization scheme called hierarchical approach is proposed to integrate the splitting and merging techniques to obtain the initialization condition for FCM. The proposed algorithm can obtain the reasonable cluster number for any kind of color images. An Adaptive Neuro-Fuzzy Color Image Segmentation (ANFCIS) approach is presented in [136]. The proposed algorithm performs color image segmentation using multilevel thresholding, which consists of a multilayer perceptron-like network.

Most fuzzy clustering algorithms have originated from FCM, and have been successfully applied in image segmentation. However, the cluster prototype of the FCM method is either hyper-spherical or hyper-ellipsoidal. FCM may not provide the accurate partition in situations where data consists of arbitrary shapes. On the other hand, the prototype of the FCRM method is hyper-planed and can either be linear or nonlinear. Thus, this chapter implements FCRM and applies it to color segmentation of images. This is the first work applying the FCRM method to the color segmentation. The results show that FCRM obtains more accurate results compared to other fuzzy clustering algorithms. Furthermore, besides presenting FCRM's competitiveness with respect to the other fuzzy clustering algorithms, FCRM's practical value is demonstrated when applied to the task of color image segmentation.

5.2. Proposed Approach

This section first describes the color space that is used for the proposed color segmentation approach, followed by the proposed fuzzy c-regression model clustering approach, and the cluster validation techniques used for the evaluation of the approach.

5.2.1. CIE-L*A*B* Color Space

Color space is a way of representing color information based on certain criteria. Color perceived by human-beings combines primary colors which are R (red), G (green), and B (blue). By using either linear or nonlinear transformations, other kind of color representations or spaces can be derived from the R, G, and B representation [137]. Color spaces like RGB, HSV (Hue-Saturation-Value) [138], and CIE-L*A*B* [139] have been successfully applied in color image segmentation. In this chapter, the CIE-L*A*B* color space is selected and explored in color image segmentation. CIE-L*A*B* is a color-opponent space with dimensions L, A, and B. L denotes as lightness, and A and B are the color-opponent dimensions. The CIE-L*A*B* color space includes all perceivable colors and it is device independent, which means that the colors are independent of the device they are displayed on. Specifically, L with a range between 0 and 100 represents the lightness; 0 represents the darkest black, while 100 represents the brightest white. The red-green opponent colors are represented by the A axis. The yellow-blue opponent colors are represented by the B axis. Both A and B have negative and positive values. Negative values of A represent green colors while positive values of A represent red colors. Similarly, negative values of B represent yellow colors, and positive values of B represent blue colors. The range of A and B can be either ± 100 or ± 128 depending on the specific implementation.

5.2.2. Fuzzy C-Regression Model Clustering

The fuzzy c-regression model clustering algorithm has become popular the past few years since the resulting model can explain and describe complex systems in a human intuitive way.

Takagi and Sugeno [140] introduced the well-known T-S fuzzy model to describe a complicated nonlinear system. A T-S fuzzy model consists of a set of fuzzy rules, each describing a local input-output relation as follows.

Rule i: IF x_1 is A_1^i and ... and x_M is A_M^i THEN

$$y_i = \theta_i^0 + \theta_i^1 x_1 + ... + \theta_i^M x_M \tag{5.1}$$

where $X = [x_1, ..., x_M]$ is the system input, M is the dimension of input vector, $i = 1, ..., c$ is the number of fuzzy rules, y_i is the i^{th} output, θ_i^M is the consequent parameter of the i^{th} output.

Fuzzy clustering as one of the soft computing techniques can allow the data points to belong to more than one cluster. Fuzzy clustering has been successfully applied in data analysis, pattern recognition, and image segmentation [132]. The shell clustering algorithms such as FCM have been largely applied in image segmentation. The shell clustering algorithms detect the special geometrical shapes like circles, rectangles, hyperbolas, and ellipses using the Euclidean distance measure [132]. Unlike the shell clustering algorithms, the Fuzzy C-Regression Model (FCRM) [133, 134], which was introduced by Hathaway and Bezdek in 1993, assumes that the data is drawn from c different models instead of one single model. The c different models represent c hyper-plane-shape clusters. The FCRM clustering algorithm is an affine T-S model with linear prototypes.

Let $S = (x(k), y_k), k = 1, ..., N$ be a set of input-output sample data pairs, where N is the number of patterns, $x_k = [x_1, x_2, ..., x_M] \subset R^n$ is the k^{th} input data vector, M is the number of input variables, y is output vector, y_k is the k^{th} desired output for x_k, and $\theta_i = [b_i^0, b_i^1, ..., b_i^M]$ is the parameter vector of the corresponding local linear model. Assume that the data pairs in S are drawn from c different fuzzy models. The i^{th} hyper-plane-shaped cluster of the k^{th} input can be denoted as:

$$
\begin{aligned}
y_k^i &= b_i^0 + b_i^1 x_{k1} + ... + b_i^M x_{kM} \\
&= [x_k, 1] \cdot \theta_i^T, i = 1, ..., c
\end{aligned} \tag{5.2}
$$

The cost function of the FCRM clustering algorithm is defined as:

$$
J(S; U, \theta) = \sum_{k=1}^{N} \sum_{i=1}^{c} (\mu_{ik}^m) E_{ik}^2(\theta_i) \tag{5.3}
$$

where the distance $E_{ik}(\theta_i)$ is defined as

$$
E_{ik}(\theta_i) = |y_k - [x_k 1] \cdot \theta_i^T| \tag{5.4}
$$

m is fuzzy weighted exponent and μ_{ik} is the membership degree of x_k to the i^{th} hyper-plane-shaped cluster. The membership values μ_{ik} have to satisfy the following constraints:

$$
\mu_{ik} \in [0\ 1], i = 1, 2, ..., c; k = 1, 2, ..., N \tag{5.5}
$$

$$
\sum_{i=1}^{c} \mu_{ik} = 1, k = 1, 2, ..., N \tag{5.6}
$$

The fuzzy c-regression model clustering algorithm is summarized as follows [133, 134]. Given data S, set $m > 1$ and specify the regression models, choose an error measure and a termination threshold $\epsilon > 0$, and initialize $U^{(0)}$ randomly.

1. Repeat for $l = 1, 2, ..., \infty$

2. Calculate the c model parameters $\theta_i^{(l)}$, which globally minimizes the cost function

3. Update $U^{(l)}$ with $E_{ik}(\theta_i^{(l)})$ to satisfy

$$U_{ik}^{(l)} = \begin{cases} [\sum_{j=1}^{c} (\frac{E_{ik}}{E_{jk}})^{\frac{2}{m-1}}]^{-1}, & \text{if } E_{ik} > 0 \text{ for } 1 \leq i \leq c. \\ 0, & \text{otherwise} \end{cases} \quad (5.7)$$

4. Until $||U^{(l)} - U^{(l-1)}|| \leq \epsilon$, then stop; otherwise, $l = l + 1$ and return to Step 1

In this chapter, the FCRM clustering algorithm is applied to color image segmentation. The procedures of the proposed approach using FCRM in color image segmentation can be summarized into four phases: image pre-processing, FCRM clustering, image reconstruction, and evaluation.

Image pre-processing: the images are converted from the RGB color space to the CIE-L*A*B* color space during this phase. The *A and *B values, which are extracted from the RGB color space, serve as the color markers in the A*B* space.

FCRM clustering: the A*B* space image data is given, and the number of clusters is fixed during this phase. A FCRM clustering algorithm is used to partition the given data into a fixed number of clusters.

Image reconstruction: the cluster results from the FCRM clustering step is used to reconstruct the image in grayscale-level during this phase.

Evaluation: the performance of the cluster results is evaluated using the results from the FCRM clustering process. The performance of the proposed algorithm is evaluated with three validity indices (explained in the following section). In addition, two other measures commonly used to access FCRM are calculated during this phase.

5.3. Clustering Validation Techniques

The aim of clustering validation is to evaluate the clustering results by finding the best partition that fits the underlying data best. Thus, cluster validity is used to quantitatively evaluate the results of clustering algorithms. Compactness and separation are two widely considered criteria for measuring the quality of the partitioning of a data set into different numbers of clusters.

Conventional approaches use an iterative approach by choosing different input values, and they select the best validity measure to determine the “optimum” number of clusters. A list of validity indices which have been introduced in Section 4.3.4 of Chapter 4 for fuzzy clustering are adopted. There are the Partition Coefficient (PC) (see Eq. 4.11), Partition Entropy (PE) (see Eq. 4.12), and Modified Partition Coefficient (MPC) (see Eq. 4.13), respectively. PC obtains its maximum value when the cluster structure is optimal. PE achieves its minimum value when the cluster structure is optimal. An optimal cluster number is found by maximizing MPC to produce the best clustering performance for a data set.

5.4. Experiments and Results

This section describes the experimental setup used, and the results obtained by the experiments conducted. In particular, a comparison of the cluster performance in the *A*B space is conducted applying FCM, GK (Gustafson-Kessel), and the proposed FCRM approach. Then, the different validity indices are compared with, followed by a comparison of the mean square error and the peak-signal-to-noise ratio. The last subsection shows the segmentation results.

5.5. Experimental Setup

The experiments are implemented and evaluated on an ASUS desktop (Intel(R) Dual Core I3 CPU @3.07 GHz, 3.07 GHz) Matlab Version 7.13. In order to evaluate the performance of the proposed method, the algorithm has been tested using 15 images from Berkeley Segmentation Database [141] for color image segmentation. In addition, the two other fuzzy clustering algorithms, FCM and Gustafson-Kessel (GK), have been used to compare FCRM with. Table 30 lists the required parameters used when running FCM, GK and FCRM.

5.6. Experimental Study

5.6.1. Comparison of cluster performance in *A*B space

The cluster performance of IMG1 (all fifteen images are denoted by their number) with $c = 3$ is displayed in Figure 19. The figure on the left is the original image. The remaining three figures show the cluster centers in CIE-L*A*B* color space using FCM, GK, and FCRM, respectively.

Table 30. Parameters and their values of the FCM, GK, and FCRM algorithms

Parameter	**Value**
$\lvert U^l - U^{l-1} \rvert < \epsilon$	10^{-3}
Fuzzification coefficient (m)	2
Maximum number of clusters	10
Image data	IMG1-15

The three hyper-spherical centers obtained by FCM and GK, and the three hyper-plane-shaped clusters obtained by FCRM are listed in Table 31.

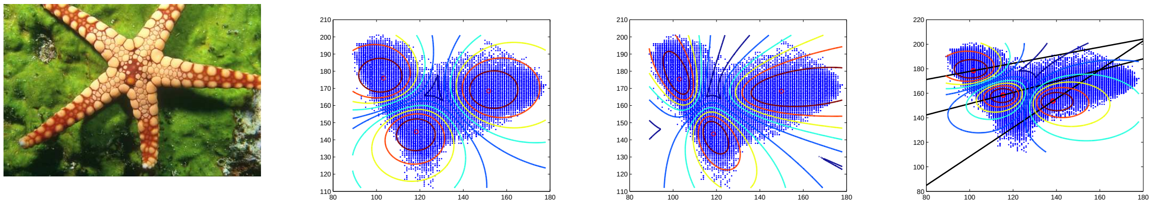

Figure 19. Original image, FCM, GK, and FCRM with c = 3 in *a*b color space

Table 31. FCM, GK, and FCRM using three different indices (PC, PE and MPC)

	FCM	**GK**	**FCRM**
Cluster 1	(151.73, 168.49)	(122.26, 146.33)	$y_1 = 0.3297 \times (x - 80) + 171.243$
Cluster 2	(103.24, 175.97)	(114.69, 132.16)	$y_2 = 1.1788 \times (x - 101.34) + 110$
Cluster 3	(118.33, 144.69)	(135.42, 158.93)	$y_3 = 0.4578 \times (x - 80) + 142.315$

As shown in Figure 19, the proposed FCRM partitions the image into 3 hyper-planed clusters, while FCM and GK group the image into hyper-spherical clusters, respectively. The FCRM method provides better results of the constructed fuzzy model as compared to FCM and GK.

5.6.2. Comparison using different validity indices

Table 32 lists the cluster performance of FCM, GK, and FCRM using validity index PC, PE, and MPC, respectively. As shown in the table, the values in bold denote the best values obtained from the three different validity indices. In most cases, FCRM has the better performance compared to FCM and GK.

Table 32. FCM, GK, and FCRM using three different indices (PC, PE and MPC)

		PC			**PE**			**MPC**	
	FCM	**GK**	**FCRM**	**FCM**	**GK**	**FCRM**	**FCM**	**GK**	**FCRM**
IMG1	0.77	0.77	**0.81**	0.44	0.44	**0.35**	0.65	0.65	**0.71**
IMG2	0.79	**0.82**	0.78	0.38	**0.34**	0.39	0.68	**0.73**	0.68
IMG3	0.72	0.69	**0.78**	0.50	0.55	**0.40**	0.58	0.53	**0.67**
IMG4	0.68	0.70	**0.80**	0.57	0.53	**0.35**	0.51	0.54	**0.70**
IMG5	0.67	0.68	**0.80**	0.58	0.56	**0.36**	0.51	0.52	**0.70**
IMG6	0.66	**0.83**	0.81	0.56	**0.32**	0.35	0.50	**0.75**	0.71
IMG7	**0.88**	0.83	**0.88**	0.23	0.30	**0.22**	0.81	0.75	**0.83**
IMG8	0.77	0.77	**0.82**	0.42	0.41	**0.33**	0.66	0.65	**0.73**
IMG9	0.74	0.75	**0.79**	0.46	0.45	**0.38**	0.61	0.63	**0.68**
IMG10	0.69	0.72	**0.78**	0.55	0.48	**0.39**	0.54	0.58	**0.67**
IMG11	0.75	**0.77**	**0.77**	0.46	**0.40**	0.41	0.62	**0.66**	**0.66**
IMG12	0.75	0.74	**0.80**	0.46	0.46	**0.37**	0.62	0.61	**0.69**
IMG13	0.75	0.74	**0.83**	0.45	0.47	**0.31**	0.63	0.62	**0.74**
IMG14	0.85	0.84	**0.87**	0.28	0.30	**0.25**	0.77	0.76	**0.80**
IMG15	0.65	0.66	**0.78**	0.62	0.60	**0.40**	0.48	0.48	**0.67**

In addition, the best cluster number of FCM, GK, and FCRM obtained by using PC, PE, and MPC are listed in Table 33. In most cases we can see that the best cluster number is 2 when using PC, PE, and MPC as the validity index.

Table 33. Best cluster number of FCM, GK and FCRM using PC, PE and MPC

		PC			**PE**			**MPC**	
	FCM	**GK**	**FCRM**	**FCM**	**GK**	**FCRM**	**FCM**	**GK**	**FCRM**
IMG1	3	2	3	2	2	2	2	2	2
IMG2	2	2	2	2	2	3	2	2	7
IMG3	2	2	2	2	2	2	2	2	2
IMG4	2	2	2	10	10	10	2	2	10
IMG5	2	2	2	2	2	2	2	2	2
IMG6	2	2	2	3	3	3	2	2	2
IMG7	2	2	2	2	2	2	2	2	2
IMG8	2	2	2	3	2	3	2	2	3
IMG9	2	2	2	2	2	2	2	2	2
IMG10	2	2	5	2	2	2	2	2	2
IMG11	2	2	4	2	2	3	2	2	4
IMG12	2	2	2	2	2	2	2	2	2
IMG13	2	2	5	2	2	3	2	2	3
IMG14	2	2	2	2	2	10	2	2	2
IMG15	2	2	2	2	2	2	2	2	2

5.6.3. Comparison with MSE and PSNR

Mean Square Error (MSE) [142] and Peak Signal-to-Noise Ratio (PSNR) [143] are used as the performance indices in fuzzy modeling, which are defined as:

$$MSE = \frac{1}{n}\sum_{k=1}^{n}(y_k - \hat{y}_k)^2 \tag{5.8}$$

$$PSNR = 10 \times log_{10}(255 \times 255/MSE) \tag{5.9}$$

Table 34. MSE ($\times 10^4$) using FCM with different cluster number

	c=2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
IMG1	0.59	0.41	0.32	0.26	0.21	0.19	0.16	0.14	0.13
IMG2	0.37	0.26	0.19	0.16	0.13	0.12	0.10	0.09	0.08
IMG3	0.79	0.54	0.40	0.32	0.27	0.23	0.20	0.18	0.16
IMG4	1.05	0.70	0.50	0.42	0.34	0.29	0.26	0.23	0.21
IMG5	0.39	0.27	0.21	0.16	0.14	0.12	0.10	0.09	0.08
IMG6	0.48	0.41	0.33	0.28	0.23	0.20	0.17	0.15	0.14
IMG7	0.41	0.27	0.26	0.25	0.21	0.17	0.16	0.14	0.13
IMG8	0.50	0.34	0.25	0.20	0.17	0.14	0.13	0.11	0.10
IMG9	0.97	0.66	0.50	0.40	0.33	0.29	0.25	0.22	0.20
IMG10	0.99	0.69	0.56	0.46	0.39	0.33	0.29	0.26	0.23
IMG11	0.93	0.62	0.47	0.38	0.31	0.27	0.24	0.21	0.19
IMG12	0.79	0.55	0.43	0.41	0.34	0.30	0.26	0.23	0.21
IMG13	0.64	0.45	0.34	0.27	0.23	0.20	0.17	0.15	0.13
IMG14	0.73	0.49	0.37	0.24	0.19	0.17	0.15	0.13	0.13
IMG15	0.58	0.40	0.30	0.24	0.20	0.18	0.16	0.14	0.13

Table 35. MSE ($\times 10^4$) using GK with different cluster number

	c=2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
IMG1	0.62	0.42	0.31	0.26	0.21	0.18	0.16	0.14	0.13
IMG2	0.37	0.26	0.19	0.16	0.13	0.12	0.10	0.09	0.08
IMG3	0.79	0.54	0.40	0.32	0.27	0.23	0.20	0.18	0.16
IMG4	0.82	0.62	0.51	0.42	0.35	0.30	0.26	0.23	0.21
IMG5	0.39	0.26	0.20	0.16	0.14	0.12	0.10	0.09	0.08
IMG6	0.52	0.52	0.40	0.32	0.26	0.20	0.18	0.15	0.14
IMG7	0.41	0.27	0.29	0.24	0.20	0.17	0.14	0.13	0.12
IMG8	0.49	0.34	0.25	0.20	0.17	0.15	0.13	0.11	0.10
IMG9	0.97	0.66	0.50	0.40	0.34	0.29	0.25	0.22	0.20
IMG10	0.98	0.78	0.58	0.47	0.39	0.33	0.29	0.26	0.23
IMG11	0.92	0.62	0.47	0.38	0.31	0.27	0.24	0.21	0.19
IMG12	0.78	0.62	0.47	0.38	0.35	0.30	0.26	0.23	0.21
IMG13	0.68	0.42	0.34	0.27	0.23	0.20	0.16	0.14	0.14
IMG14	0.73	0.49	0.37	0.29	0.24	0.21	0.18	0.16	0.15
IMG15	0.59	0.40	0.30	0.25	0.21	0.18	0.15	0.14	0.13

Table 36. MSE ($\times 10^4$) using FCRM with different cluster number

	c=2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
IMG1	0.58	0.40	0.31	0.25	0.21	0.18	0.16	0.14	0.13
IMG2	0.37	0.25	0.19	0.16	0.13	0.12	0.10	0.09	0.08
IMG3	0.80	0.54	0.40	0.32	0.27	0.23	0.20	0.18	0.16
IMG4	1.05	0.70	0.52	0.42	0.35	0.30	0.26	0.23	0.21
IMG5	0.39	0.26	0.20	0.16	0.14	0.12	0.10	0.09	0.08
IMG6	0.48	0.44	0.36	0.28	0.24	0.21	0.18	0.16	0.15
IMG7	0.69	0.27	0.30	0.25	0.21	0.19	0.16	0.14	0.13
IMG8	0.50	0.35	0.26	0.20	0.17	0.15	0.13	0.12	0.10
IMG9	0.98	0.66	0.50	0.40	0.33	0.29	0.25	0.22	0.20
IMG10	0.98	0.80	0.60	0.48	0.40	0.34	0.30	0.27	0.24
IMG11	0.93	0.63	0.47	0.38	0.31	0.27	0.24	0.21	0.19
IMG12	0.80	0.61	0.52	0.42	0.35	0.30	0.26	0.23	0.21
IMG13	0.67	0.45	0.33	0.27	0.23	0.19	0.17	0.15	0.14
IMG14	0.73	0.49	0.37	0.29	0.24	0.21	0.18	0.16	0.15
IMG15	0.57	0.40	0.31	0.25	0.21	0.18	0.16	0.14	0.13

Table 37. PSNR using FCM with different cluster number

	c=2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
IMG1	19.41	12.79	9.49	7.59	6.31	5.40	4.73	4.20	3.78
IMG2	21.48	14.18	10.60	8.37	7.03	5.96	5.21	4.65	4.14
IMG3	16.74	11.11	8.38	6.70	5.58	4.79	4.19	3.72	3.35
IMG4	14.65	9.76	7.45	5.86	4.95	4.25	3.69	3.29	2.93
IMG5	21.58	14.27	10.63	8.49	7.07	6.06	5.30	4.71	4.24
IMG6	19.77	12.12	8.83	6.98	5.82	4.99	4.36	3.88	3.49
IMG7	20.77	13.83	9.57	7.25	6.05	5.32	4.54	4.03	3.59
IMG8	19.35	12.87	9.65	7.71	6.42	5.51	4.82	4.28	3.83
IMG9	15.52	10.27	7.69	6.14	5.12	4.38	3.83	3.41	3.07
IMG10	14.23	9.76	7.05	5.57	4.64	3.97	3.47	3.08	2.77
IMG11	15.29	10.17	7.62	6.10	5.08	4.35	3.81	3.39	3.05
IMG12	16.38	10.75	7.92	5.87	4.89	4.18	3.64	3.23	2.91
IMG13	17.82	11.61	8.70	6.95	5.79	4.94	4.32	3.84	3.52
IMG14	16.89	11.26	8.45	7.25	6.15	5.18	4.51	4.02	3.57
IMG15	18.60	12.26	9.17	7.29	6.07	5.19	4.53	4.03	3.62

Table 38. PSNR using GK with different cluster number

	c=2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
IMG1	19.15	12.74	9.55	7.59	6.32	5.40	4.73	4.20	3.77
IMG2	21.56	14.16	10.60	8.47	7.05	5.98	5.22	4.65	4.17
IMG3	16.76	11.10	8.32	6.65	5.56	4.78	4.18	3.71	3.33
IMG4	16.22	10.29	7.39	5.86	4.88	4.18	3.71	3.25	2.93
IMG5	21.65	14.35	10.67	8.50	7.07	6.06	5.30	4.71	4.23
IMG6	19.23	11.09	8.30	6.63	5.53	4.93	4.31	3.88	3.49
IMG7	20.79	13.84	9.31	7.34	6.13	5.26	4.73	4.10	3.69
IMG8	19.53	12.85	9.64	7.70	6.42	5.50	4.81	4.28	3.83
IMG9	15.55	10.31	7.69	6.14	5.11	4.38	3.83	3.40	3.06
IMG10	13.91	9.20	6.96	5.55	4.63	3.96	3.48	3.09	2.78
IMG11	15.36	10.18	7.62	6.10	5.08	4.35	3.81	3.39	3.05
IMG12	16.46	10.19	7.63	6.07	4.84	4.15	3.63	3.23	2.90
IMG13	17.46	11.93	8.71	6.94	5.76	4.94	4.44	3.95	3.47
IMG14	16.90	11.27	8.45	6.76	5.63	4.82	4.22	3.75	3.37
IMG15	18.49	12.22	9.13	7.29	6.06	5.20	4.54	4.03	3.62

Table 39. PSNR using FCRM with different cluster number

	c=2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
IMG1	19.52	12.86	9.60	7.64	6.37	5.43	4.76	4.22	3.80
IMG2	21.48	14.19	10.57	8.44	6.98	5.97	5.22	4.63	4.15
IMG3	16.69	11.10	8.32	6.67	5.55	4.76	4.16	3.70	3.33
IMG4	14.66	9.78	7.33	5.86	4.88	4.19	3.66	3.25	2.93
IMG5	21.66	14.33	10.70	8.53	7.08	6.06	5.29	4.70	4.23
IMG6	19.85	11.89	8.64	6.91	5.74	4.91	4.28	3.80	3.42
IMG7	17.51	13.84	9.23	7.26	6.05	5.11	4.47	3.98	3.58
IMG8	19.31	12.73	9.54	7.71	6.36	5.45	4.77	4.24	3.81
IMG9	15.49	10.27	7.70	6.14	5.12	4.38	3.84	3.41	3.07
IMG10	15.02	9.15	6.85	5.48	4.56	3.91	3.42	3.04	2.74
IMG11	15.29	10.16	7.62	6.09	5.08	4.35	3.81	3.38	3.05
IMG12	16.32	10.25	7.28	5.81	4.84	4.15	3.63	3.22	2.90
IMG13	17.50	11.63	8.83	7.03	5.79	5.00	4.37	3.88	3.49
IMG14	16.91	11.26	8.44	6.75	5.62	4.82	4.22	3.75	3.37
IMG15	18.62	12.22	9.10	7.25	6.06	5.18	4.53	4.02	3.60

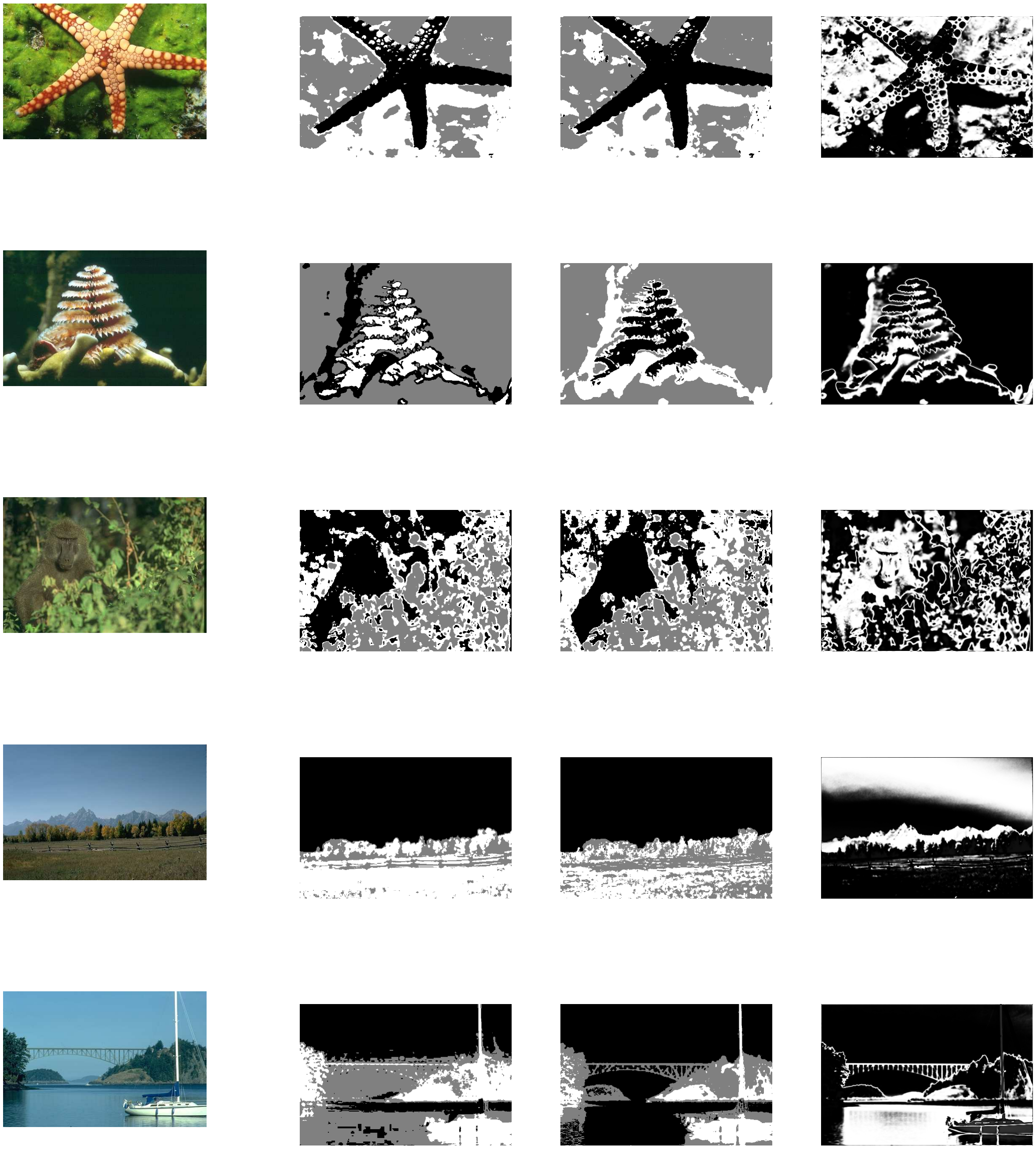

Figure 20. Original image, grayscale image using FCM, GK, and FCRM are listed, respectively

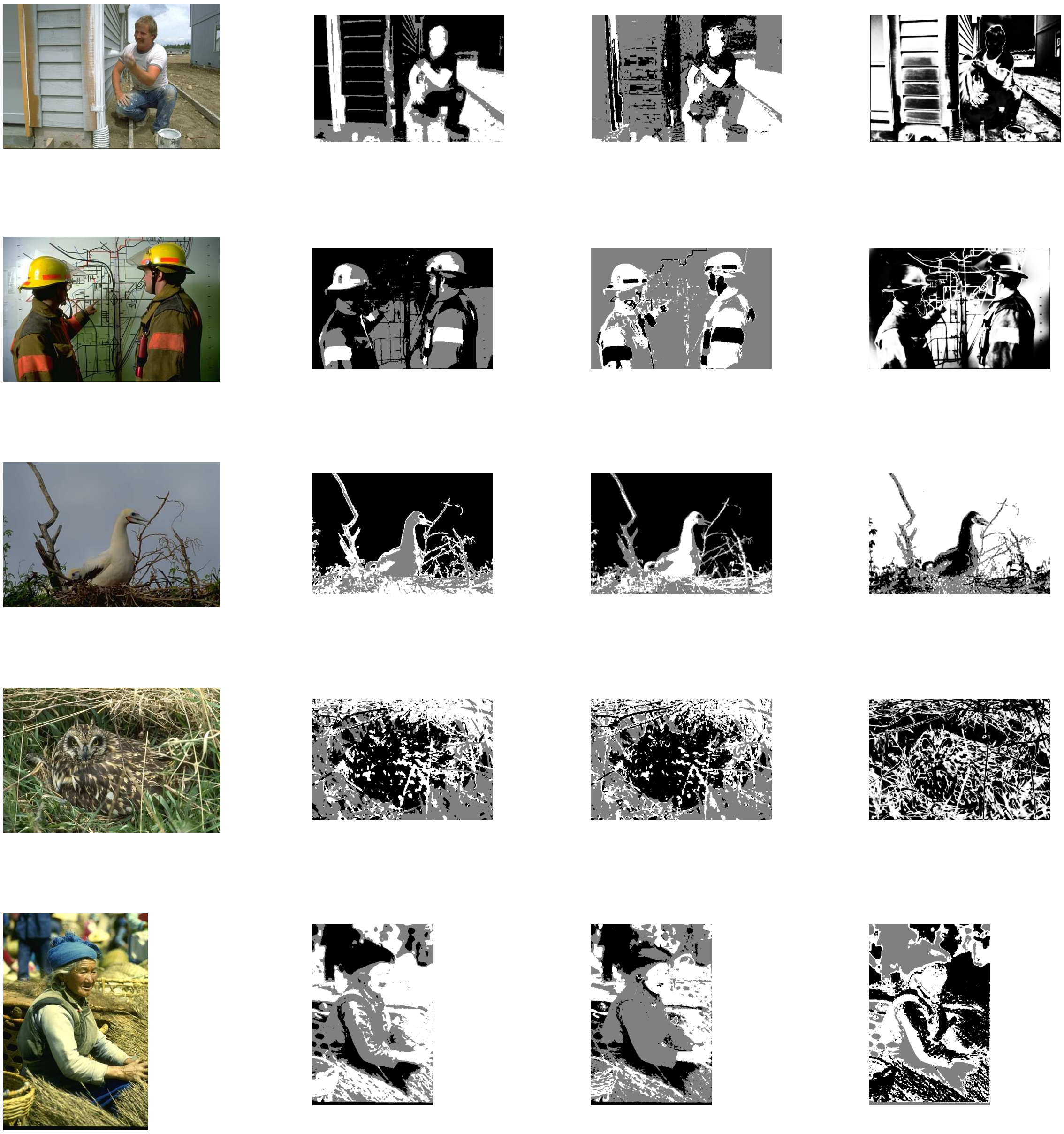

Figure 20. Original image, grayscale image using FCM, GK, and FCRM are listed, respectively (continued)

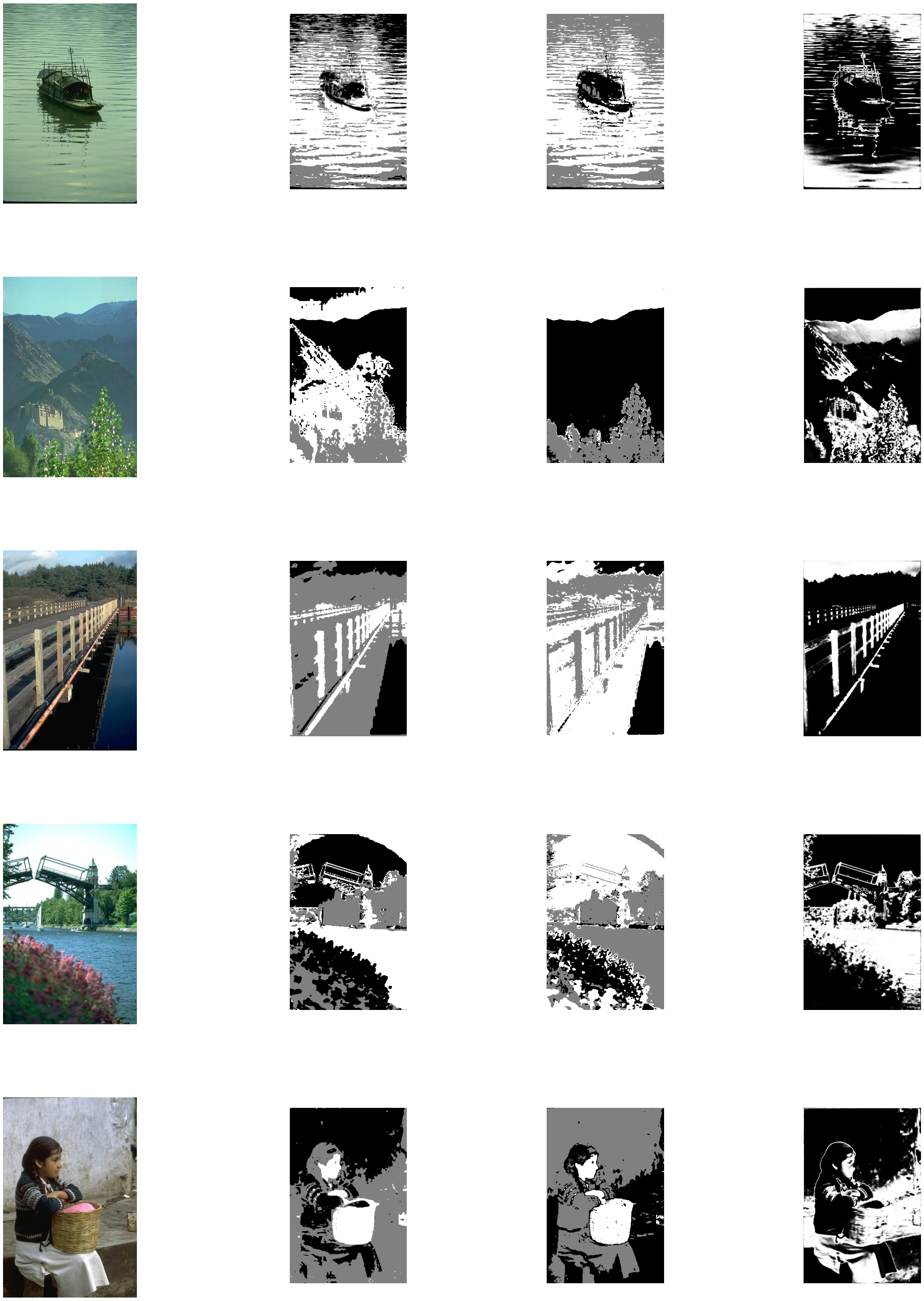

Figure 20. Original image, grayscale image using FCM, GK, and FCRM are listed, respectively (continued)

Table 34, 35, 36 list the MSE of the 15 images using FCM, GK, and FCRM, respectively. Table 37, 38, 39 list the PSNR of the 15 images obtained from FCM, GK, and FCRM, respectively. The MSE and PSNR are measured with a cluster number varying from 2 to 10.

The results show that FCM, GK and FCRM show the same trend regarding MSE and PSNR. As the number of clusters increase, the values of MSE decrease, and the values of PSNR increase for the 15 tested images. In addition, FCRM has a better performance than FCM and GK both in terms of MSE and PSNR. Therefore, FCRM has less partition errors and a more compact representation than FCM and GK.

5.6.4. Comparison on segmentation results

The cluster results are used to reconstruct the image in grayscale level as shown in Figures 20 with $c = 3$. As show in the figures, the FCM, GK, and FCRM can segment the images clearly.

5.7. Summary

Most fuzzy clustering algorithms have been successfully applied in image segmentation. However, the disadvantage they have is that the cluster prototype of FCM (Fuzzy C-Means) is either hyper-spherical or hyper-ellipsoidal. Therefore, FCM may not provide accurate partitioning in circumstances where data is better modeled by arbitrary shapes. Thus, a fuzzy c-regression model clustering algorithm has been introduced whose prototype is hyper-planed and can either be linear or nonlinear. In this chapter, the fuzzy c-regression model clustering algorithm has been successfully applied to color image segmentation. Fuzzy c-regression model is an affine T-S model, which has been successfully used in non-linear system. In addition, due to the complexity of implementation, FCRM has never been used in color image segmentation and was thus explored in this investigation.

The experiments conducted used 15 images that were taken from the Berkeley Segmentation Database. The FCRM was compared against two comparison algorithms (FCM and GK) for color image segmentation. Three validity indices have been used as well as MSE and PSNR were measured. The images were reconstructed using the grayscale level. The experimental results revealed that FCRM achieves better results in most cases than the other approaches based on the aforementioned measures.

As for future work, FCRM is similar to other fuzzy partition techniques, thus, cluster centroids and the number of clusters should be decided in advance. However, for most unknown environments, the appropriate and exact number of clusters is unknown in practice. A new cluster validity criterion needs to be developed to determine the appropriate number of clusters. In addition, FCRM is very sensitive to the initialization. A good initialization results in good quality image segmentation, while an unsuitable initialization returns poor results. Thus, in future, a new technique for automatically finding the exact number of clusters as well as obtaining good initialization need to be investigated.

6. CONCLUSION AND FUTURE WORK

In this current information age, a tremendous expansion in the volume of data is seen that is being generated and stored. The need to understand large, complex, information-rich data sets is common to all fields of studies. Given this tremendous amount of data, efficient and effective tools need to be available to analyze and reveal valuable knowledge that is hidden. The objective of the field of knowledge discovery and data mining is the discovery of knowledge that is not only correct, but also comprehensible. In this dissertation, fuzzy approaches based on fuzzy set theory, fuzzy inference system have combined with particle swarm optimization, decision tree and genetic algorithm and have been applied to solve classification and clustering problems. This chapter concludes the dissertation and is organized as follows. The conclusions of the dissertation are described in Section 6.1 and the future work is illustrated in Section 6.2.

6.1. Conclusions

The two primary goals of data mining can be classified as *prediction* and *description* [3]. *Prediction* involves using some features or fields of the data set to predict unknown or future values of interest, whereas *description* focuses on finding patterns describing the data that can be interpreted by humans. Several data mining techniques using prediction and description have emerged that include classification, clustering, regression, dependence modeling, etc. The classification technique is used to discover a predictive learning function that classifies a data item into several predefined classes. It is also known as supervised classification, whereby given class labels are ordered to objects in the data collection. Clustering analysis is one of the popular approaches and has been widely used in data mining, and is a process to identify groups or clusters based on some similarity measures. The study of this dissertation is focused on two main paradigms. The first paradigm focuses on applying fuzzy inductive learning on classification problems. The second paradigm is fuzzy cluster analysis.

Firstly, a discrete particle swarm optimization with a local strategy (DPSO-LS) for solving the classification problem is proposed. The local search strategy helps to overcome local optima in order to improve the solution quality. The DPSO-LS uses the Pittsburgh approach whereby a rule base is used to represent a ‘particle’. Furthermore, since DPSO-LS can only be applied to discrete data, an additional classifier called Fuzzy DPSO-LS (FDPSO-LS) classifier is implemented for both discrete and continuous data to tolerate imprecision and uncertainty.

Secondly, a decision tree induction method using fuzzy set theory, in other words, Fuzzy Decision Tree (FDT), is becoming an increasingly popular method to solve classification problems. FDT, like classical decision tree, uses the top-down strategy. In order to find the best so called “cut-point”, FDT is based on soft discetization and follows the DT run recursively on each partition until the best cut point is found. The data contains many redundant or irrelevant features. These features provide no useful information in any context. In order to improve the model interpretability and enhance the generalization, a Genetic Algorithm (GA) based feature selector was applied in this chapter. Mutual information is one suitable criterion for feature selection [73]. Mutual information can reduce the uncertainty about the class labels and minimize a lower bound on the Bayes classification error as investigated in [144]. Nevertheless, the estimation of mutual information is not an easy task. Mutual information is a nonlinear measure used to quantify not only linear and but also nonlinear correlations. The challenge of using mutual information for feature selection is the estimation of this measure from the available data.

Thirdly, One of the widely used methods in fuzzy clustering is Fuzzy C-Means clustering (FCM) [24]. FCM attempts to partition a data set into a collection of c fuzzy groups. The algorithm finds a cluster center in each group such that the intra-distance within the group is minimized, and the inter-distance between each group is maximized. Most of the fuzzy clustering methods that have been applied recently use an extension of the FCM algorithm. As we have mentioned before, partitional clustering suffers from the following drawbacks:

1. The number of clusters needs to be pre-specified, and prior knowledge or ground truth is required of the data.

2. Most data points in overlapping areas cannot be categorized correctly.

In order to address these two shortcomings, we proposed a fuzzy c-means clustering approach using a Particle Swarm Optimization (PSO) approach that is applied to clustering analysis.

Lastly, the Fuzzy C-Regression Model (FCRM) was introduced by Hathaway and Bezdek [133, 134]. Due to their excellent capability of describing complex systems in a human intuitive way, FCRM is capable of handling perceptual uncertainties and describing nonlinear system. FCRM, which can be viewed as an extension of FCM, divides the data set into a group of different regression models. Unlike FCM, the clustering prototype of FCRM is a hyper-plane while FCM is hyper-spherical.

However, because of the complexity of image segmentation and given that only partial prior knowledge is provided, the segmentation result would be poor if a supervised method was adopted. Thus, the unsupervised method is a better choice to solve such a problem. Although fuzzy theory has been employed in image segmentation, the application of FCRM to color images has been limited. In this study, we have explored the applicability and soundness of FCRM in color image segmentation. Although FCM can partition the fuzzy space efficiently, it does not take linearity of the divided data into consideration. In contrast, the FCRM clustering algorithm with hyperplane-shaped cluster prototypes has much more explanatory power, especially due to its multivariate nature.

6.2. Future Work

Although the proposed approaches worked well in solving classification and clustering problems, this is just a beginning. There is still much work to be done in the field of data mining using fuzzy approaches.

First, work described in Chapter 2 suffers from a large number of runs to obtain a better average result due to the stochastic nature of particle swarm optimization. Thus, it would be interesting to improve the proposed algorithm to achieve more stable predictions in less runs. In addition, the Pittsburgh approach suffers from bad quality rules within the rule set when only the overall performance is considered, the quality of each rule is not taken into account. Thus, it would be interesting to compare the Pittsburgh approach with the Michigan approach. Moreover, FDPSO-LS has shown improved results compared to FURIA, which could still be improved by minimizing the number of rules and deleting replicated rules. Furthermore, the proposed FDPSO-LS can be further improved by applying discrete data sets with larger ranges of attribute values.

Second, work illustrated in Chapter 3 only used a genetic algorithm approach on feature selection. The results revealed that the approaches using soft discretization rather than hard discretization, such as FURIA and our FDT classifier, obtained better predictive classification accuracy. The proposed classifier achieved slightly better results than FURIA in most cases. However, FDT suffers lot execution time by comparing to FURIA. In future, we will investigate and compare other feature selection techniques available in terms of improvements in accuracy but also in terms of execution time.

Third, due to the slow convergence and the stochastic nature of the PSO algorithm, the prediction results of a single run vary and thus make it difficult to prediction the correct number of clusters using the work discussed in Chapter 4. Unlike K-means and FCM, the proposed algorithm needs to be executed repeatedly in order to find the correct solution. In addition, the maximum number of clusters has to be predefined, and the iterative process to identify the optimal number of clusters is computationally expensive. As for future work, it would be interesting to improve the proposed algorithm to achieve more stable predictions with fewer runs. Thus, a kernel based technique and a new validity index would be interested to investigate to over come such problem. Moreover, we are planning to explore the proposed algorithm with big data sets, and therefore parallelization techniques are necessary.

Fourth, FCRM described in Chapter 5 is similar to other fuzzy partition techniques, thus, cluster centroids and the number of clusters should be decided in advance. However, for most unknown environments, the appropriate and exact number of clusters is unknown in practice. A new cluster validity criterion needs to be developed to determine the appropriate number of clusters. In addition, FCRM is very sensitive to the initialization. A good initialization results in good quality image segmentation, while an unsuitable initialization returns poor results. Thus, in future, a new technique for automatically finding the exact number of clusters as well as obtaining good initialization need to be investigated.

Generally, the standard definition of knowledge discovery and data mining concentrate on highly structured and precise data. The conventional methods like decision tree and neural network are hardly adequate for mining image, sound, and textual data [145]. In the light of the capability of handling uncertainty at various stages, fuzzy approaches can play an important role in data mining especially in information mining [146].

Hence, the future work of this dissertation includes two directions. First, fuzzy logic can improve the classification system by using fuzzy sets to define overlapping class definitions. The interpretability of the results can be improved and more insight into the classifier structure and decision making process would be provided by the application of fuzzy IF-THEN rules. In effect, the tolerance for imprecision and uncertainty is exploited through granulation in soft data compression by using linguistic variables and fuzzy IF-THEN rules. In future, classification tasks based on fuzzy set theory is a direction to investigate which can translate computer representations into human understandable knowledge or concepts. Second, clustering techniques have been widely applied in science, engineering, business and economics, life sciences, biological and medical disciplines [145]. Fuzzy clustering becomes quite prominent in the framework of clustering. In future, theories and scalable techniques that can extract knowledge from large and dynamic data sources need to be exploited and developed.

In conclusion, fuzzy set theory and fuzzy systems have been successfully applied to model human expert knowledge which are comprehensive and easy to understand. Thus, we believe fuzzy approaches will play a more prominent role in the area of data mining.

BIBLIOGRAPHY

[1] H. A. Edelstein, *Introduction to data mining and knowledge discovery (3rd ed),* Potomac, MD: Two Crows Corp. 1999.

[2] B. Mirkin, *Clustering: A Data Recovery Approach,* Second Edition, Chapman & Hall/CRC Computer Science & Data Analysis, 2012.

[3] J. Jackson, *Data Mining: A Conceptual Overview, Communications of the Association for Information Systems*: Vol. 8, Article 19, 2002.

[4] G. Beni, J. Wang, *Swarm Intelligence in Cellular Robotic Systems,* NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 2630, 1989.

[5] A. Colorni, M. Dorigo and V. Maniezzo, *Distributed Optimization by Ant Colonies,* Paris, France, Elsevier Publishing, 134-142, 1991.

[6] R. C. Eberhart and J. Kennedy, *A New Optimizer using Particle Swarm Theory,* In Proc. 6th Symp. Micro Machine and Human Science, Nagoya, Japan, 29-43, 1995.

[7] L. A. Zadeh, *Fuzzy sets,* Information and Control, Vol. 8, pp. 338-353, 1965.

[8] L. A. Zadeh, *The concept of linguistic variable and its application to approximate reasoning,* Inf Sci, 8, 1975.

[9] F. Provost and V. Kolluri, *A survey of methods for scaling up inductive algorithms,* Data Mining and Knowledge Discovery, 3(2):131169, 1999.

[10] P. Geurts, D. Ernst, and L. Wehenkel, *Extremely randomized trees,* Machine Learning, 36(1):342, 2006.

[11] E. Hullermeier, and S. Vanderlooy. *Why fuzzy decision trees are good rankers,* Fuzzy Systems, IEEE Transactions on 17, no. 6: 1233-1244, 2009.

[12] J. R. Quinlan. *Induction of Decision Trees,* Mach. Learn. 1, 1, 81-106, 1986.

[13] J. R. Quinlan. *C4. 5: programs for machine learning,* Vol. 1. Morgan kaufmann, 1993.

[14] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees,* Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0-412-04841-8, 1984.

[15] L. Huan, F. Hussain, C. L. Tan, and M. Dash. *Discretization: An enabling technique,* Data mining and knowledge discovery 6, no. 4: 393-423, 2002.

[16] H. S. Nguyen. *On exploring soft discretization of continuous attributes.* In Rough-Neural Computing, pp. 333-350. Springer Berlin Heidelberg, 2004.

[17] R. B. Cattell, *The description of personality: Basic traits resolved into clusters,* Journal of Abnormal and Social Psychology, 38, 476-506, 1943.

[18] S. Theodoridis, and K. Koutroubas, *Pattern Recognition,* Academic Press, 1999.

[19] T. N. Pappas, *An adaptive clustering algorithm for image segmentation,* IEEE Trans. Signal process, vol. 40, pp.901-914, 1992

[20] A. Likas, N. Vlassis, and J. Verbeek, *The global k-means clustering algorithm (Technical Report),* Computer Science Institute, University of Amsterdam, The Netherlands. ISA-UVA-01-02. 2001.

[21] V. P. Guerrero-Bote, et al., *Comparison of neural models for document clustering,* Int. Journal of Approximate Reasoning, vol. 34, pp.287-305, 2003.

[22] G. L. Carl, *A fuzzy clustering and fuzzy merging algorithm, Technical Report,* CS-UNR-101, 1999.

[23] R. Babuska, *Fuzzy Modelling for Control,* Kluwer Academic, USA, 1998.

[24] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms,* ISBN 0-306-40671-3, 1981.

[25] K. Cios, W. Pedrycz, and R. Swiniarski, *Data Miming methods for Knowledge Discovery,* Kluwer Academic Publishers, 1998.

[26] M. V. Fidelis, H. S. Lopes, A. A. Freitas and P. Grossa, *Discovering comprehensible classification rules with a genetic algorithm,* In Proceedings of the 2000 Congress on Evolutionary Computation, La Jolla, CA, USA, IEEE, vol. 1, pp. 805-810, 2000.

[27] D. R. Carvalho and A.A. Freitas, *A genetic-algorithm for discovering small-disjunct rules in data mining,* Applied Soft Computing, vol. 2, pp. 75-88, 2002.

[28] K. K. Bharadwaj and B.M. Al-Maqaleh, *Evolutionary approach for automated discovery of censored production rules,* In Proceedings of the 8th International Conference on Cybernetics, Informatics and Systemics (CIS-2005). vol. 10, Krakow, Poland, pp. 147-152, 2005.

[29] K. K. Bharadwaj and B.M. Al-Maqaleh, *Evolutionary approach for automated discovery of augmented production rules,* International Journal of Computational Intelligence. vol. 3, Issue 4, pp. 267-275, 2006.

[30] S. Yogita and D. Kumar, *Rule Exceptions: Automated discovery of comprehensible decision Rules,* IEEE International Advance Computing Conference (IACC2009), Patiala, India, pp. 1479-1483, 2009.

[31] H. Ishibuchi, and Y. Nojima. *Evolutionary multiobjective optimization for the design of fuzzy rule-based ensemble classifiers,* International Journal of Hybrid Intelligent Systems 3.3: 129-145, 2006.

[32] R. S. Parepineslli, H.S. Lpes and A. Freitas. *An Ant Colony Algorithm for classification Rule Discovery,* Data Mining: a Heuristic Approach, Idea Group Publishing, 2002.

[33] N. Holden and A. A. Freitas, *A Hybrid PSO/ACO Algorithm for Discovering Classification Rules in Data Mining,* Journal of Artificial Evolution and Applications, vol. 2008, Article ID 316145, 11 pages, 2008.

[34] Z. Wang, X. Sun and D. Zhang, *A PSO-Based Classification Rule Mining Algorithm,* In proceedings of the 3rd International Conference on Intelligent Computing: Advanced Intelligent Computing Theories and Applications, 2007.

[35] H. Ishibuchi, T. Murata, and I. B. Turksen, *Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems,* Fuzzy Sets and Systems, vol. 89, pp. 135-150, 1997.

[36] C. H. Wang, T. P. Hong, and S. S. Tseng, *Integrating fuzzy knowledge by genetic algorithms,* Fuzzy Sets and Systems, vol. 2, no. 4, pp. 138-149, 1998.

[37] H. Ishibuchi, T. Nakashima, and T. Murata, *Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems,* IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics, vol. 29, no. 5, pp. 601-618, 1999.

[38] M. Russo, *FuGeNeSys- a fuzzy genetic neural system for fuzzy modeling,* IEEE Transactions on Fuzzy Systems, vol. 6, no. 3, pp. 373-388, 1998.

[39] L. Wang and J. Yen, *Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter,* Fuzzy Sets and Systems, vol. 101, pp. 353-362, 1999.

[40] Y. Shi, R. Eberhart, and Y. Chen, *Implementation of evolutionary fuzzy systems,* IEEE Transactions on Fuzzy Sytems, vol. 7, no. 2, pp. 109-119, 1999.

[41] O. Cordon, F. Herrera, F. Hoffmann, L. Magdalena, *Genetic fuzzy system: Evolutionary tuning and learning of fuzzy knowledge bases,* Advances in fuzzy systemsapplications and theory, vol 19. World Scientific, Singapore, 2001.

[42] M. F. Ganji, M. S. Abadeh, *Using fuzzy ant colony optimization for diagnosis of diabetes disease,* Electrical Engineering (ICEE), 2010 18th Iranian Conference on, vol., no., pp.501-505, 11-13, 2010.

[43] M. F. Ganji, M. S. Abadeh, *Parallel fuzzy rule learning using an ACO-based algorithm for medical data mining,* In Bio-Inspired Computing: Theories and Applications (BIC-TA), IEEE Fifth International Conference on, pp. 573-581, 2010.

[44] C. N. Raju, O. R. Devi, S. Mani and S. Nagendram, *An Improved Ant Colony Optimization Technique by using Fuzzy Inference Rules for Image Classification and Analysis,* International Journal of Advanced Engineering & Application, pp. 230-234, 2010.

[45] R. P. Prado, S. G. Galan, J. E. Exposito and A. J. Yuste, *Knowledge Acquisition in Fuzzy-Rule-Based Systems With Particle-Swarm Optimization,* IEEE T. Fuzzy Systems 18(6): 1083-1097, 2010.

[46] F. Beloufa and M. A. Chikh, *Automatic Fuzzy rules-base generation using Modified Particle Swarm Optimization,* 2012.

[47] M. S. Abadeh, J. Habibi, S. Aliari, *Using a particle swarm optimization approach for evolutionary fuzzy rule learning: a case study of intrusion detection,* Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU06), Paris, France, 2-7 July 2006.

[48] A. Cervantes, I. M. Galvan, P. Isasi, *A comparison between the Pittsburgh and Michigan approaches for the binary PSO algorithm,* Congress on Evolutionary Computation: 290-297, 2005.

[49] L. X. Wang, J. M. Mendel, *Generating fuzzy rules by learning from examples,* IEEE Transactions on Systems, Man, and Cybernetics 22(6) 1414-1427, 1992.

[50] C. T. Sun, *Rule-base structure identification in an adaptive-networkbased fuzzy inference system,* IEEE Transactions on Fuzzy Systems 2(1) 64-73, 1994.

[51] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms,* Plenum Press, New York, 1981.

[52] E. H. Mamdani, and S. Assilian, *An experiment in linguistic synthesis with a fuzzy logic controller,* International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.

[53] M. Sugeno, *Industrial applications of fuzzy control,* Elsevier Science Pub. Co., 1985.

[54] A. Frank & A. Asuncion, *UCI Machine Learning Repository [http://archive.ics.uci.edu/ml],* Irvine, CA: University of California, School of Information and Computer Science, 2010.

[55] W. Cohen, *Fast effective rule induction,* In Proceedings of Twelfth International Conference on Machine Learning, pages 115-123, Tahoe City, CA, July 1995.

[56] E. Frank and I. H. Witten, *Generating accurate rule sets without global optimization,* In Proc. of the Intl Conf. on Machine Learning, pages 144151. Morgan Kaufmann Publishers Inc., 1998.

[57] J. C. Huhn, E. Hullermeier, *FURIA: an algorithm for unordered fuzzy rule induction,* Data Min. Knowl. Discov. 19(3): 293-319, 2009.

[58] M. Kantardzic. *Data Mining: Concepts, Models, Methods, and Algorithms,* IEEE Press & John Wiley, November 2002.

[59] C. Z. Janikow, *Fuzzy decision trees: issues and methods,* IEEE Trans. Systems Man, CyberneticsPart B: Cybernetics 28 (1) 1-14, 1998.

[60] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques,* Morgan Kaufmann, San Francisco, Calif, USA, 2nd edition, 2005.

[61] P. Domingos and M. J. Pazzan. *On the Optimality of the Simple Bayesian Classifier under Zero-One Loss,* Machine Learning 29(2-3): 103-130, 1997.

[62] I. Kononenko, *Machine learning for medical diagnosis: history, state of the art and perspective,* Artificial Intelligence in Medicine, 23(1), 89-109.

[63] P. Langley, *Induction of recursive Bayesian classi?ers,* In Proceedings of the European conference on machine learning (pp. 153-164), 1993.

[64] N. Lavrac, *Data mining in medicine: selected techniques and applications,* In Proceedings of the 2nd international conference on the practical applications of knowledge discovery and data mining (pp.1131), 1998.

[65] D. D. Lewis, *Naive (Bayes) at forty: the independence assumption in information retrieval,* In Proceedings of the 10th European conference on machine learning (pp. 4-15), 1998.

[66] J. Chen, H. Huang, S. Tian, and Y. Qu. *Feature selection for text classification with Nave Bayes,* Expert Systems with Applications 36, no. 3: 5432-5435, 2009.

[67] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty and Information,* Prentice Hall 1988.

[68] Y. H. Peng, T. J. Chen, and P.W. Tse, *Fuzzy Inductive Learning in Data Mining and its Application to Machine Fault Diagnosis,* 2000AMSMA international conference, pp.158-163, Guangzhou, China, 158-163, 2000.

[69] D. D. Lewis and M. Ringuette. *A comparison of two learning algorithms for text categorization,* In Third annual symposium on document analysis and information retrieval, vol. 33, pp. 81-93. 1994.

[70] P. A. Estevez, M. Tesmer, C.A. Perez, and J.M. Zurada. *Normalized mutual information feature selection,* Neural Networks, IEEE Transactions on 20, no. 2: 189-201, 2009.

[71] J. Chen, H. Huang, S. Tian, and Y. Qu. *Feature selection for text classification with Nave Bayes,* Expert Systems with Applications 36, no. 3: 5432-5435, 2009.

[72] J. Richard, and Q. Shen. *New approaches to fuzzy-rough feature selection,* Fuzzy Systems, IEEE Transactions on 17, no. 4: 824-838, 2009.

[73] O. Ludwig, and U. Nunes. *Novel maximum-margin training algorithms for supervised neural networks,* Neural Networks, IEEE Transactions on 21, no. 6: 972-984, 2010.

[74] H. Yang, J. Moody, *Feature selection based on joint mutual information,* Proceedings of Advances in Intelligent Data Analysis (AIDA), Computational intelligence methods and applications (CIMA), Rochester, New York, 1999.

[75] S. S. Skiena, *The Algorithm Design Manual,* Springer. p. 129. ISBN 978-1-84800-069-8, April, 2011.

[76] T. Hastie, R. Tibshirani, J.H. Friedman, *The elements of statistical learning: data mining, inference, and prediction (2nd ed.),* 2009.

[77] J. C. Dunn, *A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters,* Journal of Cybernetics 3: 32-57, 1973.

[78] J. C. Bezdek, C. Coray, R. Gunderson and J. Watson, *Detection and characterization of cluster substructure i. linear structure: Fuzzy c-lines,* SIAM Journal on Applied Mathematics, 40(2), 339-357, 1981.

[79] I. Gath and A. B. Geva, *Unsupervised optimal fuzzy clustering,* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 11(7), pp 773-781, 1989.

[80] J. Yu, and M. S Yang, *A generalized fuzzy clustering regularization model with optimality tests and model complexity analysis,* Fuzzy Systems, IEEE Transactions on 15.5 (2007): 904-915.

[81] Y. T. Kao, E. Zahara, I. W. Kao, *A hybridized approach to data clustering,* Expert Systems with Applications 34 (3), 1754-1762, 2008.

[82] D. N. Cao, J. C. Krzysztof, *GAKREM: a novel hybrid clustering algorithm,* Information Sciences 178, 4205-4227, 2008.

[83] K. R. Zalik, *An efficient k-means clustering algorithm,* Pattern Recognition Letters 29, 1385-1391, 2008.

[84] K. Krishna, Murty, *Genetic k-means algorithm,* IEEE Transactions of System Man Cybernetics Part B-Cybernetics 29, 433-439, 1999.

[85] U. Mualik, S. Bandyopadhyay, *Genetic algorithm-based clustering technique,* Pattern Recognition 33, 1455-1465, 2000.

[86] M. Laszlo, S. Mukherjee, *A genetic algorithm that exchanges neighboring centers for k-means clustering,* Pattern Recognition Letters 28 (16), 2359-2366, 2007.

[87] P. S. Shelokar, V. K. Jayaraman, B. D. Kulkarni, *An ant colony approach for clustering,* Analytica Chimica Acta 509 (2), 187-195, 2004.

[88] S. A. Ludwig, *Clonal Selection based Fuzzy C-Means Algorithm for Clustering,* Proceedings of Genetic and Evolutionary Computation Conference (ACM GECCO), Vancouver, BC, Canada, July 2014.

[89] T. A. Runkler, and C. Katz. *Fuzzy clustering by particle swarm optimization,* Fuzzy Systems, 2006 IEEE International Conference on. IEEE, 2006.

[90] H. C. Liu, J. M. Yih, D. B Wu, S. W. Liu, *Fuzzy C-mean clustering algorithms based on Picard iteration and particle swarm optimization,* Education Technology and Training, 2008. and 2008 International Workshop on Geoscience and Remote Sensing. ETT and GRS 2008. International Workshop on. Vol. 2. IEEE, 2008.

[91] B. J. Zhao, *An ant colony clustering algorithm, Machine Learning and Cybernetics,* 2007 International Conference on. Vol. 7. IEEE, 2007.

[92] G. Gan, J. Wu, and Z. Yang, *A genetic fuzzy k-Modes algorithm for clustering categorical data,* Expert Systems with Applications 36.2 (2009): 1615-1620.

[93] F. Yang, T. Sun, and C. Zhang, *An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization,* Expert Systems with Applications 36.6 (2009): 9847-9852.

[94] T. Niknam, B. Amiri, *An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis,* Applied Soft Computing, 10(1), 183-197.

[95] P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, and M. Valdez, *Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic,* Expert Systems with Applications 40, no. 8 (2013): 3196-3206.

[96] L. Y. Chuang, C. J. Hsiao, and C. H. Yang, *Chaotic particle swarm optimization for data clustering,* Expert systems with Applications 38.12 (2011): 14555-14563.

[97] H. Izakian, and A. Abraham, *Fuzzy C-means and fuzzy swarm for fuzzy clustering problem,* Expert Systems with Applications 38.3 (2011): 1835-1838.

[98] S. Das, A. Abraham, and A. Konar, *Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm,* Pattern Recognition Letters, 29(5), 688-699, 2008.

[99] M. Chen and S. A. Ludwig, *Fuzzy Clustering Using Automatic Particle Swarm Optimization,* Proceedings of 2014 IEEE International Conference on Fuzzy Systems, Beijing, China, July 2014.

[100] J. C. Bezdek, *Cluster validity with fuzzy sets,* (1973): 58-73.

[101] R. N. Dave, *Validating fuzzy partitions obtained through c-shells clustering,* Pattern Recognition Letters, 17(6), 613-623, 1996.

[102] Y. Fukuyama and M. Sugeno, *A new method of choosing the number of clusters for the fuzzy c-means method,* Proceeding of fifth fuzzy Syst. Sympo., pp.247-250, 1989.

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

[104] N. R. Pal, and J. C. Bezdek, *On cluster validity for the fuzzy c-means model,* Fuzzy Systems, IEEE Transactions on, 3(3), 370-379, 1995.

[105] K. L. Wu, and M. S. Yang, *A cluster validity index for fuzzy clustering,* Pattern Recognition Letters, 26(9), 1275-1291, 2005.

[106] A. Strehl, *Relationship-based clustering and cluster ensembles for high-dimensional data mining,* 2002.

[107] W. Pedrycz and J. V. de Oliveira, *A development of fuzzy encoding and decoding through fuzzy clustering,* IEEE Trans. Instrum. Meas., vol. 57, no. 4, pp. 829?837, Apr. 2008.

[108] B. Balasko, J. Abonyi, and B. Feil. *Fuzzy clustering and data analysis toolbox,* Department of Process Engineering, University of Veszprem, Veszprem, 2005.

[109] J. MacQueen, *Some methods for classification and analysis of multivariate observations,* Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, 281–297, University of California Press, Berkeley, Calif., 1967.

[110] L. Kaufman and P.J. Rousseeuw, *Clustering by means of Medoids,* in Statistical Data Analysis Based on the L_1 Norm and Related Methods, edited by Y. Dodge, North-Holland, 405?416, 1987.

[111] A. Othman and H. Tizhoosh, *Evolving fuzzy image segmentation,* in Proc. IEEE Int. Conf. Fuzzy Syst., pp. 16031609, Jun. (2011).

[112] Y. A. Tolias, S. M. Panas, *On applying spatial constraints in fuzzy image clustering using a fuzzy rule-based system.* Signal Processing Letters, IEEE, 5(10), 245-247, (1998).

[113] H. Tizhoosh, G. Krell, T. Lilienblum, C. J. Moore, and B. Michaelis, *Enhancement and associative restoration of electronic portal images in radiotherapy,* Int. J. Med. Inf., vol. 49, no. 2, pp. 157171, (1998).

[114] G. Karmakar and L. Dooley, *A generic fuzzy rule based image segmentation algorithm,* Pattern Recognition Letters, vol. 23, no. 10, pp. 12151227, (2002).

[115] , G. Karmakar, L. Dooley, and M. Murshed. *Fuzzy rule for image segmentation incorporating texture features.* 2002 International Conference on In Image Processing, vol. 1, pp. I-797. IEEE, (2002).

[116] S. Pal and A. Rosenfeld, *Image enhancement and thresholding by optimization of fuzzy compactness*, Pattern Recognit. Lett., pp. 7786, (1988).

[117] S. K. Pal and A. Ghosh, *Fuzzy geometry in image analysis,* Fuzzy Sets Syst., vol. 48, no. 1, pp. 2340, (1992).

[118] V. V. Krivsha, S. A. Butenkov, *Classification using fuzzy geometric features*, 2002 IEEE International Conference on Artificial Intelligence Systems, vol., no., pp.89,91, (2002).

[119] J. Noordam, W. van den Broek, and L. Buydens, *Geometrically guided fuzzy C-means clustering for multivariate image segmentation*, in Proceedings of International Conference on Pattern Recognition (ICPR00), vol. 1, pp. 462465, (2000).

[120] S. K. Pal, A. Ghosh, B. U. Shankar, *Segmentation of remotely sensed images with fuzzy thresholding, and quantitative evaluation.* International Journal of Remote Sensing, 21(11), 2269-2300.

[121] N. R. Pal, D. Bhandari, and D. D. Majumder, *Fuzzy divergence, probability measure of fuzzy events and image thresholding,* Pattern Recognit., vol. 13, pp. 857867, (1992).

[122] C. A. Murthy and S. K. Pal, *Fuzzy thresholding: Mathematical framework bound functions and weighted moving average technique,* Pattern Recognit. Lett., vol. 11, pp. 197206, (1990).

[123] Q. Wang, Z. Chi, and R. Zhao, *Image thresholding by maximizing the index of nonfuzziness of the 2-D grayscale histogram,* Comput. Vis. Imag. Understand., vol. 85, no. 2, pp. 100116, (2002).

[124] E. Lughofer, *On-line evolving image classifiers and their application to surface inspection,* Imag. Vis. Comput., vol. 28, no. 7, pp. 10651079, (2010).

[125] R. Rajesh, N. Senthilkumaran, J. Satheeshkumar, B. S. Priya, C. Thilagavathy, and K. Priya, *On the type-1 and type-2 fuzziness measures for thresholding MRI brain images,* in Proc. IEEE Int. Conf. Fuzzy Syst., pp. 992995, Jun. (2011).

[126] D. Neog, M. Raza, and F. Rhee, *An interval type 2 fuzzy approach to multilevel image segmentation,* in Proc. IEEE Int. Conf. Fuzzy Syst., pp. 11641170, Jun. (2011).

[127] T. Chaira, *Accurate segmentation of leukocyte in blood cell images using Atanassov's intuitionistic fuzzy and interval Type II fuzzy set theory.* Micron, 61, 1-8, (2014).

[128] L. Szilagyi, Z. Benyo, S. M. Szilagyi, H. S. Adam, *MR brain image segmentation using an enhanced fuzzy C-means algorithm.* In Proceedings of the 25th annual international conference of the IEEE engineering in medicine and biology society, Vol. 1, pp. 1721, (2003).

[129] M. Gong, Y. Liang, J. Shi, W. Ma, J. Ma, *Fuzzy c-means clustering with local information and kernel metric for image segmentation.* Image Processing, IEEE Transactions on, 22(2), 573-584, (2013).

[130] F. Zhao, L. Jiao, H. Liu, X. Gao, *A novel fuzzy clustering algorithm with non local adaptive spatial constraint for image segmentation.* Signal Processing, 91(4), 988-999, (2011).

[131] Z. Ji, Y. Xia, Q. Chen, Q. Sun, D. Xia, D. D. Feng, *Fuzzy c-means clustering with weighted image patch for image segmentation.* Applied soft computing, 12(6), 1659-1667, (2012).

[132] B. Sowmya, B. S. Rani, *Colour image segmentation using fuzzy clustering techniques and competitive neural network.* Applied Soft Computing, 11(3), 3170-3178, (2011).

[133] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, NY, (1981).

[134] J. C. Bezdek, J. Keller, R. Krisnapuram, and N. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Vol. 4, Springer, New York, NY, (1999).

[135] K. S. Tan, W. H. Lim, N. A. M.Isa, *Novel initialization scheme for Fuzzy C-Means algorithm on color image segmentation.* Applied Soft Computing, 13(4), 1832-1852, (2013).

[136] K. Deshmukh, G. N. Shinde, *An adaptive neuro-fuzzy system for color image segmentation.* Journal of the Indian Institute of Science, 86(5), 493, (2013).

[137] R. S. Hunter, *Photoelectric Color-Difference Meter.* JOSA 38 (7): 661, Proceedings of the Winter Meeting of the Optical Society of America, July (1948).

[138] J. D. Foley, et al. *Computer Graphics: Principles and Practice (2nd ed.).* Redwood City, CA: Addison-Wesley, (1995).

[139] R. S. Hunter, Accuracy, *Precision, and Stability of New Photo-electric Color-Difference Meter.* JOSA 38 (12): 1094, Proceedings of the Thirty-Third Annual Meeting of the Optical Society of America, December (1948).

[140] T. Takagi and M. Sugeno, *Fuzzy identification of systems and its applications to modeling and control*, IEEE Trans. Syst. Man Cybern., vol. SMC-15, no. 1, pp. 116132, (1985).

[141] C. Fowlkes, D. Martin, J. Malik. *Local Figure/Ground Cues are Valid for Natural Images*, Journal of Vision, 7(8):2, 1-9, (2007).

[142] E. L. Lehmann, G. Casella, *Theory of Point Estimation (2nd ed.).* New York: Springer, (1998).

[143] Q. Huynh-Thu, M. Ghanbari, *Scope of validity of PSNR in image/video quality assessment*, Electronics Letters 44 (13): 800, (2008).

[144] I. Guyon, and A. Elissee (2003). *An introduction to variable and feature selection.* In Journal of Machine Learning Research, volume 3, pages 1157-1182.

[145] J. V. de Oliveira and W. Pedrycz, *Advances in fuzzy clustering and its applications*. New York: Wiley, 2007.

[146] R. Kruse, D. Nauck, and C. Borgelt. *Data mining with fuzzy methods: status and perspectives.* Proc. 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT99). 1999.